LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT
School of Business and Management
Master’s Programme in Strategic Finance and Analytics

*Emilia Härkönen*

# FORECASTING STOCK INDEX TREND WITH SUPPORT VECTOR MACHINE AND LONG-SHORT TERM MEMORY – A CASE STUDY OF MODELS FITTED ON OMXH25 DATA

Examiners: Professor, D. Sc. Mikael Collan
Post-Doctoral Researcher, D.Sc. Jyrki Savolainen

# ABSTRACT

| | |
|---|---|
| **Author:** | Emilia Härkönen |
| **Title:** | Forecasting stock index trend with Support Vector Machine and Long-Short term memory – A case study of models fitted on OMXH25 data |
| **Faculty:** | School of Business and Management |
| **Master's Programme:** | Strategic Finance and Analytics |
| **Year:** | 2021 |
| **Master's Thesis:** | Lappeenranta-Lahti University of Technology<br>73 pages, 22 figure, 27 tables, 5 appendices |
| **Examiners:** | Professor Mikael Collan<br>Post-Doctoral Researcher Jyrki Savolainen |
| **Keywords:** | Machine learning, Deep learning, SVM, LSTM, Stock market forecasting, Financial time series |

The aim of this thesis is to investigate the predictability of financial markets. The research is conducted by using machine learning and deep learning techniques to predict the next day's direction of the stock index return. Support Vector Machine (SVM) is chosen as a machine learning model and Long Short-Term Memory (LSTM) as a deep learning model. The chosen models have proved their stock market predicting capability in previous studies. The studies have pointed out the superiority of deep learning models in stock market forecasting. This study involves to the debate by conducting a case study of the Finnish stock index – OMXH25. The LSTM and SVM models are trained for the OMXH25 data, but the models are tested for the three correlated datasets, namely: OMXH25, S&P 500, and FTSE 100.

The sample data is collected from the period 2009-2019. The data sets included the opening, high, low, closing, and adjusted closing price of the indices. The indices' daily returns are calculated from the adjusted closing price and transformed to binary variables indicating positive and negative returns. The empirical part consists of preprocessing of data where input variables are transformed to percentage returns and standardized. The data of OMXH25 was divided into training (80%), validation (10%), and testing data (10%). Parameter optimization of both models was conducted by predicting the validation set and based on these results, the optimal parameter combinations were chosen. The optimized models were used to predict testing sets of all three indices. Predicting performance was evaluated by using accuracies, confusion matrices, and precisions. The results of the LSTM and SVM models were also benchmarked with a random guess.

The LSTM model outperformed the SVM model and a random guess when predicting the OMXH25, S&P 500, and FTSE 100 testing sets. The results of LSTM were the most promising ones, and the LSTM model can increase the predicting accuracy over a random guess up to five percent. The SVM models' accuracy was over 50%, but the confusion matrices revealed that the predictions were overweighted to positives due to the overfitting problem. However, also the SVM model outperformed a random guess. The accuracy of the LSTM model was the highest for the OMXH25. The results still evince that the other similar indices' predictability does not significantly decrease when the model is trained with one index.

# TIIVISTELMÄ

| | |
|---|---|
| **Tekijä:** | Emilia Härkönen |
| **Tutkielman nimi:** | Osakeindeksin suunnan ennustaminen Support Vector Machine - ja Long-Short term memory -menetelmillä – tapaustutkimus OMXH25 datalla sovitetuista malleista |
| **Tiedekunta:** | Kauppakorkeakoulu |
| **Maisteriohjelma:** | Strategic Finance and Analytics |
| **Vuosi:** | 2021 |
| **Pro Gradu -tutkielma:** | Lappeenrannan-Lahden teknillinen yliopisto<br>73 sivua, 22 kuviota, 27 taulukkoa, 5 liitettä |
| **Tarkastajat:** | Professori Mikael Collan<br>Tutkijatohtori Jyrki Savolainen |
| **Avainsanat:** | Koneoppiminen, Syväoppiminen, SVM, LSTM, Osakemarkkinan ennustaminen, Taloudellinen aikasarja |


Tämän tutkielman tarkoituksena on tutkia rahoitusmarkkinoiden ennustettavuutta. Tutkimus on toteutettu hyödyntämällä koneoppimisen sekä syväoppimisen tekniikoita ja tutkimuksessa ennustetaan seuraavan päivän osakeindeksin suuntaa. Koneoppimisen tekniikaksi on valittu Support Vector Machine ja syväoppimisen tekniikaksi Long-Short term memory. Nämä mallit ovat esittäneet todisteita osakemarkkinan ennustamiskyvystä aiemmissa tutkimuksissa. Tutkimuksissa on osoitettu syväoppimisen mallien olevan parempia osakemarkkinoiden ennustamisessa, ja tässä tutkimuksessa otetaan kantaa aiheeseen tapaustutkimuksella OMX Helsinki 25 -indeksistä. LSTM ja SVM mallit opetetaan OMXH25 aineistolla, mutta samoja malleja käytetään ennustamaan seuraavia kolmea keskenään korreloituneita indeksejä: OMXH25, S&P 500 ja FTSE 100.

Tutkimusaineisto on ajanjaksolta 2009–2019. Aineistot sisältävät avaus-, korkeimman, alimman, päätös- ja oikaistun päätöskurssin indekseistä. Indeksien päivätuotot on laskettu oikaistusta päätöskurssista ja ne on muutettu binäärisiksi muuttujiksi, jotka viittaavat positiivisiin tai negatiivisiin tuottoihin. Tutkimuksen empiirisen osuuden datan käsittelyssä muuttajat muutetaan tuottoprosenteiksi ja standardisoidaan. OMXH25:n data on jaettu opetusdataan (80 %), validointidataan (10 %) ja testausdataan (10 %). Molempien mallien parametrioptimoinnissa ennustetaan validointidataa, minkä tuloksien pohjalta valitaan paras yhdistelmä parametreista. Optimoiduilla malleilla ennustettiin jokaisen kolmen indeksin testausdataa. Ennustamistehokkuutta on arvioitu käyttämällä tarkkuutta, confusion matriiseja ja täsmällisyyttä. LSTM ja SVM mallien tuloksia verrataan myös satunnaisarvaukseen.

LSTM malli suoriutui SVM mallia ja satunnaisarvausta paremmin ennustamaan OMXH25, S&P 500 ja FTSE 100 indeksien testausdataa. LSTM mallin tulokset olivat lupaavimmat ja malli pystyy kasvattamaan ennustustarkkuutta viiteen prosenttiin asti yli satunnaisarvauksen. SVM mallin tarkkuudet olivat myös yli 50 %, mutta confusion matriisit paljastivat ennusteiden painottuvan positiivisiin johtuen mallin ylisovittamisongelmasta. SVM malli suoriutui kuitenkin satunnaisarvausta paremmin. LSTM mallin tarkkuus oli korkein OMXH25:n dataa ennustettaessa. Tulokset osoittavat silti, että toisen samankaltaisen indeksin ennustettavuus ei merkitsevästi laske, vaikka malli on opetettu toisella indeksillä.

## Acknowledgements

The last years at LUT have been memorable and full of awesome moments. This has been an amazing period in my life, which has taught me a lot and helped me to grow also as a person. I'm incredibly grateful for the lifelong friendships I made during these years. It was a pleasure to go through this academic journey together with you.

I want to thank my supervisor Jyrki Savolainen for the guidance and support with this thesis. Your feedback has been invaluable, and the thesis would not be the same without your help.

I want to express my gratitude to my family for always being there for me. Thank you, Teemu, for your continuous encouragement and support you have given me. I appreciate your help more than you all can ever imagine.

Lappeenranta, 25th of May 2021
Emilia Härkönen

Table of Contents

## LIST OF APPENDICES

## LIST OF FIGURES



## LIST OF TABLES

## List of abbreviations

| | |
|---|---|
| **AB** | AdaBoost |
| **ANN** | Artificial neural network |
| **AR** | Autoregressive model |
| **ARIMA** | Autoregressive integrated moving average model |
| **BLSTM** | Bidirectional long short-term memory |
| **CBR** | Case-based reasoning |
| **CEEMD-PCA-LSTM** | Combination of complementary ensemble empirical mode decompostion, principal component analysis, LSTM |
| **CNN** | Convolutional neural network |
| **DNN** | Deep neural network |
| **EWT-dpLSTM-PSO-ORELM** | Combination of empirical wavelet transform, outlier robust extreme learning machine, dropout LSTM, particle swarm optimization |
| **GMM** | Generalized methods of moments |
| **GRNN** | General regression neural network |
| **GRU** | Gated recurrent unit |
| **KF** | Kernel Factory |
| **KNN** | K-Nearest Neighbors |
| **LOG** | Logistic regression |
| **LSTM** | Long short-term memory |
| **MLP** | Multilayer perceptron |
| **PNN** | Probabilistic neural network |
| **PCA** | Principal component analysis |
| **RF** | Random forest |
| **RNN** | Recurrent neural network |
| **SFM** | State Frequency Memory |
| **SLSTM** | Stacked long short-term memory |
| **SVM** | Support vector machine |
| **TARCH** | Threshold autoregressive conditional heteroskedasticity |
| **WLSTM** | Combination of WT (wavelet transforms) and LSTM |
| **WSAEs-LSTM** | Combination of wavelet transforms, stacked autoencoders and LSTM |

# 1 INTRODUCTION

Stock market forecasting is a difficult task which many academics and practitioners have tried for decades (Kim, 2003; Atsalakis & Valavanis, 2009; Weng, Ahmed & Megahed, 2017). Successful and reliable forecasting models would reduce the risk investors need to bear in their investment decisions (Baek & Kim, 2018). Stock price forecasting is also highly motivated by possible profits that can be gained from speculating (Kumar & Thenmozhi, 2006; Tsantekidis, Passalis, Tefas, Kanniainen, Gabbouj & Iosifidis, 2017).

## 1.1 Background and motivation

Fama (1965) introduced the efficient market hypothesis (EMH), and according to this theory, stock prices are random, and they are not predictable. Original EMH is categorized into three forms which are weak, semi-strong and strong (Fama, 1970). The literature about market efficiency is vast, and many researchers have questioned market efficiency (Atsalakis & Valavanis, 2009; Bao, Yue & Rao, 2017). As a response to criticisms, Fama (1991) reconstructed efficiency into three new forms. The first new form of efficiency means that returns cannot be predicted by using historical data, and the success of some contrary studies is due to measurement errors. Fama replaced semi-strong efficiency with event studies, and he argued that new information is reflected in prices quickly and efficiently. Inefficiency is found only from tests for private information, and there is evidence that corporate insiders have information that is not fully reflected in prices. (Fama, 1991)

Despite EMH, stock price forecasting has gained a lot of interest, and according to Baek and Kim (2018), methods of forecasting stock prices have changed over time. Hellstörm and Holmströmm (1998) divided stock price prediction methodologies into three categories: technical analysis, time series forecasting, and machine learning and data mining. Forecasting methods have developed during the last decades. Traditionally, autoregressive integrated moving average (ARIMA) and autoregressive moving average (ARMA) models have been primarily used in time series forecasting (Pai & Lin, 2005; Ballings, Van den Poel, Hespeels and Gryp, 2015). Also, vector autoregression models (VAR) have been used in time series prediction (Baek & Kim, 2018). Time series forecasting techniques have performed worse than machine learning techniques in the past because of the non-linearity of stock price behavior (Baek & Kim, 2018; Sezer, Gudelek & Ozbayoglu, 2020). Widely used machine learning methods have been artificial neural network (ANN) and support vector machine (SVM) (Baek & Kim, 2018). Recently, deep learning techniques have been the best-performed techniques, and they have outperformed traditional machine learning models in the field of financial time series forecasting (Sezer et al., 2020).

The challenge of forecasting stock returns is due to the complex characteristics of stock returns. Datasets of stock returns are noisy because of high volatility, and forecasting models need to capture non-linearity caused by different volatility periods. There might be periods of low volatility, which can turn to high volatility rapidly. Periods of recession and expansion have different kinds of characteristics that the models need to capture. (Huang, Nakamori & Wang, 2005; Atsalakis & Valavanis, 2009) Stock markets are also highly affected by irrational human behavior, which mathematical models often fail to capture. Especially, deep learning models have been seen as an answer to overcome this problem. (Tsantekidis et al., 2017)

### 1.2 Research questions and the aim of the study

There have been different conclusions about the predictability of stock returns in the literature (Henrique, Sobreiro & Kimura, 2019; Sezer et al., 2020). This study tests Fama's efficient market hypothesis and examines, is the stock market predictable or not. Therefore, the main research question is:

*"How to predict stock indices using machine learning and deep learning techniques?"*

It has been shown in the literature that deep learning models have outperformed machine learning techniques in financial time series forecasting. The first sub-question is motivated by that conclusion (Sezer et al., 2020), and it is:

*"How the performance of deep learning techniques and machine learning techniques differ in stock index prediction?"*

Several studies show that the United States stock market and some developing markets such as China or Taiwan are predicted successfully (Henrique et al., 2019). The predictability of Nordic stock markets is relatively unknown; hence the second sub-question is as follows:

*"How do the selected methods perform with data of OMXH25 index in the period of 2009-2019?"*

This study forecasts the daily movements of the OMXH25 index and, precisely, the direction of the index. The target of the study is to evaluate whether the Finnish stock market is predictable or not with two methods that have provided promising results in previous studies. This thesis will pursue to supplement the field of stock market prediction and unify the results of previous studies. If the models were tested only with the OMXH25 data, the reliability of the results or the predictability would be low. Therefore, the data of the S&P 500 and FTSE 100 are also tested for the models. Thus, also the generalization of those models will be evaluated. The last sub-question is:

*"How the models, fitted with OMXH25-dataset, generalize to correlated datasets of S&P 500 and FTSE 100?"*

The prediction will be implemented with one machine learning and one deep learning model, which have performed well in previous studies. Support vector machine is selected as a machine learning method and long short-term memory as a deep learning method. The one target of this study is to evaluate differences between machine learning and deep learning models in their forecasting performance. (Atsalakis & Valavanis, 2009; Henrique et al., 2019; Sezer et al., 2020)

### 1.3 Limitations of the study

The main object is to study the predictability of the OMXH25 index. The reliability of the study is increased by also predicting the S&P 500 and FTSE 100 by using the same model, which is trained for the OMXH25 data. All three indices belong to developed markets, so the predictability of developing markets will be left out of the scope in this study. Stock markets across the globe have different kinds of characteristics, thus the performance of forecasting models may differ between the markets. Evidence shows that forecasting of stock markets of developing countries has been more challenging compared to developed ones. (Bao et al., 2017; Zhang, Yan & Aasma, 2020) The second limitation is that LSTM's modifications are left out of the scope of this study. Modifications and extensions of LSTM will be discussed in Chapter 3 within the literature review. One of the study's targets is to compare the forecasting performance of machine learning and deep learning. It is implemented by choosing well-performed models in previous studies from both categories. The research data is collected from years between 2009 and 2019. The stock market was relatively stable during the study's time period, and major crises, such as financial crisis and stock market meltdown due to the Covid-19 crisis, are not included. Therefore, the predictability of the Finnish stock market during highly volatile periods will not be evaluated in this study.

### 1.4 Structure of the thesis

The primary theory of the study will be discussed in the next chapter. Chapter 2 explains the most important machine learning and deep learning concepts from the point of view of this study. The most relevant and promising models according to the literature will be theoretically discussed. The third chapter consists of the literature review where previous stock market studies with machine and deep learning methods will be discussed. The empirical part is presented in Chapter 4, and also the sample data, forecasting methods, and performance evaluation will be presented. Chapter 5 presents the results, and the last chapter is for conclusions and a summary of the study.

## 2 THEORETICAL FRAMEWORK

The name of Artificial intelligence (AI) was first introduced at a conference which was organized by John McCarthy in 1956. McCarthy's definition of AI is as follows: "The goal of AI is to develop machines that behave as though they were intelligent." Still, the roots of AI were founded earlier. (Ertel, 2011, 10) One of the most significant achievements was the Turing test, where the intelligence of machine was tested (Turing, 1950). Nobody has invented an inclusive definition for AI even up to this date, but Elaine Rich formulated a generic definition in 1983: "Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better." (Ertel, 2011, 2)

As Figure 1 shows, machine learning is one of the subsets of AI, and it is used in many fields. It is used, for instance, in voice recognition, stock trading, advertising, and medicine. There have been suggested several definitions for machine learning. (Bell, 2020, 1-8) One of the earliest definitions for machine learning was introduced by Arthur Samuel (1959) that machine learning "gives computers the ability to learn without being explicitly programmed." Deep learning is a subset of machine learning. Briefly explained, deep learning differs from normal neural networks because deep learning models have more than one hidden layer (Schmidhuber, 2015). Next, machine learning and deep learning and some of their relevant methods for this study are introduced.

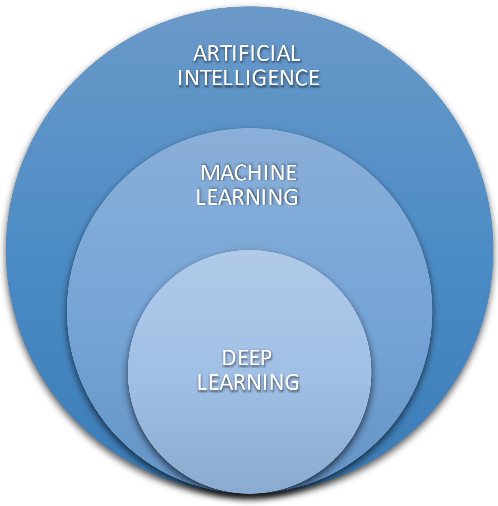


Figure 1. The theoretical framework of the study (Schmidhuber, 2015; Bell, 2020, 3)

## 2.1 Machine Learning

Machine learning can be classified into three classes. Those classes are supervised learning, unsupervised learning, and reinforcement learning, as shown in Figure 2. Supervised learning requires that correct answers to the problem are known in the training phase. Once the model has learned patterns from the training data, the model can be used in regression or classification tasks. Methods concerning unsupervised learning learn new features or patterns from the training data without having correct answers. Problems are often related to clustering and dimensionality reduction. Clustering and dimensionality reduction are both methods where the target is to group similar kinds of features. Reinforcement learning is based on trial and error, and an agent is trying to learn the most feasible solution to solve the given problem. (Dey, 2016) Because this study has applied support vector machine as a machine learning technique, only the path of support vector machine is presented in detail. Classification methods classify observations to the classes, and one observation has to belong only to one class (Bramer 2020, 21-22). Part of the widely used machine learning methods, such as SVM, random forest, and artificial neural network, is introduced in the following sections (Henrique et al., 2019).

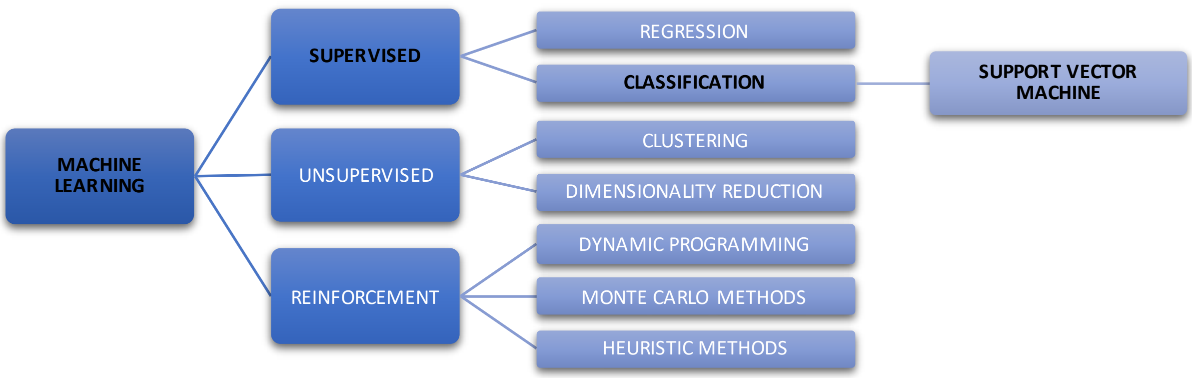


Figure 2. The path from the concept of machine learning to support vector machine (Dey, 2016)

### 2.1.1 Support Vector Machine

Cortes and Vapnik (1995) published the seminal study of support vector machines (SVM). The main idea in SVM is to fit a separating hyperplane with the widest feasible margin to training data. The points which lie in the margin line are called support vectors, and they define boundaries to a binary class. (Cortes & Vapnik, 1995; Burges, 1998; Kim, 2003) The middle line of the hyperplane is called a linear discriminant (Provost & Fawcett 2013, 92-94). The fitting process of SVM is the following: input vectors are mapped into high-dimensional feature space, which is determined beforehand. The mapping process is performed with some nonlinear functions, called Kernel functions which are represented below shortly. Next, a linearly separable decision surface is constructed in the high-dimensional feature space, and this surface is the nonlinear decision boundary in the original feature space. The optimal separating hyperplane, which can also be called a maximum margin hyperplane, is constructed in the high-dimensional space. (Cortes & Vapnik, 1995; Kim, 2003; Huang et al., 2005)

Narrow mathematical expressions of the linearly separable case are presented, and equations follow the notation of Kim (2003). Studies of Cortes and Vapnik (1995), Burges (1998), and Evgeniou, Pontil and Poggio (2000) illustrate more complex expressions of SVM. Equation (1) presents a hyperplane that separates three features in the binary classification task

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3, \tag{1}$$

where $y$ is the result, $x_i$ are the feature vectors and $w_i$ are corresponding weights that the SVM needs to learn. The hyperplane is determined by $w_i$ parameters from equation (1) and equation (2) represent the maximum margin hyperplane concerning the support vectors:

$$y = b + \sum \alpha_i \, y_i x(i) \cdot x, \tag{2}$$

where $y_i$ is the label of the trained observation and $x(i) \cdot$ illustrates the dot product. The support vectors are $x(i)$ and the vector $x$ represents the testing observations. This time, parameters $b$ and $\alpha_i$ determine the hyperplane. Determining the parameters $b$ and $\alpha_i$ as well as identification of support vectors corresponds to solving a linearly constrained quadratic programming problem.

Equation (3) is a high-dimensional version of equation (2), and it is used to solve nonlinear decision boundaries for a separable case as follows:

$$y = b + \sum \alpha_i \, y_i K(x(i), x). \tag{3}$$

Function $K(x(i), x)$ represents the Kernel function which is used to create high-dimensional feature space. Common kernels are:

polynomial kernels (4),

$$K(x, y) = (xy + 1)^d \tag{4}$$

radial basis functions kernels (5),

$$K(x, y) = exp(-1/\delta^2 (x - y)^2) \tag{5}$$

and sigmoid functions (6) (Hochreiter & Schmidhuber, 1997)

$$f(x) = \frac{1}{1 + exp(-x)} \, . \tag{6}$$

In equation (4), $d$ is the degree of the polynomial kernel and $\delta^2$ is the bandwidth of the Gaussian radial basis function kernel in equation (5). (Kim, 2003)

In a case where there is no perfect linear discriminant to classify all data points from the training data, Cortes and Vapnik (1995) introduced the soft margin hyperplane. In that case, SVM optimizes the trade-off between the training error and the width of the margin. As a result, the sum of training errors is minimized, and the margin for correctly classified observations is maximized in a unique solution. (Cortes & Vapnik, 1995; Kim, 2003)

The advantage of SVM is its capability always to find global optimum and avoiding the overfitting problem. Kim (2003) argues that the excellent generalization capability of SVM is due to the structural risk minimization principle. This principle means that the upper bound of generalization error is minimized rather than training error. SVM has only three parameters that need to be determined. The parameters are kernel function, kernel parameter $\delta^2$ and upper bound of the generalization error. This upper bound determines the trade-off between the width of the margin and the training error. (Tay & Cao, 2001; Kim, 2003)

### 2.1.2 Artificial neural network

Artificial neural networks (ANNs) mimic the structure of human brains in their training processes (Sarle, 1994). Neural networks can be grouped into single-layer and multilayer neural networks. Single-layer networks have a single input layer and an output node, and this can be called a perceptron. The structure of a simple perceptron resembles classical linear regression. The single-layer network has as many nodes $(d)$ in the input layer as it has features or dependent variables. Predicted values in the case of a binary classification task are computed as equation (7) shows (Aggarwal 2018, 1-2, 4-6):

$$\hat{y} = sign\{\overline{W} * \bar{X} + b\} = sign\,\{\sum_{j=1}^{d} w_j x_j + b\}, \tag{7}$$

where $\overline{W}$ represents a set of weights and $\bar{X}$ represents input values. A sign function is used to convert aggregated input values to class labels. In other words, the sign function is used as an activation function. Other classical activation functions are, for instance, sigmoid, tanh function, and rectified linear unit function or their derivatives. In equation (7), $b$ represents a bias neuron, and it is used to map predicted values to the desired form. (Aggarwal 2018, 5–17)

Single $\bar{X}$ instance is fed to the network in small batches or individually one by one in a training process to produce a prediction. Weights are iteratively updated based on some error term, for example, $E(\bar{X}) = (y - \hat{y})$. However, in practice, the loss function needs to be smoothed. Learning rate is controlled through parameter $\alpha$. The algorithm loops all training observations, and weights are optimized until convergence is reached. Each data point can cycle through the system several times, and this cycle is called an epoch. (Aggarwal, 2018, 7) Gradient descent or updating of weights $\overline{W}$ can be written (Aggarwal, 2018, 7):

$$\overline{W} \Leftarrow \overline{W} + \alpha E(\bar{X})\bar{X}\,. \tag{8}$$

Multilayer networks contain one or multiple distinct computational layers, which are called hidden layers. The structure of a simple ANN without bias term is illustrated in Figure 3. Data flows forward from inputs to hidden layer or layers where computation occurs and afterward moves to the output layer. This kind of structure is called a feed-forward network. In a standard structure, all neurons are connected to all neurons from the next layer. (Aggarwal 2018, 17–18)

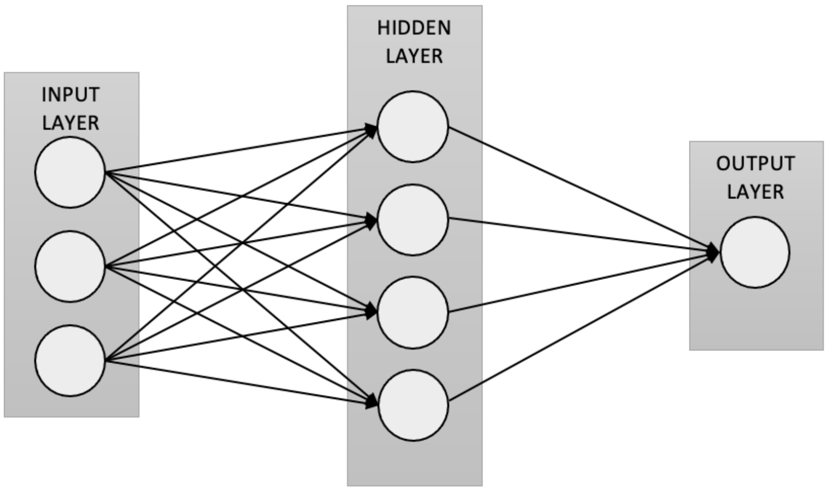


Figure 3. Structure of a simple ANN (Kara, Boyacioglu & Baykan, 2011)

Multilayer networks are trained with a backpropagation algorithm, which can be divided into two phases. The first is the forward phase, where inputs are fed to the network, and training error and derivative of the loss function are calculated. The second is the backward phase, where learning starts from the output and proceeds backward to the input layer. The gradient of the loss function for the different weights is calculated using the chain rule of differential calculus. (Aggarwal, 2018, 21) Typically, ANN is said to perform very well in generalizing arbitrary functions, but they may get into problems in the training process.

Backpropagation may cause vanishing gradient or exploding gradient problems. (Dezsi & Nistor, 2016) The vanishing gradient problem refers to a situation where the biggest product decreases exponentially, causing the error to vanish, and thus nothing is learned. Exploding vanishing problem is the opposite, and the largest error increase exponentially, causing weights to oscillate, and learning becomes unstable. (Hochreiter & Schmidhuber, 1997)

A neural network consists of several neurons that are connected processors in a single layer. Each one of the neurons produces a sequence of real-valued activations. Input neurons are activated when the environment adds inputs to the model, and weighted connections from previously activated neurons activate other neurons. (Schmidhuber, 2015)

#### 2.1.3 Random forest

Random forests (RF), which were first introduced in a study by Breiman (2001), are ensemble models because they are constructed from several decision trees. Individual decision trees are built by using only a random subset of independent variables to make classifications. These random subsets are the reason why RF can handle very well data, where are a vast number of features. Each decision tree is used to give the final class label in a training process. (Murty & Devi, 2015 144-145) Breiman (2001) showed that random forests are beneficial to use in classification and regression problems, and they also provide information of variable importance. One benefit of RF is that due to the law of large numbers, they will always converge, so the overfitting problem is avoided with low generalization error. Broader descriptions and mathematical details can be found from references. (Breiman, 2001)

### 2.2 Deep learning

Deep learning refers to a neural network with multiple processing layers, and therefore it models data with a high level of abstraction. Deep learning models extract beneficial features and complex functions of input data automatically using a general-purpose learning procedure which is the main reason for their superiority. Deep learning models are used in many fields to solve complex problems like image recognition, speech recognition, drug discovery, and genetics. Deep learning models typically require large amounts of data. The basic version of the deep learning model is a deep multilayer perceptron (DMLP) which is nothing more than a typical ANN introduced earlier, with more than one hidden layer (Figure 4). Recurrent neural networks and convolutional networks and their modifications are presented next. (Lecun, Bengio & Hinton, 2015; Sezer et al., 2020)

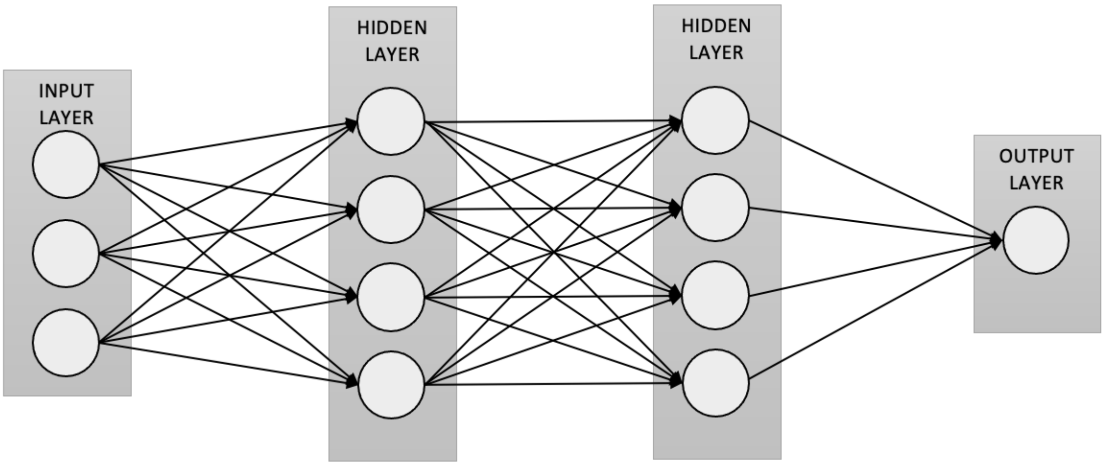


Figure 4. Example of deep learning model (Lecun et al., 2015)

### 2.2.1 Recurrent neural network

A recurrent neural network (RNN) uses sequential data such as speech or time series (Sezer et al., 2020). RNN models include a state vector in their hidden units, and this state vector consists of information about earlier elements of a sequence. In their training process, every input sequence is processed by one element at a time. (Lecun et al., 2015) The main difference between a fully connected neural network (FNN) and RNN is that RNN processes earlier and current inputs simultaneously. RNN uses internal memory in input processing, which is another difference between those models. (Sezer et al., 2020)

The computing process of RNN is the following: hidden units which are grouped under node $s$ and have values $s_t$ given time $t$, get inputs from previous time steps. This feedback and one-time step delay are presented with a black square in Figure 5. This feedback is how RNN maps elements of the input sequence $x_t$ into elements of the output sequence $o_t$. Every $o_t$ is dependent on all the previous $x'_t$ when $t' \leq t$. The same parameters, which are matrices *U, V, W*, are used for every time step.

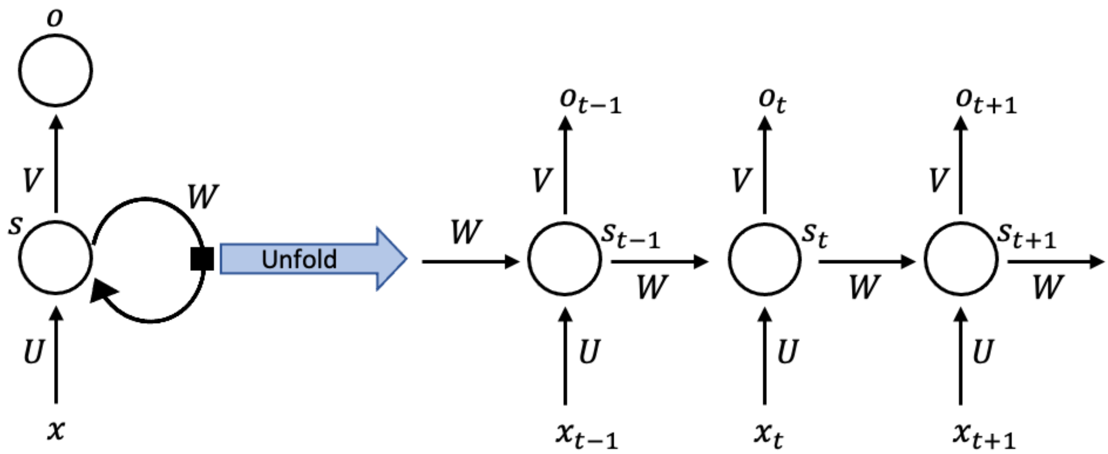


Figure 5. Forward computation of RNN and unfolding in time (Lecun et al., 2015)

The backpropagation can be used to calculate the total error of derivative for all the states $s_t$ and each parameter in a computational graph of the unfolded network. The unfolded network is presented on the right side of Figure 5. (Lecun et al., 2015) The major problem in the training process of RNN is that backpropagated gradients grow or get smaller, and when this iterates for many time steps, it causes gradients to explode or vanish (Bengio, Simard & Frasconi, 1994).

### 2.2.2 Long-Short Term Memory

Hochreiter and Schmidhuber (1997) first introduced long-short term memory (LSTM), and it is one version of RNN. Both models use sequential data, such as time series data or speech. As stated earlier, the main difference between a standard neural network and RNN is that RNN unit uses current and previous inputs simultaneously, and recurrent networks are built to model long-term dependencies. (Sezer et al., 2020)

LSTM models have an input layer, several hidden layers, and an output layer. The number of neurons in the input layer corresponds to the number of independent variables, and the output layer has two neurons in the case of the binary classification task. (Fischer & Krauss, 2018) Vanishing and exploding gradient problems cause problems in the training phase of vanilla RNN. However, the standard LSTM can overcome these problems by constant error flow through constant error carousels. (Hochreiter & Schmidhuber, 1997; Sak, Senior &

Beaufays, 2014) This improvement is gained by adding more complex units, which are memory cells in a hidden layer, and those cells are the main reason for the success of LSTM to model long-term dependencies. As illustrated in Figure 6, a memory cell includes a forget gate ($f_t$), an input gate ($i_t$), and an output gate ($o_t$), and these gates are used to adjust a cell state ($s_t$). Briefly said, the input gate controls the arrival of new information to the cell state, and the forget gate controls what information will be removed from the cell state. The output gate is used to decide what information is used as an output of the cell state. (Hochreiter & Schmidhuber, 1997; Fischer & Krauss, 2018)

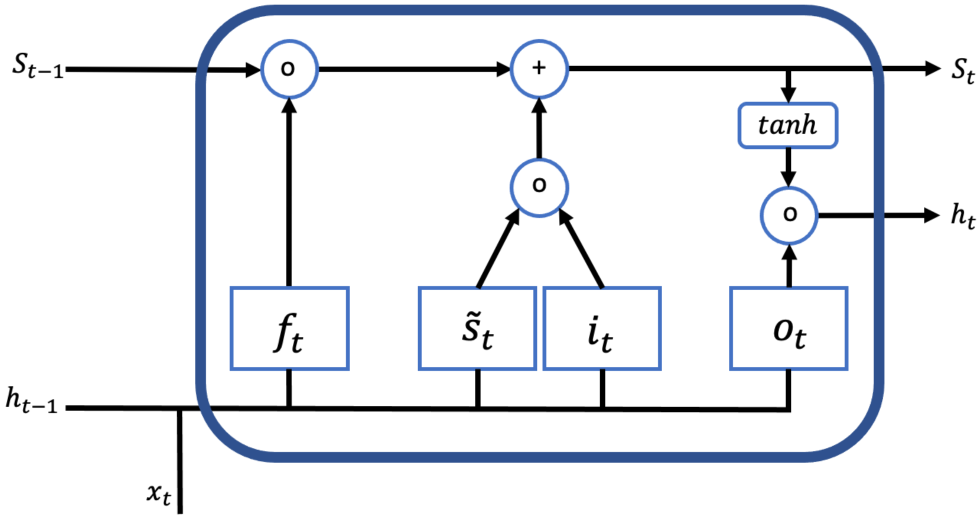


Figure 6. LSTM memory cell (Fischer & Krauss, 2018)

Next, the memory cell of LSTM is discussed in more detail by using the study of Fischer and Krauss (2018). On the left side of Figure 6 is the cell state $s_{t-1}$ and a vector of output $h_{t-1}$ from the previous memory cell and an input vector $x_t$. First, on the left side is the forget gate, which was added to the first version of LSTM in a study by Gers, Schmidhuber and Cummins (2000). There activation values $f_t$ are computed based on the input $x_t$ at timestep t and the outputs $h_{t-1}$ from the previous time step t-1. Both values are scaled with a sigmoid function in equation (9):

$$f_t = sigmoid(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f). \tag{9}$$

Values are scaled from zero to one, and zero means that information is completely removed, and one means that information is fully retained in the previous cell state $s_{t-1}$. $W$ denotes weight matrices, and $b$ stands for a bias vector. (Fischer & Krauss, 2018)

The input gate, which is the middle part of Figure 6, is discussed next. The second step of the memory cell is twofold, where the input gate decides what information should be added to the cell state $s_t$. First, candidate values $\tilde{s}_t$, which could be added to the cell state, are calculated using *tanh* function in equation (10) (Fischer & Krauss, 2018):

$$\tilde{s}_t = tanh(W_{\tilde{s},x}x_t + W_{\tilde{s},x}h_{t-1} + b_{\tilde{s}}). \tag{10}$$

The second part is to calculate activation values $i_t$ which are updated. Activation values are calculated with equation (11) (Fischer & Krauss, 2018):

$$i_t = sigmoid(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i). \tag{11}$$

Calculations from previous steps are used to update the new cell state $s_t$ in equation (12):

$$s_t = f_t \circ s_{t-1} + i_t \circ \tilde{s}_t, \tag{12}$$

where activation values $f_t$ contain information which values are forgotten, $\circ$ denotes elementwise Hadamard product, activation values $i_t$ contain information from values which will be updated and how much, and $\tilde{s}_t$ denotes the candidate values. (Fischer & Krauss, 2018)

Finally, on the right side of Figure 6, the output of the memory cell $h_t$ is calculated by using equations (13 and 14):

$$o_t = sigmoid(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o), \tag{13}$$

$$h_t = o_t \circ tanh(s_t). \tag{14}$$

Equation (14) is used to determine which parts from a cell state $s_t$ will be outputted. Then these values are multiplied with cell state $s_t$ which are first scaled to the interval (-1,1) with the *tanh* function. (Fischer & Krauss, 2018)

### 2.2.3 Convolutional neural network

A convolutional neural network (CNN) is constructed from convolutional layers based on the convolutional operation. CNN is a widely used model in classification tasks that include image or object processing. (Sezer et al., 2020) CNN takes input data in the form of multiple arrays.

CNN structure is based on four ideas, and the structure of CNN can be divided into several stages. One stack of layers comprises a convolutional layer, some nonlinear activation function, and a pooling layer. There can be many of these stacks in consecutive form. (Lecun et al., 2015)

The first idea is local connections in the first stage, which include a convolutional layer. This layer notices local similarities of features from the previous layer. Typically, pooling layers follow convolutional layers, and feature maps from the convolutional layer pass filtered information to the next layer and reduce dimensionality. Pooling is the second main idea in a CNN structure, and the object of pooling layers is to group similar features. In the case of array data, there are often local values that form a distinctive group, and these values are commonly highly correlated. The third key idea of CNN architecture is shared weights, which stems from the idea that the distinctive groups can appear in any part of the array, hence units in different parts of the array should have the same weights. The fourth idea comes from using multiple layers. CNN can be trained with backpropagation, just like a vanilla neural network. (Lecun et al., 2015)

# 3 LITERATURE REVIEW

Many literature reviews as Atsalakis and Valavanis (2009), Henrique et al. (2019), and Sezer et al. (2020) have discussed the prediction of stock prices. Authors of literature reviews have categorized research papers in many ways. The most important factors to classify papers have been the target variable (stock index or individual stocks) and is it forecasted price or direction of price. Also, the forecasting method, input data, and the forecasting period have been the classifiers. Multiple stock exchanges are predicted in previous studies, and abbreviations of the exchanges are explained in Table1. According to the literature, several different forecasting methods have been developed, and deep learning models have increased their popularity in the last years. In the following sections, the previous literature about forecasting the stock price is discussed. (Atsalakis & Valavanis, 2009)

Table 1. Abbreviations of stock exchanges

| Abbreviation | Stock exchange |
|---|---|
| GEM | Chinext Price Index |
| CSE | Colombo Stock Exchange (Sri Lanka) |
| CAC40 | Cotation Assistée en Continu (France) |
| DJIA | Dow Jones Industrial Average index |
| HSI | Hang Seng index |
| ISE | Istanbul Stock Exchange Index |
| KOSPI | Korea composite stock price index |
| NSE | National Stock Exchange of India |
| N225 | Nikkei 225 from Japan |
| Nifty 50 | NSE Nifty-50 |
| OMXH | OMX Helsinki |
| CSI 300 | Shanghai Shenzhen CSI 300 |
| SSE | Shanghai stock exchange |
| SZSE | Shenzhen stock exchange |
| S&P 500 | Standard & Poor's 500 index |
| TWSE | Taiwan Stock Exchange |
| NYSE | The New York Stock Exchange |

The three most used machine learning techniques are SVM, ANN, and RF, so chapter 3.1 focuses on those methods (Henrique et al., 2019). Chapter 3.2 will discuss studies where mainly deep learning models are used in stock market prediction. LSTM and different variations have been implemented as a primary method in many of the discussed papers. All of the cited studies are summarized in Table 2. The studies of literature review have been chosen by following studies of Atsalakis and Valavanis (2009), Henrique et al. (2019), and Sezer et al. (2020). The set of discussed studies is also complemented by selecting articles that are often referred in other studies which were selected initially.

Table 2. Summary of the cited articles

| Reference | Year | Data | Forecasting object | Variables | Method | The best performed methods | Performance measures | Frequency | Period |
|---|---|---|---|---|---|---|---|---|---|
| Althelaya et al. | 2018 | S&P 500 | price | closing prices | BLSTM, SLSTM, LSTM, MLP | 1. BLSTM 2. SLSTM 3. LSTM 4. MLP | MAE, RMSE, R^2 | daily data | 2010-2017 |
| Baek Kim | 2018 | 10 stocks of S&P 500, 10 stocks of KOSPI200 | price | closing prices | LSTM, modified LSTM, RNN, DNN | 1. Modified LSTM 2. LSTM 3. RNN 4. DNN | MAPE, MAE, MSE | daily data | 2000-2017 |
| Ballings et al. | 2015 | 5767 stocks from European market | direction | 81 financial indicators and economic variables | RF, AB, KF, SVM, ANN, LOG, KNN | 1. RF 2. SVM | AUC | yearly data | 2009-2014 |
| Bao et al. | 2017 | CSI 300, Nifty 50, HSI, N225, S&P 500, DJIA | price | OHCL, 12 technical indicators, 2 macroeconomic variables | WSAEs-LSTM, WLSTM, LSTM, RNN, buy-and-hold method | 1.WSAEs-LSTM 2. WLSTM 3. LSTM 4. RNN | MAPE, R, Theil U | daily data | 2008-2016 |
| Chen et al. | 2003 | TWSE | direction | several economic variables and index closing price | PNN, GMM, random walk | PNN | accuracy returns | monthly data | 1982-1992 |
| Chen et al. | 2015 | SSE, SZSE | 7 return categories | OHLCV | LSTM | LSTM | Accuracy | daily data | 1990-2015 |
| Chong et al. | 2017 | 38 stock returns of KOSPI | returns | stock returns | DNN, AR | DNN | NMSE, RMSE, MAE, MI | 5 minute data | 2010-2014 |
| Dezsi Nistor | 2016 | BRD stock from Romania | return | OHLC in logarithmic returns | LSTM, TARCH | 1. TARCH 2. LSTM | RMSE, MAE | daily data | 2001-2016 |
| Enke Thawornwong | 2005 | constituents of S&P 500 | direction and level | financial and economic variables | DNN, GRNN, PNN, linear regression | DNN for classification | RMSE, COR, sign of returns | monthly data | 1976-1999 |
| Fischer Krauss | 2018 | constituents of S&P 500 | direction | total return indices of closing prices | LSTM, RF, DNN, LOG | LSTM | accuracy, return (%), STD, Sharpe ratio | daily data | 1992-2015 |
| Hiransha et al. | 2018 | 3 stocks from NSE, 2 stocks from NYSE | price | closing prices | MLP, RNN, LSTM, CNN, ARIMA | 1. CNN 2. LSTM | MAPE | daily data | 1996-2017 |
| Kara et al. | 2011 | ISE | direction | 10 technical indicators | ANN, SVM | 1. ANN 2.SVM | Accuracy | daily data | 1997-2007 |
| Kim | 2003 | KOSPI | direction | 12 technical indicators | SVM, ANN, CBR | SVM | Accuracy | daily data | 1989-1998 |
| Liu Long | 2020 | S&P 500, DJIA, China Minsheng Bank stock | return | closing prices | EWT-dpLSTM-PSO-ORELM and its versions are used as a benchmark | Hybrid framework | MAE, MAPE, RMSE, SDE | daily data | 2010-2017 |
| Pai Lin | 2005 | 10 stocks from NYSE and Nasdaq | price | closing prices | ARIMA, SVM, Hybrid of ARIMA and SVM | 1. Hybrid of ARIMA and SVM, 2. SVM, 3. ARIMA | MAE, MAPE, MSE, RMSE | daily data | 2002 (3 months) |
| Samarawickrama Fernando | 2017 | 3 stocks from CSE | price | two days lagged CHL prices | LSTM, RNN, GRU, MLP | 1. MLP 2. LSTM (most often the best) 3. RNN 4. GRU | MAD, MAPE | daily data | 2002-2013 |
| Selvin et al. | 2017 | 3 stocks from NSE | price | Minute wise stock price | CNN, LSTM, RNN, ARIMA | CNN | RMSE | Minute wise data | 2014-2015 |
| Tay Cao | 2001 | 5 futures (S&P 500, CAC40, 3 government bonds) | direction | closing prices as 5 day lagged percentage differences | SVM, DNN | SVM | NMSE, MAE, DS, WDS | daily data | 1992-1999 |
| Tsantekidis et al. | 2017 | 5 stocks from OMXH | direction | high-frequency limit order book | CNN, MLP, SVM | CNN | Kohen's kappa, recall, precision | high-frequency limit order book | 2 weeks in 2010 |
| Zhang et al. | 2017 | 50 stocks from US | direction | opening prices | SFM, LSTM, AR | 1. SFM 2. LSTM 3. AR | Average square error | daily data | 2007-2016 |
| Zhang et al. | 2020 | SSE, SZSE, GEM, S&P 500, DJIA, HSI | return | closing prices | CEEMD-PCA-LSTM and different versions of it, RNN, LSTM | 1. CEEMD-PCA-LSTM 2. LSTM 3. RNN | RMSE, MAE, NMSE, DS | daily data | 2010-2018 |

### 3.1 Stock market prediction with machine learning

SVM technique is a broadly used method in stock prediction. A study by Kim (2003) is seminal research and the second-highest cited machine learning paper according to Henrique et al. (2019). Kim forecasted Korea composite stock price index with SVM, Case-Based Reasoning (CBR), and Back-propagation neural networks (BPN) methods. In CBR, Kim (2003) uses five nearest-neighbors based on Euclidean distance to retrieve relevant cases for the prediction. BPN is often referred to as ANN or MLP because, most often, they all are the same artificial neural networks that are trained with backpropagation (Atsalakis & Valavanis, 2009; Schmidhuber, 2015). The target variable is the change in price direction to up or down in the next day.

Evidence shows that SVM outperforms compared methods in the study by Kim (2003), as also Tay and Cao (2001) stated in their study. Tay and Cao (2001) compared the forecasting ability of SVM and BPN with five different futures. According to their study, SVM exceeded BPN in every performance criteria, hence SVM forecasted price direction better than BPN. Tay and Cao (2001) argued success of SVM was due to four reasons. The first is that SVM minimizes an upper bound of the generalization error instead of training error, leading to better generalization than BPN. The other reason is that there are only three parameters in SVM which have to be determined, and they are a penalty parameter, gamma, and kernel function. BPN includes much more parameters. The third significant reason is that BPN is easily stuck in a local minimum in the training section whether SVM finds a global minimum. The last reason is the tendency of BPN for overfitting. (Tay & Cao, 2001)

Pai and Lin (2005) used a hybrid ARIMA and SVM model to predict ten different stocks and a forecasting period was always one day. A one-step forecasting period was used to avoid problems of the cumulative errors from the previous forecasts. The hybrid model performed better than ARIMA or SVM models individually with all used performance measures. ARIMA model is data-oriented, and it adapts to the data structure by using lagged values and error terms. The hybrid model calculates a residual of ARIMA model, and SVM is used to estimate this residual, thus predictions of both models are merged. This study concluded that the hybrid models should be used instead of individual models allowing the use of the best features of different models. In this case, those features were ARIMA's ability to model linearity and SVM's ability to model non-linearity. (Pai & Lin, 2005)

In another study, the authors forecasted the price direction of the Istanbul Stock Exchange using ANN and SVM models and ten technical indicators. ANN was more accurate than SVM, but both models were better than earlier studies in Turkey's stock market. A polynomial function was better than a radial basis function in SVM. The authors also optimized the degree of the polynomial function, gamma constant of radial basis function, and the regularization parameter. In the case of ANN, the authors optimized the number of neurons, value of learning rate, momentum constant, and the number of iterations. Although both models proved to be efficient in forecasting stock index, it is still possible to improve classification accuracy by enhancing choosing of parameters or including macro variables, according to the authors. (Kara et al., 2011)

Chen, Leung and Daouk (2003) examined forecasting of stock index in developing markets. They forecasted the price direction of the Taiwan Stock Index. Chen et al. (2003) used a probabilistic neural network (PNN), a generalized method of moments (GMM) with a Kalman filter, and a random walk method. PNN's main difference from ANN is that PNN uses probability density functions and a Bayesian decision rule. Kalman filter is an updating method that uses current estimates based on previous estimates. In other words, the current data is added to previous estimates. GMM is a parametric estimation method that handles heteroscedasticity and serial correlation better than ordinary least-squares (OLS) regression. The benefits of GMM stems from the Hansen-White variance-covariance matrix, which is estimated from residuals of OLS. The PNN had the best performance compared to other models. The authors argued that it is partially due to the PNN's ability to handle outliers and noisy data. The trading performance was the best with the PNN forecasts, and it outperformed the buy-and-hold strategy. The authors used macro variables and technical analysis indicators as input variables. (Chen et al., 2003)

Enke and Thawornwong (2005) predicted the S&P 500 index using different neural networks, and the aim was to examine that it is more beneficial to predict the price or the direction of price. The authors used an information gain data mining analysis to choose the correct variables for neural network models. According to data mining, they used 15 out of 31 economic and financial variables. Results imply that forecasting the S&P 500 index using classification methods was the most efficient way to predict; hence, price direction was the most accurate. It proved to be the most accurate predicting method with statistic measures, and it was also the most productive method in a trading simulation. The results were also compared to linear regression and the buy-and-hold strategy. The prediction ability of the linear model was the weakest of all models. Also, the lack of calculating transaction cost needs to be noted, hence practical advantages remained unproven in the paper, although the superiority of neural networks was irrefutable. Still, the results indicated that even neural network models were not accurate in their predictions. (Enke & Thawornwong, 2005)

Ballings et al. (2015) predicted the stock price direction for the one-year timestep of 5767 stocks from the European market, and they compared ensemble methods with single classifiers models. That study differs from the majority of papers because it concerns European stock markets, and besides, there are more benchmark models than usually. The study was also one of the first papers where several ensemble methods were compared. RF, AdaBoost (AB), and Kernel factory (KF) were used as ensemble methods, and SVM, Neural network (NN), logistic regression, and K-Nearest neighbors (KNN) were used as single classifier models. AB updates weights sequentially for training data in its training procedure, and in each iteration, more weight is assigned to misclassified observations. KF randomly divides the training data into partitions which are used to train a random forest. KNN is a classification algorithm that uses k-nearest observations to predict the class of new observations. As the authors assumed, ensemble methods outperformed individual classification models because all three ensemble methods were ranked to the top four methods. RF was the best-performed method, and the second-best was SVM. RF outperformed all the other models significantly except SVM. (Ballings et al., 2015)

### 3.2 Stock market prediction with deep learning

Deep learning models have proved to be better than traditional machine learning models in many papers about financial time series forecasting (Sezer et al., 2020). Fischer and Krauss (2018) forecasted the price trend of S&P 500 index constituents from 1992 until 2015. They used four different methods: LSTM, random forest, a standard deep neural net (DNN), and logistic regression. LSTM and RF performed clearly better than the standard deep neural net and the logistic regression. LSTM outperformed all three memory-free classification methods. Only during the financial crisis, starting from 2008 until 2009, random forest outperformed LSTM. LSTM was able to produce significant daily returns, which were also statistically significant before transaction costs. After the costs, LSTM was still the most accurate forecasting model, and it generated positive returns. Nevertheless, the authors noticed that models' forecasting ability decreased during the last years of the research period, starting from 2010. (Fischer & Krauss, 2018)

Also, Baek and Kim (2018) predicted stock prices with LSTM, and the results were promising. Results improved further by using two LSTM modules. Another module was used to predict prices, and the other module was used to avoid overfitting. They used ten stocks from the Korea composite stock price index (KOSPI200) and ten stocks from the S&P 500 indices. (Baek & Kim, 2018) In 2018, Hiransha, Gopalakrishnan, Menon and Soman predicted stock prices of three stocks from the National stock exchange (NSE) of India and two stocks from the New York stock exchange (NYSE). The paper differs from other similar papers because deep learning models were trained with a single stock price data from NSE, and then the same model was used to predict other stock prices. Four of the applied deep learning models clearly outperformed ARIMA model. When deep learning models were compared, CNN was the most accurate model to predict stock prices, and it was only slightly more accurate than LSTM. (Hiransha et al., 2018) Selvin, Vinayakumar, Gopalakrishnan, Menon and Soman (2017) also used the data of one stock as training data and tested the model for all three stocks listed in NSE. Also, in that study, three deep learning models clearly outperformed ARIMA model, and of the used deep learning models, CNN was the best performer. CNN was able to adapt to changing trends in the time series, whereas RNN and LSTM used previous lags for prediction, hence the changes in the structure of the data could not be captured. (Selvin et al., 2017)

Several expansions have been suggested to LSTM, like Stacked LSTM (SLSTM) and bidirectional LSTM (BLSTM). SLSTM is an LSTM model where are stacked multiple LSTM layers. BLSTM differs from the other models in the way that it uses future and past values in its training procedure when vanilla LSTM uses only past values. In the study of Althelaya, El-Alfy and Mohammed (2018), more developed LSTM models evidenced that they were the most accurate models when forecasted the development of the S&P 500 index with short- and long-term periods. According to the authors, all three LSTM models outperformed the single-layer MLP model, while BLSTM is the most accurate. (Althelaya et al., 2018)

Liu and Long (2020) also extended vanilla LSTM but used only closing prices of three stock indices to predict a one-time step ahead. The authors introduced a complex hybrid model that consists of several parts: empirical wavelet transform (EWT), dpLSTM, particle swarm optimization (PSO), and outlier robust extreme learning machine (ORELM). EWT builds wavelets adaptively and is a signal processing technique. It is used to preprocess the input data for the LSTM layers. LSTM with dropout strategy is the main predictor of the model, and PSO is used to optimize LSTM parameters together with dropout strategy. ORELM corrects the forecasting errors of LSTM and enhances robustness and avoids overfitting problems. Errors of LSTM and corrections of ORELM are summed up and then the final prediction of the model is obtained. The hybrid model was called EWT-dpLSTM-PSO-ORELM and the model outperformed all benchmark models. (Liu & Long, 2020)

Results of Zhang et al. (2020) are consistent with the papers of Althelaya et al. (2018) and Liu and Long (2020) because the proposed hybrid model (CEEMD-PCA-LSTM) was the best model to predict the price trend of several stock indices. Standard LSTM was extended with complementary ensemble empirical mode decomposition (CEEMD) and principal component analysis (PCA). CEEMD was used for sequence smoothing and it split fluctuations in trends to intrinsic mode functions (IMF), which denoises the complex signal. Then series of IMF are processed with PCA in order to reduce dimensionality and extract only relevant features. In the next module, high-level and abstract features are used as input to the LSTM model. Finally, the predictive synthesis module combines all the predicted values to provide the final prediction. Results showed that deep learning models were more accurate in predicting developed stock markets compared to developing markets. (Zhang et al., 2020) The difference in predicting accuracy is also noted in the study of Bao et al. (2017).

In the literature of the stock price prediction, there also exist studies where the input data has included other features in addition to data of stock prices. Tsantekidis et al. (2017) used a high-frequency limit order book which consisted of 10 orders of bid and ask sides. Every observation consisted of bid or ask price and their volume. 4.5 million observations from 5

Finnish stocks were used to predict price movements to up, down, or stay in place. CNN was significantly and clearly the best model with every tested prediction horizon compared to linear SVM and single layer MLP. (Tsantekidis et al., 2017) Bao et al. (2017) forecasted six different stock indices and used technical indicators and macroeconomic variables in addition to the typical stock price data. They developed a novel deep learning model where wavelet transforms (WT), stacked autoencoders (SAEs), and LSTM were combined. WT is first used to decompose stock prices to eliminate noise. SAE is the central part of the model, and it generates deep high-level features which are afterward used in the LSTM model to make final predictions. According to results, novel WSAEs-LSTM outperformed wavelet LSTM (WLSTM), LSTM, RNN, and the buy-and-hold methods statistically significantly when measured with accuracy, error, and trading profits. The prediction performance of all the models was consistently better in developed markets compared to developing markets. (Bao et al., 2017)

Some evidence is presented where deep learning models do not outperform traditional time series forecasting methods, as Dezsi and Nistor (2016) and Chong, Han and Park (2017) argued in their studies. In the study of Chong et al. (2017) only a minor advantage of deep neural network (DNN) over autoregressive model (AR) was found with testing data. Despite the weak performance of DNN, the authors highlighted that DNN is convenient to implement in practice because it does not require preprocessing of data nor earlier knowledge of independent variables. (Chong et al., 2017) Either Chen, Zhou and Dai (2015) did not get superior results in their study where they predicted stock returns of Shanghai (SSE) and Shenzhen (SZSE) stock exchange. They categorized returns into seven classes, and the best accuracy that they achieved was 27.2 %, when the accuracy of random guess was 14.3 %. The accuracy increased due to adding features to the model, and the best results were obtained by using only stocks from SSE 180. (Chen et al., 2015) However, deep learning models have mostly outperformed traditional time series forecasting methods (Zhang, Aggarwal & Qi, 2017). Zhang et al. (2017) predicted the trend of 50 stock prices with several frequencies using AR, LSTM and novel State frequency memory (SFM) which is an extension to LSTM. SFM differs from LSTM in a way that it includes the state-frequency component for multiple frequencies. Therefore, a joint state-frequency forget gate determines how much information is kept from different frequencies. Recurrent neural network models clearly outperformed AR model, and SFM was capable of capturing better multi-frequency patterns than LSTM.

Better performance of SFM was caused by the additional joint state-frequency memory states (Zhang et al., 2017).

There are also recent studies where only different deep learning techniques have been examined. In 2017, three RNNs were used to predict the prices of three stocks in the Colombo stock exchange. Standard RNN, LSTM, and gated recurrent unit (GRU) were benchmarked with multilayer perceptron (MLP). The structure of GRU is close to LSTM, but GRU does not include an output gate. The results were conflicting with the results of prior studies where LSTM has often outperformed comparable models. MLP was the best performing model, and according to the authors, the reason for that was probably the use of only two days lags of stock prices as an input. The poor performance of GRU and remarkably high accuracies (about 99 %) of MLP and LSTM need to be noticed. (Samarawickrama & Fernando, 2018)

# 4 METHODOLOGY AND DATA

ANN, SVM, and RF have been widely used among machine learning techniques in financial time series forecasting (Henrique et al., 2019). SVM is used in this study because it has shown excellent predicting capability and has often been the best performing method among the previous studies. For instance, SVM outperformed ANN and CBR in a study by Kim (2003). Also, Tay and Cao (2001) showed that the results of SVM were better than DNN, and Pai and Lin (2005) proved that SVM outperformed ARIMA in their study. SVM always finds a global optimum, so avoiding the overfitting problem is one reason why SVM was chosen over ANN. (Cortes & Vapnik, 1995; Kim, 2003)

LSTM is the chosen deep learning method in this study because LSTM has been argued to learn long-term dependencies of time series, and LSTM has provided promising results in previous studies (Sezer et al., 2020). LSTM, or some variations of it, has been the best performing model in studies of Fischer and Krauss (2018), Dezsi and Nistor (2016), Baek and Kim (2018), and Althelaya et al. (2018), for instance. Proposed hybrid models in Zhang et al. (2020), Bao et al. (2017), and Liu and Long (2020) have outperformed single LSTM models, but the hybrid models are left outside of the scope in this study. Figure 7 illustrates how the empirical part of the study will proceed. After the raw data and statistics of it have been described, data preprocessing is introduced. Also, the implementation of the models is presented in the following subsections with the forecasting. Finally, the results of the models and a comparison of them are discussed. The preprocessing, model building, and forecasting are entirely conducted by using the Matlab programming language.


Figure 7. Progress of the empirical part

## 4.1 Description of the data

This study focuses on the stock index prediction, and the predicted indices are OMX Helsinki 25, Standard & Poor's 500, and Financial Times Stock Exchange 100. The data sets have been collected from Datastream. The OMXH25 index consists of 25 of the most traded shares on the Helsinki Stock Exchange. The maximum weight of individual stock can be 10 %. The index is rebalanced four times per year, and the index constituents are updated twice a year. Seligson & Co. OMXH25 ETF can be used in practice to trade the OMXH25 index. (Nasdaq 2021) The S&P 500 index consists of the 500 large companies which are listed in the United States markets. The committee of S&P 500 has chosen those companies to the lists, and for example, the companies which have made operation losses for many years are removed from the index. (S&P Dow Jones Indices 2021) FTSE 100 is an index for blue-chip companies based in the United Kingdom. London Stock Exchange is used to trade all FTSE 100 constituents. The index is market-capitalization weighted, and it measures the performance of the 100 largest companies from the London Stock Exchange, which need to fulfill some size and liquidity constraints. (London Stock Exchange 2021)

### 4.1.1 Data description of the OMXH25 index

Used variables are open, high, low, close (OHLC) and adjusted close prices of the index. The data sample is collected from the period 01.01.2009-30.12.2019, and the dataset contains 2759 observations after missing values are removed. Descriptive statistics of data have been presented in Table 3. Statistics of open, high, low and close prices are very close to each other, and hence they likely include similar information compared to themselves. A range between a minimum value and a maximum value in each variable is large. For instance, the minimum of open prices has been 1189.40, and the maximum of open prices has been 4386.50 in the time period. Positive skewness means that the distributions of variables have longer right tails than in normal distribution. Also, the number of smaller values dominates larger values in the case of positive skewness.

Table 3. Descriptive statistics of the OMXH25 (2009-2019)

| | Min | 25 % | Mean | Median | 75 % | Max | Skewness | Kurtosis | Std |
|---|---|---|---|---|---|---|---|---|---|
| **Open** | 1189.40 | 2217.10 | 2933.60 | 2907.90 | 3760.80 | 4386.50 | 0.03 | 1.76 | 828.25 |
| **High** | 1199.60 | 2233.60 | 2953.40 | 2923.20 | 3778.90 | 4404.00 | 0.04 | 1.76 | 829.14 |
| **Low** | 1168.30 | 2196.50 | 2912.60 | 2891.90 | 3741.30 | 4375.70 | 0.03 | 1.77 | 827.18 |
| **Close** | 1189.10 | 2214.70 | 2933.50 | 2906.00 | 3761.20 | 4387.70 | 0.04 | 1.76 | 828.69 |
| **Adj. close** | 984.09 | 2026.70 | 3164.70 | 3020.90 | 4355.70 | 5463.10 | 0.28 | 1.73 | 1247.70 |

Adjusted closing price takes into account dividends and stock splits when opening, closing, low, and high prices do not do that. That can be noticed when compared statistics of those prices. For instance, this difference is visible in maximum values of variables, and adjusted closing is about 1000 points larger than the closing price. That is why the adjusted closing price can also be called as total return index. When Figure 8 is examined, it can be noted that trend of OMXH25 has been positive during the time period. There have been some drops, but the total return index has increased its value relatively steadily.


Figure 8. Price development of the OMXH25 during 2009–2019

### 4.1.2 Data description of the S&P 500 index

The S&P 500 data is collected from the same period as the OMXH25 data, and the data cleaning process is identical to what is previously explained. Variables are also the same as used with OMXH25 data. Descriptive statistics of the S&P 500 are presented in Table 4. The difference between the minimum and the maximum value of each variable is very large. The development of the S&P 500 has been strong during 2009-2019, and the primary trend has been positive, as can be seen from Figure 9. Statistics support visual inspection, and the maximum adjusted closing price is six times greater than the minimum value. The skewness of all variables is also positive. Statistics of open, high, low, and close are very close to each other, but the adjusted closing price differs from others. It takes into account dividends, and that is why it has much larger values.

Table 4. Descriptive statistics of the S&P 500 (2009-2019)

| | Min | 25 % | Mean | Median | 75 % | Max | Skewness | Kurtosis | Std |
|---|---|---|---|---|---|---|---|---|---|
| **Open** | 679.28 | 1309.37 | 1868.98 | 1904.09 | 2366.08 | 3247.23 | 0.20 | 1.92 | 634.27 |
| **High** | 695.27 | 1316.16 | 1878.23 | 1918.40 | 2371.54 | 3247.93 | 0.20 | 1.92 | 635.17 |
| **Low** | 666.79 | 1302.42 | 1859.18 | 1885.58 | 2356.21 | 3234.37 | 0.20 | 1.92 | 632.94 |
| **Close** | 676.53 | 1309.66 | 1869.54 | 1903.82 | 2364.87 | 3240.02 | 0.20 | 1.92 | 634.07 |
| **Adj. close** | 1095.04 | 2221.12 | 3468.44 | 3497.11 | 4539.25 | 6571.03 | 0.31 | 1.96 | 1387.01 |

Figure 9 presents the development of close and adjusted close prices of the S&P 500. The slope of the adjusted close price seems to be steeper compared to OMXH25. There has been a bull market during the time period from 2009 to 2019, but the data still includes some plummets. S&P 500 and OMXH25 behave relatively similarly based on Figures 8 and 9. The Pearson correlation coefficient for the adjusted close price of S&P 500 and OMXH25 is 0.98. It means that prices move almost linearly together.

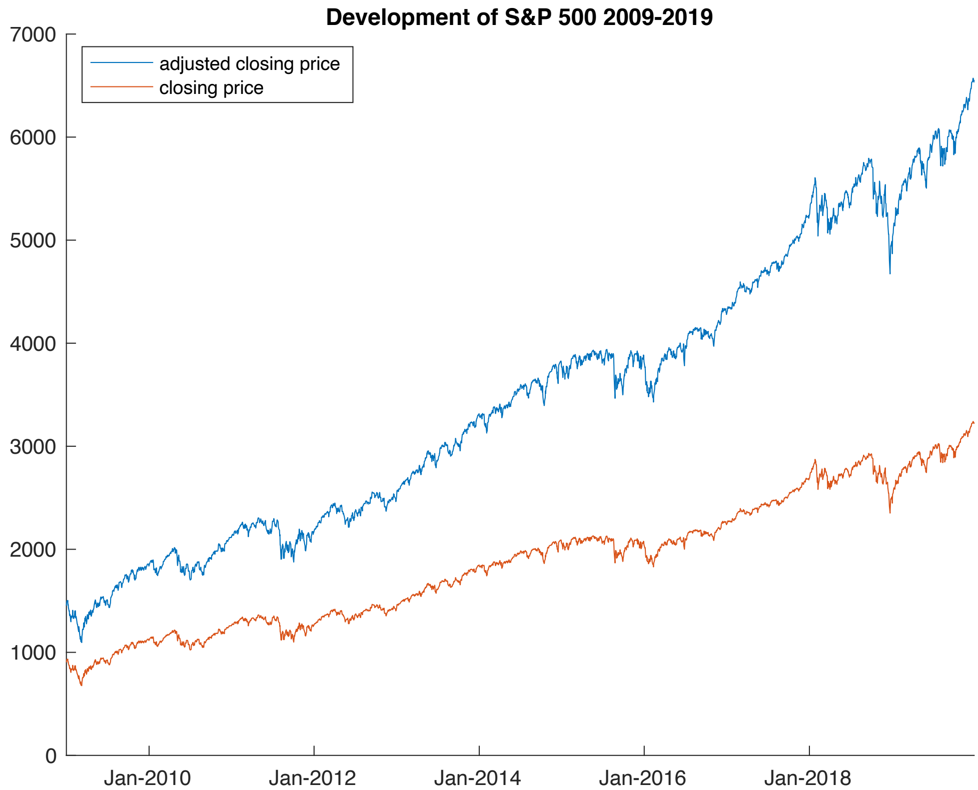


Figure 9. Price development of the S&P 500 during 2009–2019

### 4.1.3 Data description of the FTSE 100 index

Descriptive statistics of the FTSE 100 are presented in Table 5. The data preprocessing is identical to what is explained before. Index values are negatively skewed and distributed slightly platykurtic. The form of FTSE 100 distribution differs from OMXH25 and S&P 500 because those were more platykurtic, and the skewness was positive for those two.

Table 5. Descriptive statistics of the FTSE 100 (2009-2019)

| | **Min** | **25 %** | **Mean** | **Median** | **75 %** | **Max** | **Skewness** | **Kurtosis** | **Std** |
|---|---|---|---|---|---|---|---|---|---|
| **Open** | 3512.09 | 5747.91 | 6333.93 | 6486.19 | 7053.69 | 7877.45 | -0.61 | 2.87 | 898.27 |
| **High** | 3564.75 | 5782.91 | 6372.34 | 6524.46 | 7097.17 | 7903.50 | -0.61 | 2.87 | 892.73 |
| **Low** | 3460.71 | 5711.55 | 6295.63 | 6439.16 | 7021.19 | 7854.58 | -0.61 | 2.88 | 903.82 |
| **Close** | 3512.09 | 5748.57 | 6335.09 | 6487.19 | 7053.97 | 7877.45 | -0.61 | 2.88 | 897.75 |
| **Adj. close** | 2147.05 | 3866.81 | 4826.83 | 4859.15 | 5983.48 | 7075.77 | -0.02 | 2.02 | 1191.95 |

Figure 10 illustrates the price development of the FTSE 100. The adjusted closing price starts with a lower index value at the beginning of 2009 but almost reaches the closing price value at the end of 2019. Those two series move in the same direction, and the primary trend was also positive in FTSE 100 index during 2009-2019. There are some periods where the index has faced severe losses. In 2009, the index dropped due to the financial crisis, but the largest losses happened around 2016 during the Brexit voting.

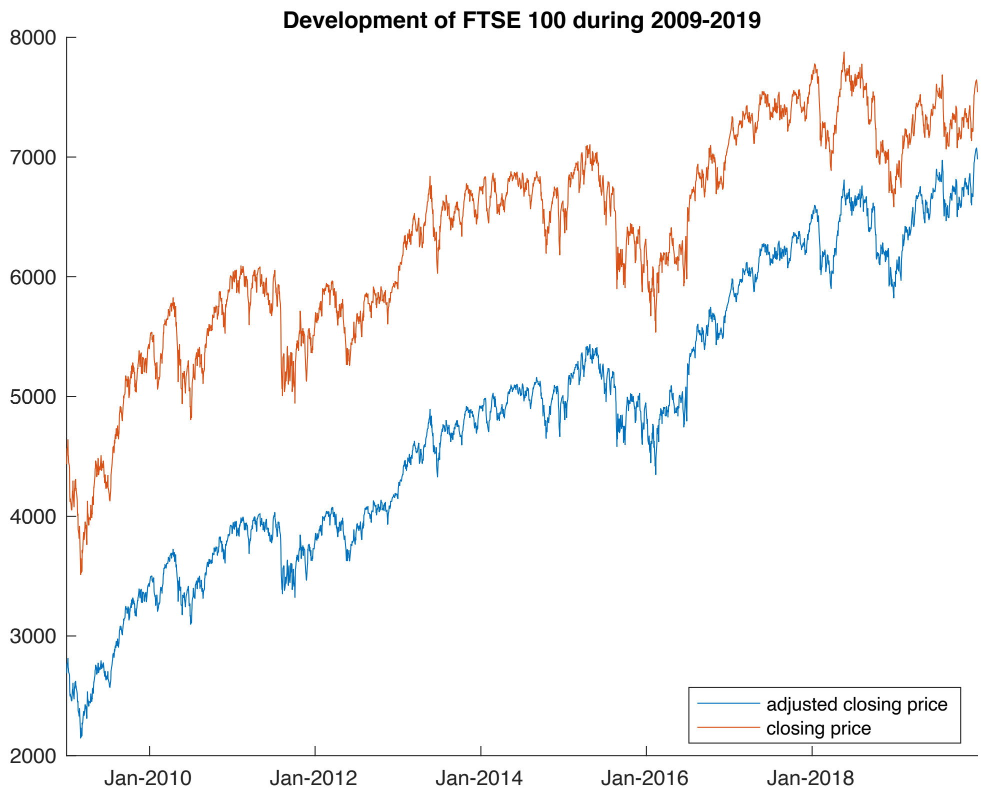


Figure 10. Price development of the FTSE 100 during 2009–2019

Pearson correlation for S&P 500 and FTSE 100 is 0.98 and 0.97 for OMXH25 and FTSE 100. Visual inspection says that the development of OMXH25 and S&P 500 are closer to each other compared to the development of FTSE 100. Still, the correlation coefficients are almost identical between all three indices.

## 4.2 Data preprocessing, implementation and training phase

The OMXH25 dataset includes 2759 observations of daily returns where 1295 returns of adjusted closing price have been negative and 1464 returns have been neutral or positive. Due to the large scale and trend of prices, the entire data is transformed into daily percentage returns. The statistics of variables in a return form are in Table 6. 25$^{th}$ percentile of adjusted closing price returns (afterward returns) has been -0.66%, and the 75$^{th}$ percentile of returns has been 0.78%, hence the majority of returns have been distributed close to zero. That can also be observed from the mean (0.06 %) and median (0.07 %). The minimum return has been -8.38% which is a large daily drop for an index. The maximum return has been 8.35%. Returns of OHLC have relatively similar statistics compared to adjusted closing returns. None of the statistics is still identical, and the standard deviation of variable high as measured in returns is smaller than that of others. The minimum and maximum of variable high are also smaller, respectively. Despite the similarity between statistics of returns, using them is justifiable because several other authors, for example, Chen et al. (2015), Dezsi and Nistor (2016), Bao et al. (2017), and Samarawickrama and Fernando (2017), have used them successfully.

Table 6. Statistics of OMXH25 returns

| | Min | 25 % | Mean | Median | 75 % | Max | Skewness | Kurtosis | Std |
|---|---|---|---|---|---|---|---|---|---|
| **Open** | -9.89 % | -0.63 % | 0.05 % | 0.06 % | 0.74 % | 7.38 % | -0.19 | 6.91 | 1.31 % |
| **High** | -5.99 % | -0.50 % | 0.04 % | 0.07 % | 0.58 % | 5.83 % | -0.07 | 6.20 | 1.10 % |
| **Low** | -7.60 % | -0.58 % | 0.05 % | 0.10 % | 0.69 % | 6.49 % | -0.18 | 6.23 | 1.23 % |
| **Close** | -8.38 % | -0.67 % | 0.05 % | 0.06 % | 0.76 % | 8.31 % | -0.01 | 6.03 | 1.32 % |
| **Adj. close** | -8.38 % | -0.66 % | 0.06 % | 0.07 % | 0.78 % | 8.35 % | -0.02 | 6.13 | 1.32 % |

Next, statistics of the S&P 500 calculated as returns (Table 7) are presented. It is worth noting that returns are negatively skewed even though the trend has been mainly positive during the sample period. That is true also with the OMXH25 data, but negative skewness is larger for the S&P 500. The excess kurtosis of the adjusted closing price is 5.06, which indicates how leptokurtic the return distribution is. Maximum and minimum values are enormous for each variable, but values inside the range between the 25$^{th}$ and 75$^{th}$ percentile are close to zero. Large absolute values in minimum and maximum columns indicate only the extreme events. Almost every minimum and maximum value of S&P 500 returns are smaller in absolute terms compared to OMXH25, respectively. The mean and median are above zero, which is not surprising because of the bull market during 2009-2019.

Table 7. Statistics of S&P 500 returns

| | Min | 25 % | Mean | Median | 75 % | Max | Skewness | Kurtosis | Std |
|---|---|---|---|---|---|---|---|---|---|
| **Open** | -6.53 % | -0.34 % | 0.05 % | 0.08 % | 0.51 % | 6.63 % | -0.28 | 7.78 | 0.97 % |
| **High** | -3.76 % | -0.29 % | 0.05 % | 0.05 % | 0.43 % | 4.37 % | -0.16 | 6.52 | 0.78 % |
| **Low** | -7.97 % | -0.37 % | 0.05 % | 0.12 % | 0.55 % | 5.09 % | -0.65 | 8.50 | 0.94 % |
| **Close** | -6.66 % | -0.35 % | 0.05 % | 0.07 % | 0.54 % | 7.08 % | -0.23 | 8.06 | 1.03 % |
| **Adj. close** | -6.65 % | -0.34 % | 0.06 % | 0.07 % | 0.54 % | 7.10 % | -0.23 | 8.06 | 1.03 % |

Statistics of FTSE 100 returns are presented in Table 8. The minimum and maximum returns are smaller in absolute terms compared to OMXH25 and S&P 500. The values are around 5% which is still a large value for a broad index. Standard deviations are under 1%, which is the smallest value from the examined indices. The excess kurtosis is again enormous, and returns are negatively skewed. Statistically, the FTSE 100 data is relatively similar compared to the OMXH25 and S&P 500 data.

Table 8. Statistics of FTSE 100 returns

| | Min | 25 % | Mean | Median | 75 % | Max | Skewness | Kurtosis | Std |
|---|---|---|---|---|---|---|---|---|---|
| **Open** | -5.33 % | -0.47 % | 0.03 % | 0.05 % | 0.56 % | 5.16 % | -0.11 | 5.77 | 0.99 % |
| **High** | -5.13 % | -0.35 % | 0.02 % | 0.02 % | 0.43 % | 4.53 % | -0.12 | 6.68 | 0.80 % |
| **Low** | -7.55 % | -0.43 % | 0.02 % | 0.09 % | 0.49 % | 4.32 % | -0.67 | 8.18 | 0.93 % |
| **Close** | -5.33 % | -0.48 % | 0.02 % | 0.05 % | 0.56 % | 5.16 % | -0.11 | 5.78 | 0.99 % |
| **Adj. close** | -5.33 % | -0.46 % | 0.04 % | 0.06 % | 0.57 % | 5.16 % | -0.12 | 5.82 | 0.99 % |

In the case of OMXH25, daily returns of adjusted closing price are presented in Figure 11. It is clearly visible that the sample period contains different volatility clusters. At the beginning of the period, there was a high volatility period in 2009 following a financial crisis, some extreme returns in 2010, and a second high volatility period was in 2011-2012 during the euro crisis. Otherwise, volatility was smaller, excluding few periods in 2015-2016. Returns seem to be heteroscedastic based on Figure 11. Heteroscedasticity means that the variability of data variates during the period, which is typical for stock returns (Chen et al., 2003).

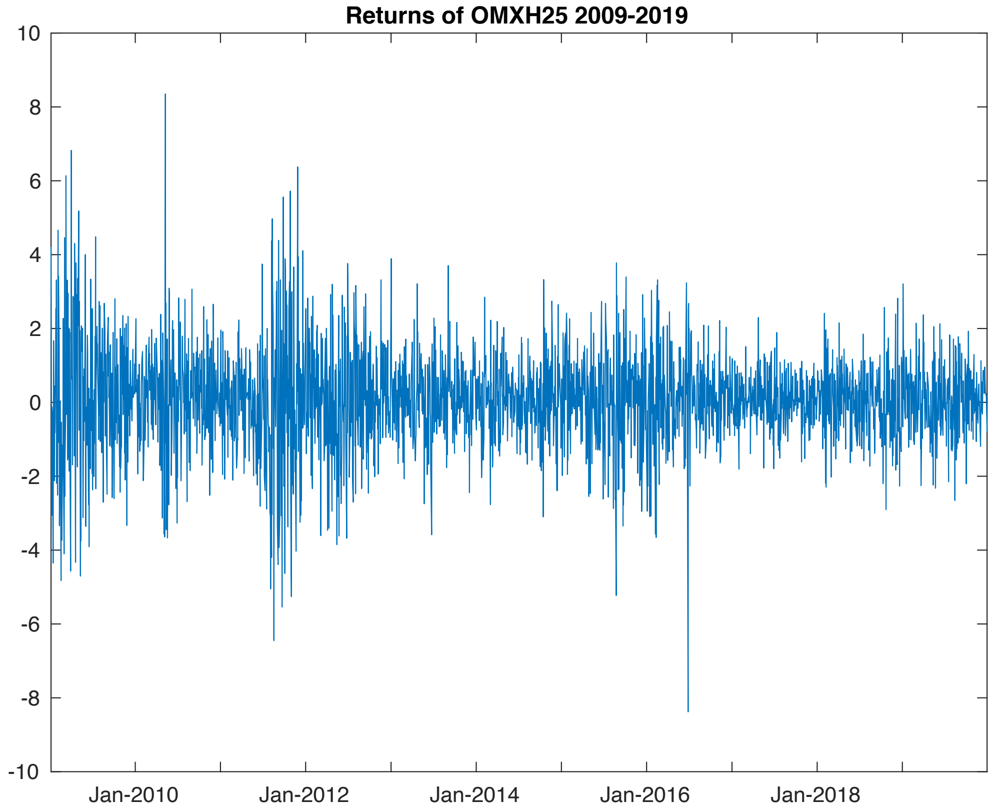


Figure 11. Daily returns of OMXH25 2009-2019

Daily returns of the S&P 500 are presented in Figure 12. Returns have behaved relatively similarly compared to returns of OMXH25. High volatility clusters have occurred at the same time with OMXH25, but volatility has been clearly larger at the beginning of 2019 in the S&P 500. It seems like S&P 500 has been more volatile during the testing data period, which may influence results. S&P 500 returns also show signs of heteroscedasticity.

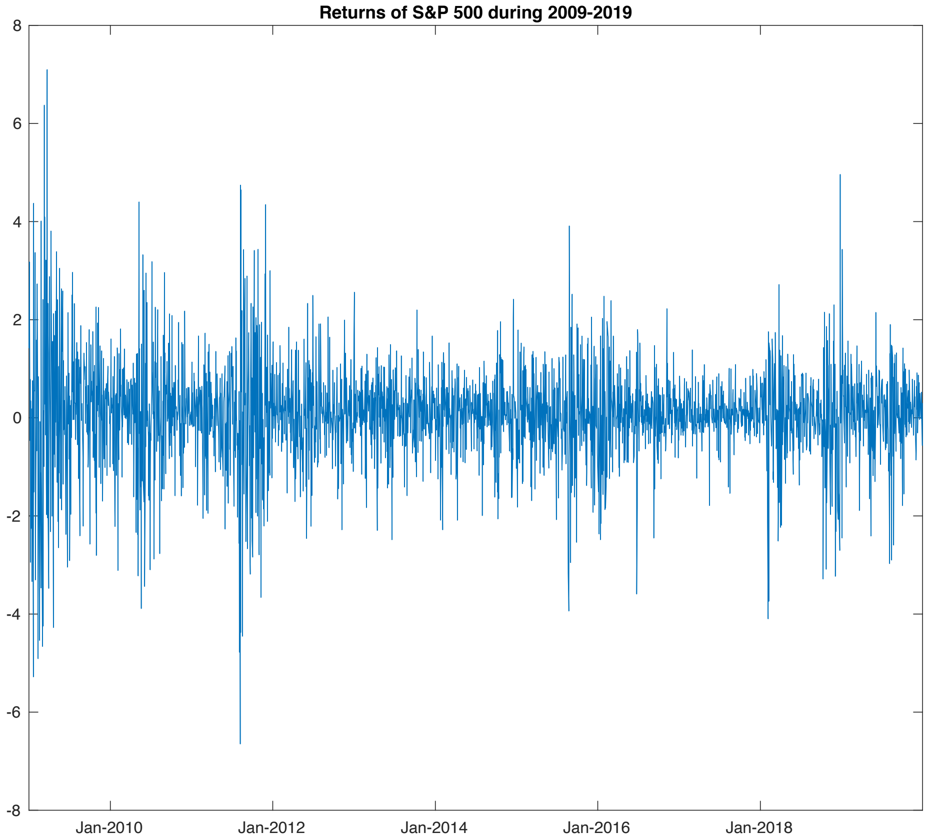


Figure 12. Daily returns of S&P 500 2009–2019

Daily returns of FTSE 100 are presented in Figure 13. Different volatility clusters are easily visible from the data. High volatilities are present at the same time when OMXH25 and S&P 500 faced high volatility periods. In other words, the dynamics of each index seem to be rather similar. The main difference between the returns of the S&P 500 and FTSE 100 is the magnitude of the returns. Around Brexit, the volatility of the FTSE 100 has been larger compared to other indices. The testing data also includes relatively large returns in absolute terms.

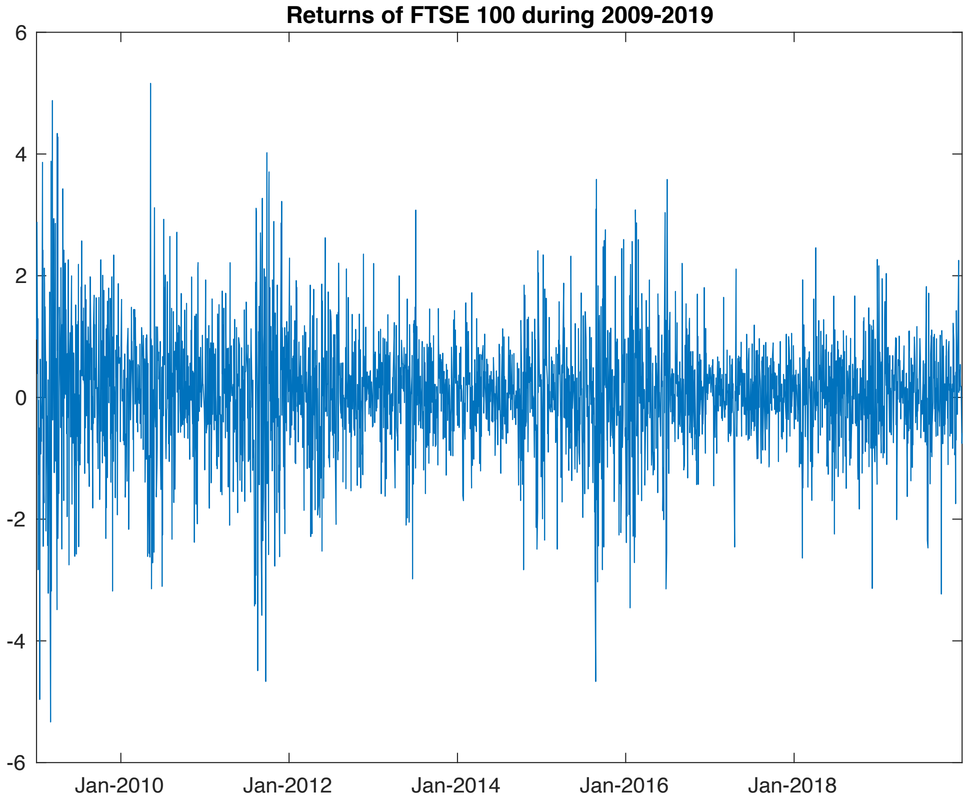


Figure 13. Daily returns of FTSE 100 2009-2019

Return distribution of OMXH25 is leptokurtic, as Figure 14 shows. Most of the returns are distributed close to zero, causing the histogram to be higher than a normal distribution. The data of the OMXH25 index responds a typical stock market data where the prices oscillate, but in the long run, the trend is positive. Due to that, the returns are basically normally distributed but positively skewed. The excess kurtosis of returns is 3.13, which is in line with the graphical interpretation. Return distribution also has several outliers, as can be seen from long tails of the histogram, and observations of high volatility periods from Figure 11 support this finding.

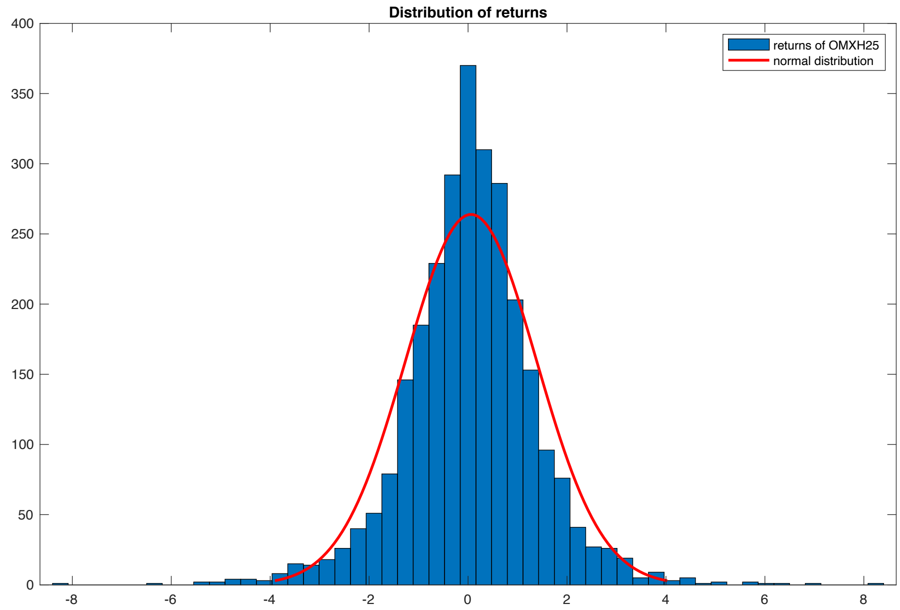


Figure 14. Distribution of OMXH25 daily returns 2009-2019

Returns of the S&P 500 are presented in Figure 15, and returns are much more leptokurtic compared to OMXH25. The excess kurtosis is 5.06, which proves the visual interpretation. Returns are heavily distributed around zero, but the tails are long. S&P 500 returns also include several extreme values or outliers equivalently to OMXH25. This is the first figure or statistic which shows a clear difference between the S&P 500 data and the OMXH25 data.

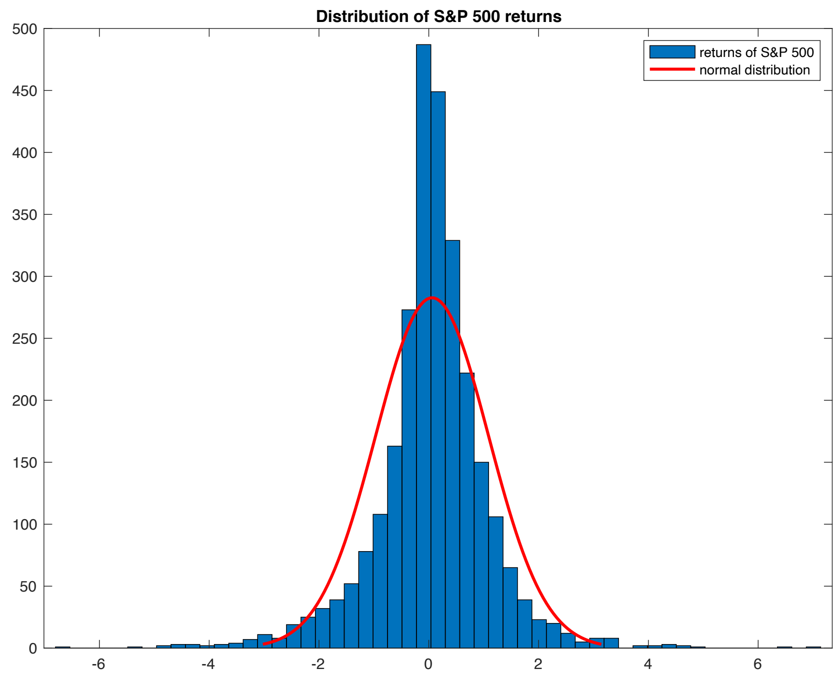


Figure 15. Distribution of S&P 500 daily returns 2009-2019

FTSE 100 return distribution and corresponding normal distribution are presented in Figure 16. Excess kurtosis is 2.82, which is the lowest of the three examined indices. Returns are leptokurtic, which is typical for stock returns, but the distribution of FTSE 100 returns is closest to the normal distribution. Most of the returns are close to zero, and in the long run, the distribution is almost normal, but the tails are heavier than in a normal distribution.

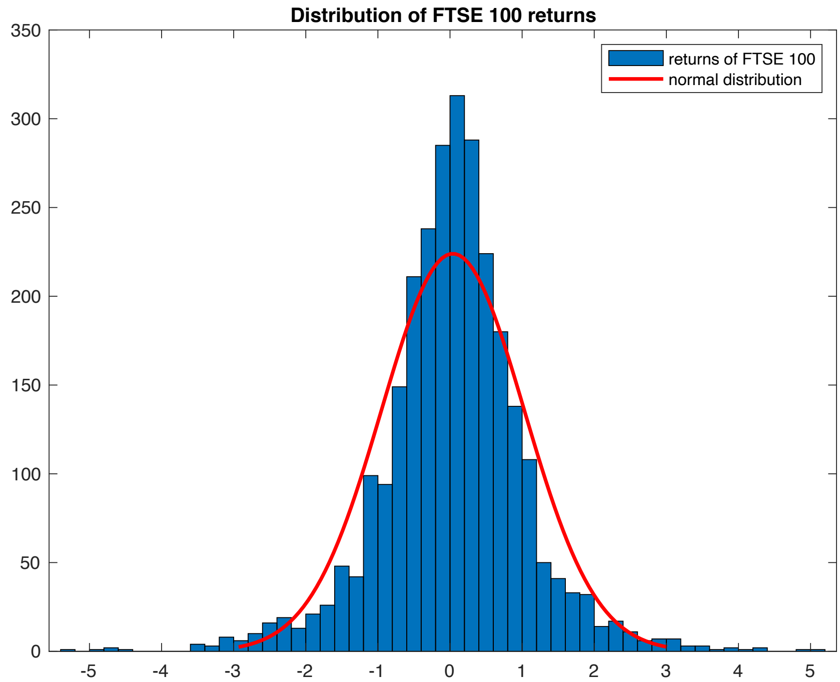


Figure 16. Distribution of FTSE 100 daily returns

Before the training phase, all datasets were standardized. The target is to transform the data to a mutual scale which makes the learning process more efficient. The input data is standardized by taking means and standard deviations from the variables and then by subtracting mean ($\mu$) from returns ($R_t$) and dividing the difference by standard deviation ($\sigma$) as follows (Fischer & Krauss, 2018):

$$\tilde{R}_t = \frac{R_t - \mu}{\sigma}. \tag{15}$$

Following a study by Bao et al. (2017), the OMXH25 data is divided into three subsets. The first 80 % of data belongs to training data from the entire dataset, the next 10 % is assigned to validation data, and the last 10 % belongs to testing data. Both models are trained with the training data and different combinations of parameters are tested with a validation set.

Based on validation results, the optimal model is selected and tested with testing data. If the final model would only fit on the training data, part of the time series was left out from the whole data. That could lead to the situation where some features of the dataset were not trained for the model. To avoid that problem, the final model is fitted on the training and validation data.

The training process is conducted only once in this study. The models are trained and optimized using OMXH25, and the same models are used to predict S&P 500 and FTSE 100 without any model manipulation between the predictions. OMXH25, S&P 500, and FTSE 100 have developed quite similarly during the sample period. The Pearson correlations are close to one for all three indices, and descriptive statistics and the previously discussed figures support the similarity of the indices. Therefore, the fitted models can be used to predict also S&P 500 and FTSE 100 testing sets. Some studies use the same idea to predict different assets with a model trained by using other data. That is done to provide more reliable and robust results. If the trained model can predict other stocks or indices accurately without retraining, it increases the trust in the model's generalization capability. Hiransha et al. (2018) trained their LSTM model with one stock from NSE data and then used the same model to predict three other stocks from NSE and two from NYSE. Selvin et al. (2017) also used the same idea and forecasted different assets after the training was conducted by using only one stock. (Hiransha et al., 2018; Selvin et al., 2017)

Parameter optimizations are explained in the following subsections. The predicted variable is a categorical binary variable that takes a value one if the daily return is equal or larger than zero. The negative return is marked with a zero.

### 4.2.1 LSTM implementation

After standardization, the input data is fed into sequences which are used as inputs for LSTM model. In Table 9, a sequence length is 30 and every sequence includes 30 daily returns of every input variable. In an example of sequence length 30, sequence one is constructed from 30 first returns of each variable, sequence two includes returns from days two to 31, and so on. Each sequence is then one input observation with 5 x 30 dimensions. In Matlab, the input data is formatted into a cell type variable. The predicted variable is a categorical binary variable, as mentioned earlier. Therefore, the target variable is the 31st daily return (1 or 0) for sequence one and 32nd for sequence two.

Table 9. Input sequences when the sequence length is 30

| Date | 1 | 2 | 3 | 4 | … | 30 | 31 | 32 |
|---|---|---|---|---|---|---|---|---|
| | | | | **Sequence 1** | | | | |
| High | 3.88 | 3.50 | 1.32 | -2.54 | … | -2.15 | | |
| Low | 2.28 | 3.64 | 1.86 | -3.58 | … | -3.75 | | |
| Open | 2.13 | 3.42 | 3.17 | -1.03 | … | -0.93 | | |
| Close | 3.39 | 3.15 | -1.02 | -2.34 | … | -3.68 | | |
| Adj. close | 3.37 | 3.13 | -1.03 | -2.36 | … | -3.69 | | |
| | | | | **Sequence 2** | | | | |
| High | | 3.50 | 1.32 | -2.54 | … | -2.15 | -2.71 | |
| Low | | 3.64 | 1.86 | -3.58 | … | -3.75 | -1.30 | |
| Open | | 3.42 | 3.17 | -1.03 | … | -0.93 | -2.89 | |
| Close | | 3.15 | -1.02 | -2.34 | … | -3.68 | -0.44 | |
| Adj. close | | 3.13 | -1.03 | -2.36 | … | -3.69 | -0.45 | |
| | | | | **Sequence 3** | | | | |
| High | | | 1.32 | -2.54 | … | -2.15 | -2.71 | -0.95 |
| Low | | | 1.86 | -3.58 | … | -3.75 | -1.30 | -0.54 |
| Open | | | 3.17 | -1.03 | … | -0.93 | -2.89 | -0.58 |
| Close | | | -1.02 | -2.34 | … | -3.68 | -0.44 | -1.04 |
| Adj. close | | | -1.03 | -2.36 | … | -3.69 | -0.45 | -1.05 |

Parameter optimization is done by switching three different parameters: the number of hidden layers, sequence length, and the number of epochs. The initial LSTM-model parameters were based on previous studies like Bao et al. (2017), Baek and Kim (2018), Zhang et al. (2020), and MathWorks (2021b-d) documentation. Those parameters did not produce adequate results because the model started to overfit the data. Training accuracy was often 100%, but the model generalized poorly because validation accuracy was low. A too complex model caused the overfitting. In order to train the model with better generalization capability, the size of parameters was reduced.

Thus, the parameter optimization set included hidden units from 80 to 140. The number of hidden units was increased by 20 units in each trial. The range of sequence length was from

10 to 50 with an increment of 5. The number of epochs ranged from 10 to 30 with an increment of ten. With those parameters, the number of different combinations was 108, and those combinations and validation results are in Appendix 1. The best validation accuracy was achieved with the combination 55, where the number of hidden units was 120, the sequence length was 30, and the number of epochs was 20. The best-achieved validation accuracy was 58.61%. The model predicted returns to be negative in 46.89% of days and positive 53.11% of days. Actual returns were 48.72% negative and 51.28% positive, so forecasts with validation data are almost correctly distributed. After the best parameters were found, those parameters are used to train the final model for predictions. As mentioned earlier, the final model is trained with a combined training and validation data, and the predictions are implemented with testing data. Let us recall that the whole parameter optimization and training are conducted by using the OMXH25 data.

### 4.2.2 SVM implementation

Identical to how is proceeded with LSTM, model training on parameter optimized is executed by using only OMXH25 data. Parameter optimization of SVM is executed by first testing three different Kernel functions with validation data. Those Kernel functions were linear, polynomial (from second to sixth degree), and Gaussian functions. Validation accuracies of those seven models are presented in Table 10. Without a closer examination, validation accuracy of a linear and polynomial (order of two, three, and six) functions yields better results than the Gaussian function.

Table 10. The first stage of SVM optimization

| Model number | Kernel function | Validation Accuracy |
|---|---|---|
| 1 | Linear | 51.81 % |
| 2 | Polynomial ^2 | 51.81 % |
| 3 | Polynomial ^3 | 48.55 % |
| 4 | Polynomial ^4 | 48.19 % |
| 5 | Polynomial ^5 | 48.19 % |
| 6 | Polynomial ^6 | 51.81 % |
| 7 | Gaussian | 49.28 % |

Validation data includes 275 daily returns, 125 (45.45%) are negative, and 150 (54.55%) are positive. SVM with the Gaussian function is the only model that even moderately classifies returns to negative or positive classes. The Gaussian kernel predicts 35.14% of returns to be negative and 64.86% positive. Other models predict 94.75-100% of daily returns to be either positive or negative, which is far from reality. An overfitting or underfitting problem may cause that kind of unbalanced prediction distribution. Models 1 and 2 predict 100% of returns to be positive and 0% of returns negative, so the model cannot predict correctly. Still, the validation accuracy for both models is 51.81% because there are 51.81% positive daily returns during the period. As a conclusion, the Gaussian kernel function was the best SVM model based on the validation data, and it is chosen to be used as a final SVM model. The results and classification distributions to negative and positive classes are illustrated in Table 11.

Table 11. Results of SVM parameter optimization with seven different kernel functions

| Model number | Negative | % | Positive | % | Validation Accuracy |
|---|---|---|---|---|---|
| 1 | 0 | 0.00 % | 276 | 100.00 % | 51.81 % |
| 2 | 0 | 0.00 % | 276 | 100.00 % | 51.81 % |
| 3 | 261 | 94.57 % | 15 | 5.43 % | 48.55 % |
| 4 | 266 | 96.38 % | 10 | 3.62 % | 48.19 % |
| 5 | 274 | 99.28 % | 2 | 0.72 % | 48.19 % |
| 6 | 0 | 0.00 % | 276 | 100.00 % | 51.81 % |
| 7 | 97 | 35.14 % | 179 | 64.86 % | 49.28 % |
| Actual values | 125 | 45.45 % | 150 | 54.55 % | - |

SVM with Gaussian kernel function was clearly the best performing model among the tested kernel functions. The next step was to optimize the rest of the parameters. The Gaussian kernel function has two parameters that can be optimized: a bandwidth of the kernel function or gamma ($\delta^2$) and a penalty term (C). This study follows a paper by Tay and Cao (2001), where those two parameters were also optimized. Tay and Cao (2001) argued that the optimal bandwidth is in a range of 1 to 100, and the optimal penalty is in the range of 10 to 100. The range of values of the penalty term was extended in this study to get a better probability of finding a global optimum for the parameter. Combinations of bandwidths in this study are in the range between 0.1 to 10. The experimented parameter value is increased with 0.1 units from 0.1 to 1 and 1 unit after parameter value 1. The penalty term ranges from 0.1 to 100 with a 0.1 unit increment until the parameter value is 1. The increment is 5 units after the parameter value is 2. These yields in a total of 570 combinations and those are presented in Appendix 2 together with the validation accuracies. The best validation accuracy was achieved with the combination number 281. When the penalty term was 2, and the bandwidth was 1, the best validation accuracy (60.22%) was achieved. Therefore, those parameters are used for the final model to predict testing data. The model with the best parameters forecasted returns to be positive in 79.04% of days, which means that the model was not able to predict negative returns effectively.

### 4.3 Model performance evaluation

The primary performance measure in this study is accuracy. The accuracies have calculated by the following equation (16):

$$accuracy = \frac{Number\ of\ correct\ predictions\ made}{Total\ number\ of\ predictions\ made} \tag{16}$$

A poorly working model may give reasonable results if the model predicts all the observations to belong to one class. The prediction distribution is also examined to avoid misleading interpretation. A confusion matrix of the models is presented to justify the performance evaluation of the models better. The matrix, illustrated in Table 12, presents how the predictions are distributed to be true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Each column in the confusion matrix represents actual classes, and rows predicted classes. (Provost & Fawcett 2013, 187-190)

Table 12. A confusion matrix (Provost & Fawcett 2013, 187-190)

| | | actual values | |
|---|---|---|---|
| | | positive | negative |
| **predicted values** | yes | TP | FP |
| | no | FN | TN |

Table 13 presents complementary statistics, which are calculated for LSTM and SVM. For instance, a true positive rate measures how often the model gives the correct prediction when the actual value has been positive. A false negative rate is the percentage of wrong predictions when the actual value has been positive. A true negative rate and a false positive rate are similar measures, but actual values have been negative in these cases. Precision tells how often the positive predictions were correct. (Provost & Fawcett 2013, 203-204)

Table 13. Calculations of evaluation metrics

| **Evaluation metrics of confusion matrix** | |
|---|---|
| True positive rate | TP/(TP+FN) |
| True negative rate | TN/(TN+FP) |
| False negative rate | FN/(FN+TP) |
| False positive rate | FP/(FP+TN) |
| Precision | TP/(TP+FP) |

Finally, the forecasting ability of LSTM and SVM is benchmarked with a random guess, following a study by Chen et al. (2015). Random guess of one-day price direction is implemented by randomizing a number from a uniform distribution for each day for the testing data. This simulation is conducted 10 000 times, and results will be averaged. The mode of each day in testing data will be reported as a prediction in results. Reported accuracy is an averaged accuracy of each 10 000 simulations. First, in each simulation of 272 days, accuracy is calculated, and an average of 10 000 simulations is reported as accuracy for a random guess.

# 5 RESULTS

The results of LSTM and SVM models will be introduced in this chapter. Also, the comparison of results is presented, and prediction performance is evaluated. Those models will also be compared with the random guess models. Parameters of the final LSTM and SVM models are also presented.

## 5.1 Results of LSTM

The self-determined parameters used in the LSTM model are presented in Table 14. The first three parameters are optimized, and the rest of the parameters are defined according to the Matlab documentations (MathWorks 2021b-d). The best validation accuracy was achieved when the maximum number of epochs was 20, the sequence length was 30, and the number of hidden units was 120 in the parameter optimization. Therefore, those parameters are also used in forecasting the testing data.

Table 14. Used LSTM parameters in Matlab (MathWorks 2021b-d)

| Matlab function | Description of Parameter | Value of parameter |
|---|---|---|
| MaxEpochs | Maximum number of epochs | 20 |
| Sequence length | Sequence length | 30 |
| Num Hidden units | Number of hidden units | 120 |
| | | |
| Solver name | Solver for training network | adam |
| Gradient threshold | Gradient threshold | 1 |
| Initial learn rate | Initial learning rate | 0.005 |
| Learn rate schedule | Option for dropping the learning rate during training | piecewise |
| Learn rate drop period | Number of epochs for dropping the learning rate | 125 |
| Learn rate drop factor | Factor for dropping the learning rate | 0.2 |
| Shuffle | Option for data shuffling | never |
| Verbose | Indicator to display training progress information | FALSE |

Because the sequence length was 30, the predicted variable was the 31st day's direction of daily return. Correspondingly, the second predicted variable was the 32nd day's sign of the return and so on.

### 5.1.1 LSTM predictions for the OMXH25 data

The model achieved 55.51% accuracy for the OMXH25 testing data predictions, which is moderately lower compared to the forecasting accuracy (58.61%) for the validation data. The predictions for negative and positive returns are distributed relatively correctly. The model predicted returns to be too often negative, and the distribution is presented in Table 15. The generalization capability of the LSTM model is appropriate because the model's performance with the testing data is close to the performance with validation data. Figures 8 and 11 illustrate the price development and the magnitude of returns. It is clearly visible that the validation data has lower volatility compared to the testing data. There are two major drops in the index in late 2018 and the beginning of 2019. According to the results and despite the more considerable volatility, the LSTM model seems to perform with the unseen testing data.

Table 15. Distribution of LSTM predictions for the OMXH25 data

| | Negative | % | Positive | % |
|---|---|---|---|---|
| **Testing data predictions** | 116 | 42.65 % | 156 | 57.35 % |
| **Actual values of testing data** | 123 | 45.22 % | 149 | 54.78 % |

Predictions of the LSTM model and the actual daily returns are illustrated in Figure 17. The bars represent actual daily returns, and if the model predicted the daily direction correctly, the color of the bar is green. If the model predicted the direction of the return incorrectly, the color of the bar is red. There are no clear observable patterns in the incorrect and correct predictions. For example, the correctness of the prediction does not seem to depend on the absolute size of the returns. There are few periods where correct predictions follow each other, and incorrect predictions follow each other, but mostly there is no visible pattern. For instance, there seem to be many subsequent correct predictions before day 100.

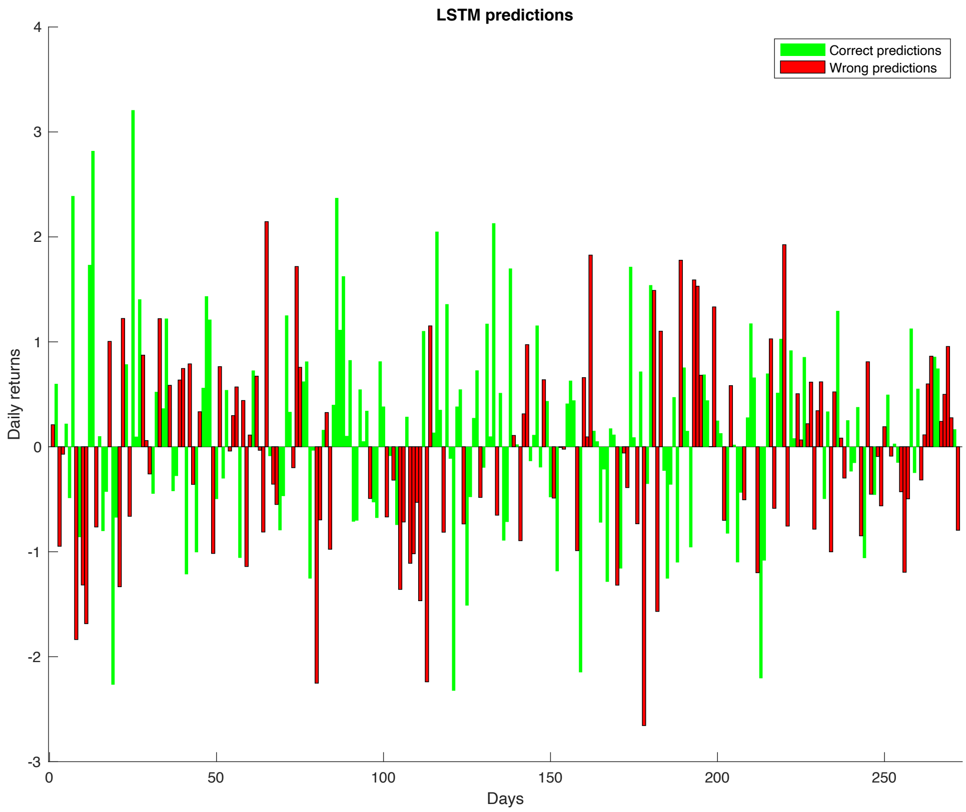


Figure 17. LSTM predictions and actual values in the OMXH25 testing set

### 5.1.2 LSTM predictions for the S&P 500 data

The trained model had a 53.31% accuracy for the S&P 500 testing data, as shown in Table 16. The testing dataset of S&P 500 includes 272 observations which are in line with other data sets used in this study. The period starts from the end of 2018 and lasts until the end of 2019. The model overweighted negative returns of the S&P 500. Predictions were distributed with 53.31% to negatives and with 46.69% to positive ones. Actually, 58.09% of the returns were positive in the S&P 500 during the examined period. 53.31% accuracy indicates that the LSTM model performs with a similar type of market data. Also, the correlation between the indices can be seen as a contributing object to the prediction performance.

Table 16. Distribution of LSTM predictions for the S&P 500 data

| | Negative | % | Positive | % |
|---|---|---|---|---|
| **Testing data predictions** | 145 | 53.31 % | 127 | 46.69 % |
| **Actual values of testing data** | 114 | 41.91 % | 158 | 58.09 % |

Figure 18 shows correct and incorrect predictions of S&P 500 returns direction. There seems to be no clear pattern of which days are predicted correctly or incorrectly. The model is capable of predicting correctly negative and positive returns. The size of the return in absolute terms seems to be also irrelevant for the model's performance because small and large returns are predicted relatively evenly correctly and incorrectly.

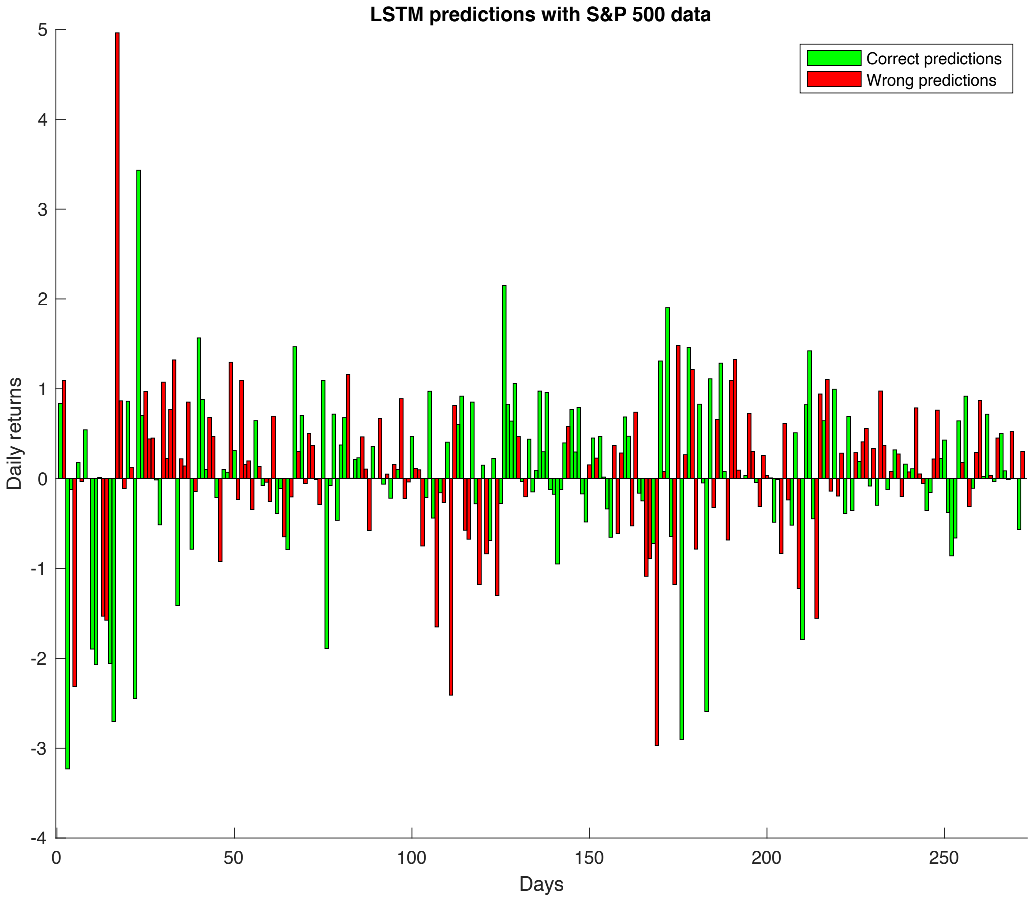


Figure 18. LSTM predictions and actual values in the S&P 500 testing set

### 5.1.3 LSTM predictions for the FTSE 100 data

The LSTM model achieved 54.41% accuracy for the FTSE 100. The distribution of predictions and actual values are presented in Table 17. Testing data also includes 272 days, and returns were positive in 53.31% of days. The model predicted 52.57% of the daily direction of returns to be positive, so predictions were almost correctly distributed. The accuracy of 54.41% is another indication that the LSTM can perform with different stock markets.

Table 17. Distribution of LSTM predictions for the FTSE 100 data

| | **Negative** | **%** | **Positive** | **%** |
|---|---|---|---|---|
| **Testing data predictions** | 129 | 47.43 % | 143 | 52.57 % |
| **Actual values of testing data** | 127 | 46.69 % | 145 | 53.31 % |

Figure 19 presents which predictions for the FTSE 100 were correct. Predictions for the FTSE 100 are in line with the previous two similar figures (17 and 18) because correct and incorrect predictions do not have a clear pattern. Large returns in absolute terms are evenly predicted correctly and incorrectly. The correct and incorrect predictions are not stacked together, even though there are some periods where predictions are correct a few times in a row.

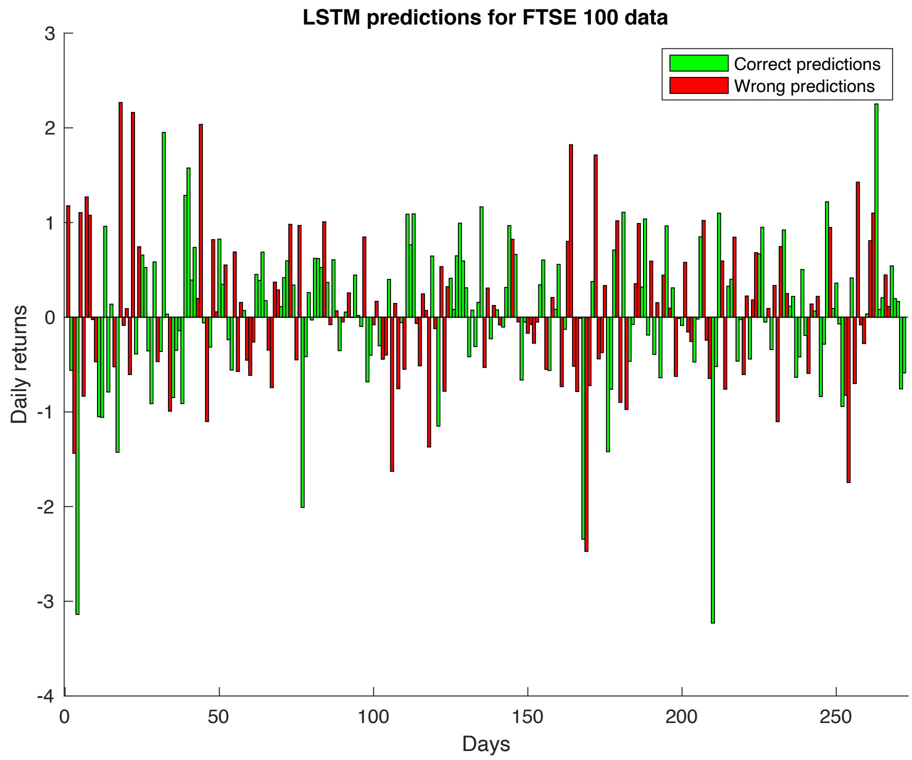


Figure 19. LSTM predictions and actual values in the FTSE 100 testing set

## 5.2 Results of SVM

The procedure with the final model selection for the SVM is the same as explained for the LSTM. The final forecasting of testing data is executed with the optimal parameters, and those are presented in Table 18. Based on the parameter optimization, the kernel function is the Gaussian function, and the penalty term is 2, and the gamma is 1.

Table 18. Used SVM parameters in Matlab (MathWorks 2021a)

| Matlab function | Description of Parameter | Value of parameter |
|---|---|---|
| KernelFunction | Used kernel function | Gaussian kernel |
| KernelScale | Bandwidth of kernel function | 1 |
| boxconstraint | Penalty term or cost (C) | 2 |

### 5.2.1 SVM predictions for the OMXH25 data

The accuracy for the OMXH25 testing data is 53.68% which is lower compared to the forecasting accuracy (60.22%) of validation data. The distribution of the SVM predictions is presented in Table 19. The model does not seem to perform well with the testing data because 79.04% of predictions are positive. Based on the results, the trained SVM model may overweight the positive returns. Returns are positive in 54.78% of days, but SVM cannot capture negative returns as a properly working model should. The generalization capability of SVM is not desirable due to the remarkable difference between the testing and validation accuracy.

Table 19. Distribution of SVM predictions in the OMXH25 testing set

| | Negative | % | Positive | % |
|---|---|---|---|---|
| **Testing data predictions** | 57 | 20.96 % | 215 | 79.04 % |
| **Actual values of testing data** | 123 | 45.22 % | 149 | 54.78 % |

The testing data is identical to the data which was used to test the LSTM model. The SVM model's training may have produced underfitting or overfitting because positive and mainly stable returns dominate the training data. If we recall that the final model was trained using combined training and validation data with optimized parameters, the primary data trend has been positive. Forecasting results support an argument that the trained SVM model predicts the trend of positive returns to remain positive. Therefore, the model is unable to predict accurately negative returns, as also Figure 20 shows. An only a small part of the bars is green in case of negative returns. Respectively, many positive returns are predicted correctly.

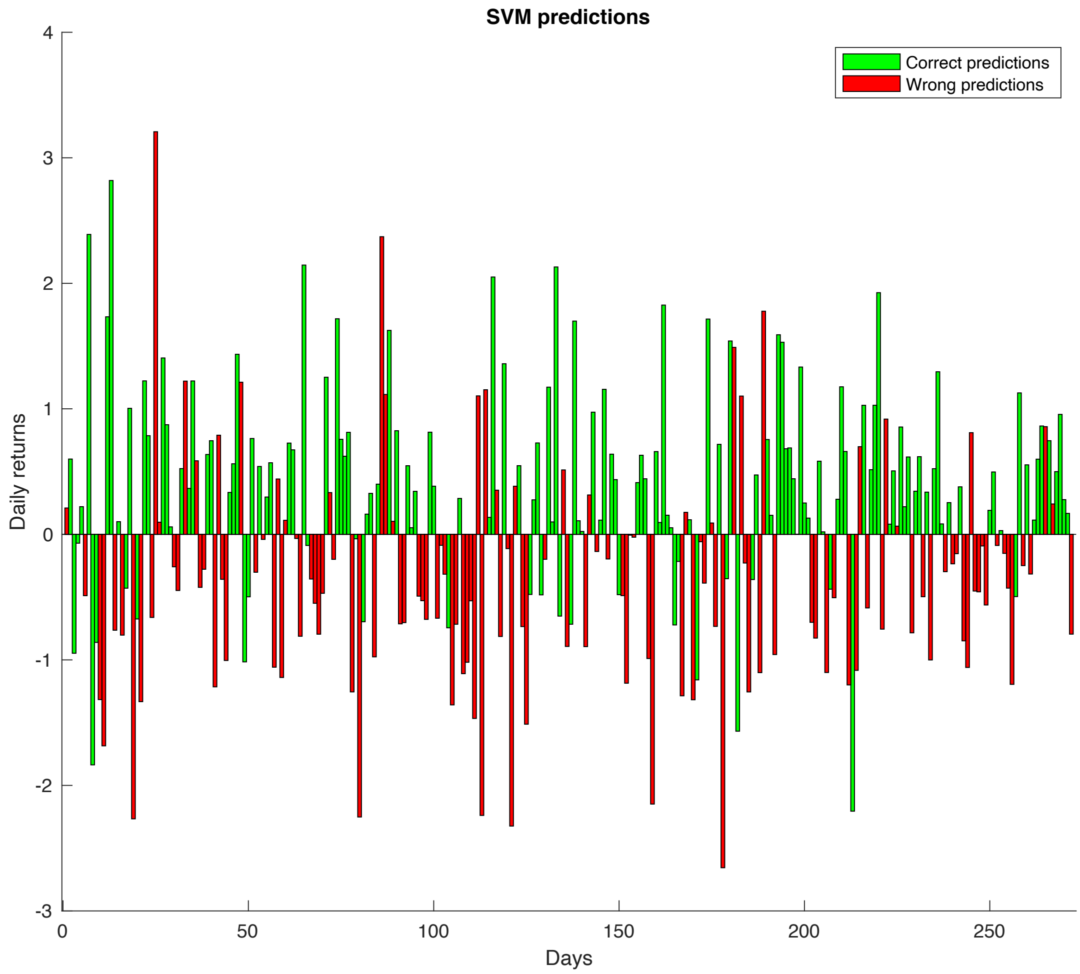


Figure 20. SVM predictions and actual values in the OMXH25 testing set

Based on the sensitivity analysis of Tay and Cao (2001), the small penalty term may potentially cause the model to underfit the data. The used penalty term (2) can be considered as a relatively small value because Tay and Cao (2001) argued that the penalty term should be in the range of 10 to 100. Small gamma should cause the opposite effect, and the gamma in the range of 0.1 to 1 may lead to overfitting. The appropriate choice should be in the range of 1 to 100, so the used value (1) is at a smaller end of the appropriate range. (Tay & Cao, 2001)

### 5.2.2 SVM predictions for the S&P 500 data

SVM predicted the S&P 500 with an accuracy of 55.15%, which is higher than the achieved accuracy (52.36%) for OMXH25. Relatively high accuracy may give a misleading interpretation about a well-performing model, but the predicting capability of SVM is rather weak. According to Table 20, SVM predicted S&P 500 returns to be negative only 22.06% of days which deteriorates the predicting performance. Actual returns were 41.91% negative, so returns were much often actually positive. Because the SVM model predicted too often positive signs for the returns, it caused the accuracy to be relatively high.

Table 20. Distribution of SVM predictions in the S&P 500 testing set

| | **Negative** | **%** | **Positive** | **%** |
|---|---|---|---|---|
| **Testing data predictions** | 60 | 22.06 % | 212 | 77.94 % |
| **Actual values of testing data** | 114 | 41.91 % | 158 | 58.09 % |

The same is visually identifiable from Figure 21. Most of the negative returns are marked with a red color, meaning incorrect prediction. The previously discussed possible overfitting problem is also true with S&P 500 data. The model seems to assume a positive trend to continue even when different stock market data is predicted.

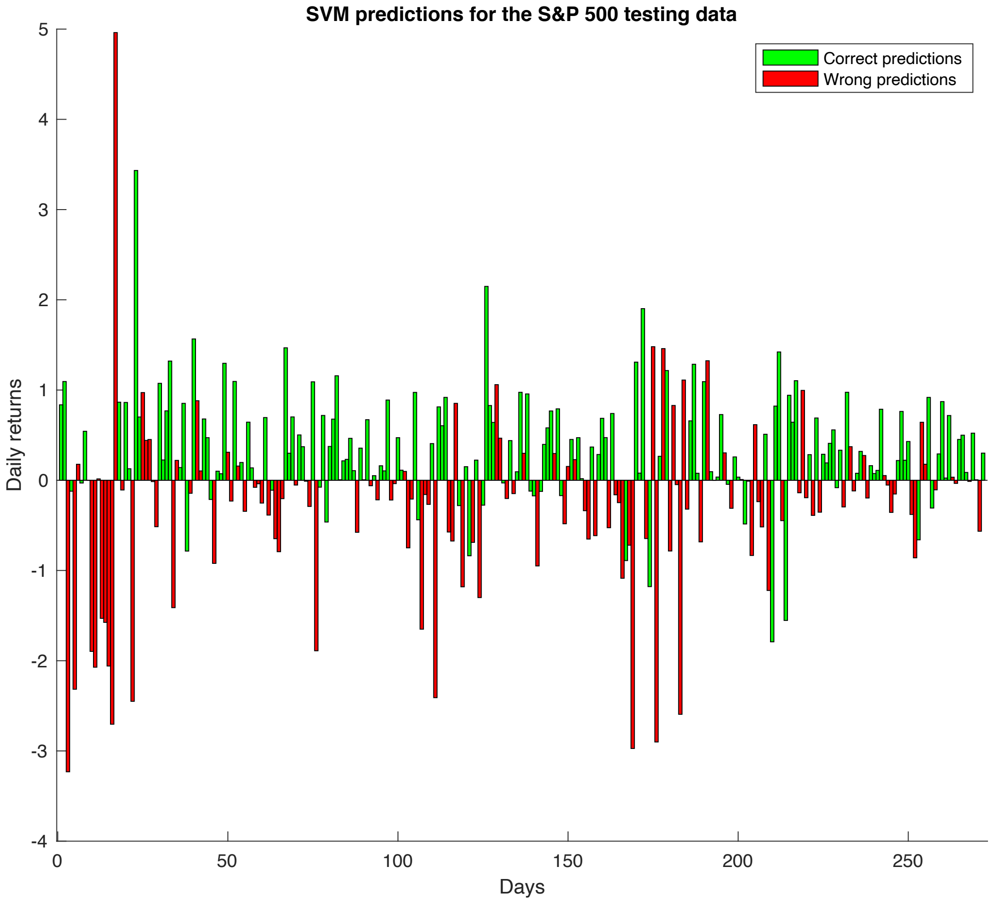


Figure 21. SVM predictions and actual values in the S&P 500 testing set

### 5.2.3 SVM predictions for the FTSE 100 data

The SVM model was able to predict the FTSE 100 with 51.84% accuracy. Interpretation of SVM results for the FTSE 100 does not differ from what is previously discussed. The model predicted returns to be negative only 24.26% of days, much lower than the actual number (46.69%) of negative returns. Distributions are presented in Table 21. Predicting accuracy is still over 50%, but the reason is the large portion of positive returns in the FTSE 100 testing data.

Table 21. Distribution of SVM predictions in the FTSE 100 testing set

| | Negative | % | Positive | % |
|---|---|---|---|---|
| **Testing data predictions** | 66 | 24.26 % | 206 | 75.74 % |
| **Actual values of testing data** | 127 | 46.69 % | 145 | 53.31 % |

Figure 22 shows the same pattern that is previously discussed about the predicting capability of SVM. The most often positive returns are predicted correctly because most predictions (75.74%) were positive. Still, especially at the beginning of the testing data, there are some negative returns correctly predicted. Despite the large number of positive predictions, there is no clear pattern of correct or incorrect predictions.

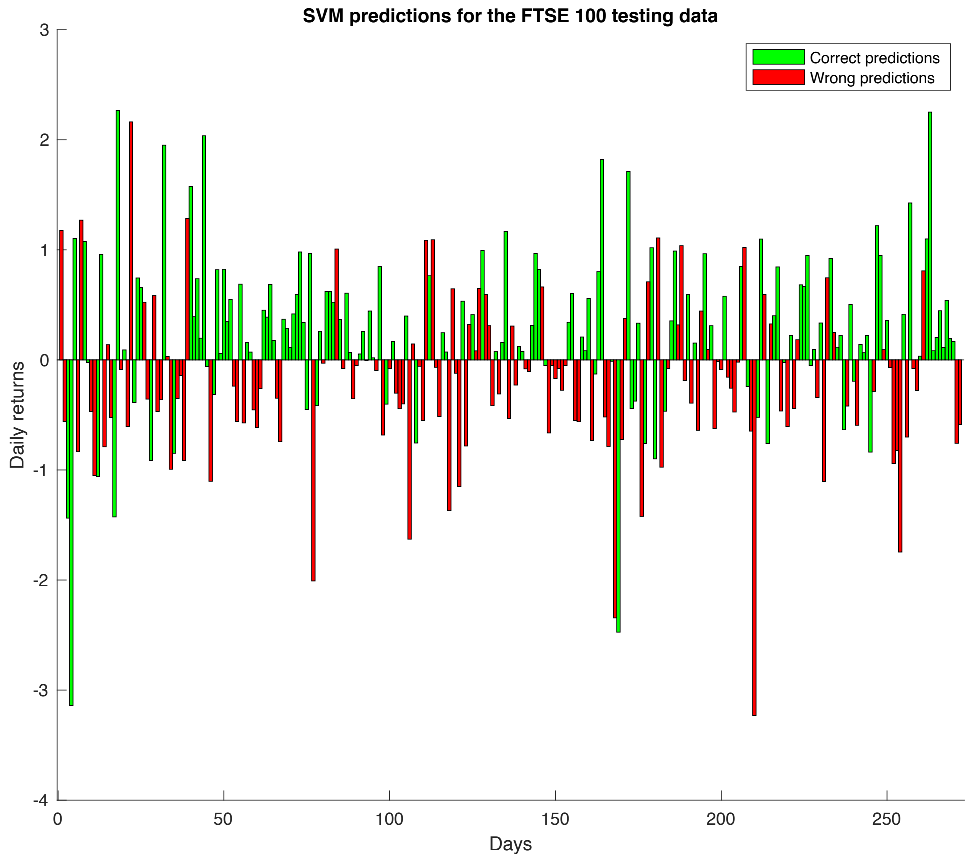


Figure 22. SVM predictions and actual values in the FTSE 100 testing set

## 5.3 Comparison of results

The actual values of the OMXH25 testing data, daily returns, and all of the predictions are presented in Appendix3. The first column shows the day of testing data (1-272), the column "returns" stands for the actual percentage returns, and "actual" means dependent binary variable. The last three columns show the predictions of each model. The column "LSTM" refers to predictions made with the final LSTM model, and the column "SVM" presents predictions with the final SVM model. The column "random guess" shows the mode of random guess calculated from 10 000 simulations. Correctly predicted target variables are presented with green color and false predictions with red, respectively. The same tables for S&P 500 and FTSE 100 are presented in Appendix 4 and Appendix 5.

### 5.3.1 OMXH25 predicting performance

The LSTM model is the most accurate (55.51%) predictor for the OMXH25 based on the results, as shown in Table 22. The LSTM model outperforms the SVM model with three percentages measured by accuracy because the accuracy of SVM is 52.36%. Both LSTM and SVM still yield better accuracies than a random guess. The accuracy of random guess is 50.01% which is almost identical to its expected value. When a sign of the next day's return is randomized, one has a 50% chance of making a right guess.

Table 22. Accuracies with the testing data of OMXH25

| Model | Accuracy |
|---|---|
| LSTM | 55.51 % |
| SVM | 52.36 % |
| Random guess | 50.01 % |

The confusion matrices of LSTM and SVM are presented in Table 23. Numbers that are in parenthesis express values of TP, TN, FP, and FN. The percentages are the rates of them. For example, the LSTM model's TN is 59, and the true negative rate is 47.97%. It is clearly visible that LSTM predictions are distributed more evenly compared to SVM. TP rate is 61.74% for LSTM, meaning that the model predicted returns to be positive when they were actually also positive. SVM has a higher TP rate because it predicted a larger portion of returns to be positive. Confusion matrices show that the LSTM model was much more efficient in predicting negative returns correctly. FP rate for LSTM is 47.97% and 21.95% for SVM, respectively. FP rate is relatively low for both models, but the FN rate (78.05%) is high for SVM. Based on confusion matrices, LSTM predicted the next day's price direction of OMXH25 more accurately and more evenly compared to SVM.

Table 23. Confusion matrix for OMXH25 predictions

| | | ACTUAL VALUES | |
|---|---|---|---|
| | | **1** | **0** |
| **LSTM PREDICTED VALUES** | **1** | 61.74% (92) | 52.03% (64) |
| | **0** | 38.26% (57) | 47.97% (59) |
| **SVM PREDICTED VALUES** | **1** | 79.87% (119) | 78.05% (96) |
| | **0** | 20.13% (30) | 21.95% (27) |

Precision for the LSTM model indicates that when the model predicted a positive direction for the next day, 58.97% was actually correct. In the case of SVM, the precision was 55.35%. That led to the conclusion that the LSTM model forecasted the positive direction better than the SVM model. TP rate can give a misleading interpretation at first sight, but the precision can be seen as a more reliable measure in this case.

### 5.3.2 S&P 500 predicting performance

The SVM model is the most accurate (55.15%) predictor for the S&P 500 based on the accuracies, as shown in Table 24. The SVM model outperforms the LSTM model almost with two percentages measured by accuracy because the accuracy of LSTM is 53.31%. Both LSTM and SVM still yield better accuracies than a random guess. The accuracy of a random guess was 50.04% in this simulation, and it is almost identical to the expected value of a random guess.

Table 24. Accuracies with the testing data of S&P 500

| Model | Accuracy |
|---|---|
| LSTM | 53.31 % |
| SVM | 55.15 % |
| Random guess | 50.04 % |

In this case, the accuracy is not serving as a good performance measure because the distribution of SVM predictions was concentrated on positive predictions. Table 25 indicates the superiority of LSTM over SVM. When LSTM predicted a negative return, the accuracy or TN rate was 57.89%. SVM achieved only 22.81% TN rate, which can be considered as a poor performance. TP rate is higher for SVM (78.48%) due to before mentioned reasons. TP rate for LSTM is 50.00%, meaning that every second time prediction was correct in the case of the model predicted positive return for the next day.

Table 25. Confusion matrix for S&P 500 predictions

| | | ACTUAL VALUES | |
|---|---|---|---|
| | | **1** | **0** |
| **LSTM PREDICTED VALUES** | **1** | 50.00% (79) | 42.11% (48) |
| | **0** | 50.00% (79) | 57.89% (66) |
| **SVM PREDICTED VALUES** | **1** | 78.48% (124) | 77.19% (88) |
| | **0** | 21.52% (34) | 22.81% (26) |

Precision is 62.20% for LSTM and 58.49% for SVM, respectively. In the same way, as discussed with OMXH25 results, precision is a more reliable measure than the TP rate. When LSTM (SVM) predicted that the next day's return is positive, it was correct in 62.2% (58.49%) cases. That is a clear signal about the superiority of LSTM over SVM. Even the accuracy of SVM was higher, LSTM outperforms SVM in its capability to predict the correct direction of S&P 500 return accurately one day ahead.

### 5.3.3 FTSE 100 predicting performance

The LSTM model achieved the highest accuracy (54.41%) for the FTSE 100 testing data. Table 26 presents the accuracies for each model. SVM is also able to outperform the random guess but only with a small margin. The SVM model predicted 51.84% of returns correctly. In this specific simulation, the accuracy for the random guess was 49.98% which is slightly lower than its expected value.

Table 26. Accuracies with the testing data of FTSE 100

| Model | Accuracy |
|---|---|
| LSTM | 54.41 % |
| SVM | 51.84 % |
| Random guess | 49.98 % |

The confusion matrices of LSTM and SVM are presented in Table 27. Again, the accuracy does not give a correct interpretation of the predicting performance of the SVM. Predictions were heavily overweighted to positive returns, and the TP rate was high (75.86%). TN rate was 24.41% which is a low value. The confusion matrix of LSTM indicates the superiority of LSTM. TP rate is 56.55% and TN rate 51.97%, so the model is able to predict positive and negative returns evenly well. FP rate and FN rate are both under 50%, which means that over 50% of both negative and positive predictions have been correct.

Table 27. Confusion matrix for FTSE 100 predictions

| | | **ACTUAL VALUES** | |
|---|---|---|---|
| | | **1** | **0** |
| **LSTM PREDICTED VALUES** | **1** | 56.55% (82) | 48.03% (61) |
| | **0** | 43.45% (63) | 51.97% (66) |
| **SVM PREDICTED VALUES** | **1** | 75.86% (110) | 75.59% (96) |
| | **0** | 24.14% (35) | 24.41% (31) |

The precision of LSTM is 57.34%, and it is higher than the precision of SVM (53.40%). The LSTM models predict positive returns more accurately than the SVM model even though the SVM model outweighs positive returns in its predictions. Precision gives a more reliable performance measure for the FTSE 100 predictions because the SVM predicted the sign of the returns to be positive too often.

# 6 CONCLUSIONS AND DISCUSSION

This study conducts stock market forecasting research using machine learning and deep learning techniques to predict a stock index's next day's price direction. The predicted stock indices were OMXH25, S&P 500, and FTSE 100. OMXH25 consists of stocks listed in Finland, S&P 500 is for US-based equities and FTSE 100 for stocks listed in the United Kingdom. The aim was to predict the direction of daily returns by using only historical price data as inputs. The used techniques were long short-term memory and support vector machine. Parameters for LSTM and SVM models were optimized in a validation phase before the final predictions, and the models were trained only for the OMXH25 data. Based on this study's results, LSTM can predict the direction of OMXH25, S&P 500, and FTSE 100 daily returns better than a random guess. LSTM demonstrated promising results and statistics about its predicting capability. SVM was not able to predict the direction of daily returns efficiently. Accuracies for SVM were relatively high, but the model overweighted positive returns. There were more positive daily returns in each testing set which caused the accuracy of the SVM model to be over 50%. The training data of OMXH25 was from a bull market where the market rose significantly. The training process of SVM faced most likely an overfitting problem, and the model predicted too often returns to be positive. Closer examination of confusion matrices revealed that the LSTM was clearly better than SVM in these three specific cases. Both models still yielded better results than a pure random guess.

The study was extended to predict S&P 500 and FTSE 100 in addition to OMXH25. The motivation for this was to enhance the reliability of the results. If a model performed well in one specific dataset, it would leave a question about how the model performs with another dataset. It was noticed in this study that parameter optimization plays a significant role to train a properly working model. It is possible that the model does not perform well in another case. Still, the results of this study prove that the predicting performance of LSTM and SVM did not significantly differ even the predicted market was changed from the OMXH25 to the S&P 500 and FTSE 100.

## 6.1 Answers to the research questions

Next, answers are provided to the research questions, and future research suggestions are discussed. The first and the main research question was:

*"How to predict stock indices using machine learning and deep learning techniques?"*

Based on the results, this study supports the evidence that machine learning and deep learning techniques provide better predictions than a random guess. Previous studies have introduced several machine learning and deep learning models which can predict stock price or price direction. In this research, the next day's price direction was predictable using LSTM and SVM, but the accuracy was lower than the best results introduced in previous studies (Zhang et al., 2020). Still, results of LSTM and SVM outperform the random guess. Based on this study, parameter optimization plays a significant role in achieving better results.

The first sub-question was:

*"How the performance of deep learning techniques and machine learning techniques differ in stock index prediction?"*

LSTM was used as a deep learning model in this study, and SVM was used as a machine learning model. LSTM outperformed SVM in its predicting capability when tested on all three stock indices and measured with multiple evaluation metrics. The accuracies of the LSTM model were 55.51% for the OMXH25 testing set, 53.31% for the S&P 500 testing set, and 54.41% for the FTSE 100 testing set. The accuracies of the SVM model were 52.36%, 55.15%, and 51.84%, respectively. The SVM model achieved higher accuracy for the S&P 500 testing set, but accuracies may give a misleading interpretation. The confusion matrices and the precisions indicate that the prediction performance of LSTM is better compared to SVM. Results evince that the generalization capability of LSTM is higher, and statistics of confusion matrices support the finding that LSTM can predict both positive and negative returns. Predictions of the SVM model were mainly positive for all testing sets, and the model was not able to predict negative results accurately. Results are in line with the majority of earlier studies.

The second sub-question was the following one:

*"How do the selected methods perform with data of OMXH25 index in the period of 2009-2019?"*

LSTM and SVM are able to provide more accurate predictions than a random guess, but the SVM model cannot predict the sign of the returns efficiently. LSTM was clearly a better model and proved to be efficient in predicting the OMXH25 testing set. The higher volatility may be the reason for the poor performance of the SVM model. The testing set of OMXH25 did not include as stable positive trend than the training data. The LSTM model performed much better with the testing set. Selvin et al. (2017) argued that even the prediction capability of LSTM decreases if there are changes in the structure of the data. Bao et al. (2017) and Zhang et al. (2020) pointed out that it is more convenient to predict developed stock markets compared to developing markets. This study supports the argument that developed stock markets are at least, to some extent, predictable.

The last sub-question was:

*"How the models, fitted with OMXH25-dataset, generalize to correlated datasets of S&P 500 and FTSE 100?"*

The predicting performance of the LSTM model did not significantly differ when the predictions were executed for the data of S&P 500 and FTSE 100. The LSTM model achieved the highest accuracy (55.51%) when the OMXH25 was predicted, but accuracies and precisions were close to each other. The SVM model did not perform well with any of the datasets, but the results remained on the same level despite the predicting object. The SVM model achieved its highest accuracy for the S&P 500 because the testing set included 58.09% positive returns.

This is the first study that compares optimized SVM and optimized LSTM in stock price direction forecasting based on the literature review. That is the main contribution of this study to stock price forecasting research. The results of this study have several similarities compared to previous research. In this study, the accuracy of LSTM is close to what Fischer and Krauss (2018) reported in their study. The highest accuracy (54.3%) of Fischer and Krauss (2018) is lower compared to the accuracy of 55.51%. It needs to be noted that the authors forecasted individual stocks, which were constituents of the S&P 500 index. Selvin et al. (2017) and Hiransha et al. (2018) trained their models with one asset and used the same model to predict other assets successfully. This study used the same idea and provided another evidence of the generalization capability of the LSTM model.

In many studies like Baek and Kim (2018), Bao et al. (2017), Fischer and Krauss (2018), and Selvin et al. (2017), deep learning models showed superiority against machine learning models. This study makes no difference to earlier findings that state-of-the-art deep learning models outperform traditional machine learning models with nonlinear and noisy stock market data. The accuracies of SVM and LSTM were highly dependent on parameter selection and optimization. Tay and Cao (2001) and Kim (2003) pointed out the importance of parameter tuning for SVM. Zhang et al. (2017) and Bao et al. (2017) also highlighted how crucial it is to select optimal parameters in the case of LSTM.

## 6.2 Limitations and future research

The main limitation of this study, as well as other similar type of studies, was the selection of the dataset. The year 2020 was not included in the sample data. A stock market crash could potentially have a significant negative effect on the predicting capability of LSTM and SVM models. There were major plummets in OMXH25 during 2009-2019, but none of them was as remarkable as a bear market due to Covid-19. This study is also limited to developed stock markets. Financial markets differ from each other, and it would be interesting to forecast Nordic markets together with developing markets in the future (Bao et al., 2017).

Some future research paths emerged based on this study. Since hybrid models were left out from this study, it could be beneficial to predict the price behavior of the Finnish stock market by using the newest modifications of deep learning models. The accuracy of LSTM was only 5.5% higher than the random guess in the case of OMXH25. When predicting the S&P 500 and FTSE 100, the improvements over the random guess were 3.27% and 4.43%. It would be beneficial to find a model which achieved even slightly better results than the basic version of LSTM. Liu and Long (2020) and Zhang et al. (2020) introduced complex hybrid models that were superior compared to the vanilla version of LSTM. In studies of Hiransha et al. (2018) and Selvin et al. (2017), the convolutional neural network was more accurate than LSTM. It would be beneficial to compare the best-performing hybrid LSTM and hybrid CNN models by performing a case study using OMXH25 data. The last-mentioned future research target could be examining how larger input sets and more comprehensive parameter optimization influence predicting accuracy. Potential inputs could be technical indicators and macroeconomic variables.

## LIST OF REFERENCES

Aggarwal, C. C. (2018) Neural Networks and Deep Learning A Textbook. Cham, Springer International Publishing.

Althelaya, K. A., El-Alfy, E. S. M. & Mohammed, S. (2018) Evaluation of Bidirectional LSTM for Short and Long-Term Stock Market Prediction. in 2018 9th International Conference on Information and Communication Systems, ICICS. Institute of Electrical and Electronics Engineers Inc., 151–156.

Atsalakis, G. S. & Valavanis, K. P. (2009) Surveying stock market forecasting techniques - Part II: Soft computing methods. *Expert Systems with Applications* 36, 3 PART 2, 5932–5941.

Baek, Y. & Kim, H. Y. (2018) ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications* 113, 457–480.

Ballings, M., Van Den Poel, D., Hespeels, N. & Gryp, R. (2015) Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications* 42, 20, 7046–7056.

Bao, W., Yue, J. & Rao, Y. (2017) A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *Plos ONE* 12, 7.

Bell, J. (2020) Machine Learning: Hands-On for Developers and Technical Professionals. 2nd ed. Indianapolis, Wiley.

Bengio, Y., Simard, P. & Frasconi, P. (1994) Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks* 5, 2, 157–166.

Bramer, M. (2020) Introduction to Data Mining. London, Springer.

Breiman, L. (2001) Random forests. *Machine Learning* 45, 1, 5–32.

Burges, C. J. C. (1998) A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, 2, 121–167.

Chen, A. S., Leung, M. T. & Daouk, H. (2003) Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan Stock Index. *Computers and Operations Research* 30, 6, 901–923.

Chen, K., Zhou, Y. & Dai, F. (2015) A LSTM-based method for stock returns prediction: A case study of China stock market. in Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015. Institute of Electrical and Electronics Engineers Inc., 2823–2824.

Chong, E., Han, C. & Park, F. C. (2017) Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications* 83, 187–205.

Cortes, C. & Vapnik, V. (1995) Support-vector networks. *Machine Learning* 20, 3, 273–297.

Dey, A. (2016) Machine Learning Algorithms: A Review. I*nternational Journal of Computer Science and Information Technologies* 7, 3, 1174–1179.

Dezsi, E. & Nistor, I. A. (2016) Can Deep Machine Learning Outsmart The Market? A Comparison Between Econometric Modelling And Long- Short Term Memory. *Romanian Economic Business Review* 11, 4.1, 54–73.

Enke, D. & Thawornwong, S. (2005) The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications* 29, 927–940.

Ertel, W. (2011) Introduction to Artificial Intelligence. London, Springer.

Evgeniou, T., Pontil, M. & Poggio, T. (2000) Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics* 13, 1, 1–50.

Fama, E. F. (1965) The Behaviour of Stock Market Prices. *The Journal Of Business* 38, 1, 34–105.

Fama, E. F. (1970) Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance* 25, 2, 383–417.

Fama, E. F. (1991) Efficient Capital Markets: II. *The Journal of Finance* 46, 5, 1575–1617.

Fischer, T. & Krauss, C. (2018) Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research* 270, 2, 654–669.

Gers, F. A., Schmidhuber, J. & Cummins, F. (2000) Learning to Forget: Continual Prediction with LSTM. *Neural Computation* 12, 10, 2451–2471.

Hellstörm, T. & Holmströmm, K. (1998) Predictable Patterns in Stock Returns. Technical Report Series IMa-TOM-1997-09.

Henrique, B. M., Sobreiro, V. A. & Kimura, H. (2019) Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications* 124, 226–251.

Hiransha, M., Gopalakrishnan, E. A., Menon, V. K. & Soman, K. P. (2018) NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science* 132, 1351–1362.

Hochreiter, S. & Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Computation* 9, 8, 1735–1780.

Huang, W., Nakamori, Y. & Wang, S.-Y. (2005) Forecasting stock market movement direction with support vector machine. *Computers & Operations Research* 32, 2513–2522.

Kara, Y., Boyacioglu, M. A. & Baykan, Ö. K. (2011) Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications* 38, 5, 5311–5319.

Kim, K. J. (2003) Financial time series forecasting using support vector machines. *Neurocomputing* 55, 1–2, 307–319.

Kumar, M. & Thenmozhi, M. (2006) Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest. Indian Institute of Capital Markets 9th Capital Markets Conference Paper. SSRN Electronic Journal.

Lecun, Y., Bengio, Y. & Hinton, G. (2015) Deep learning. *Nature (London)* 521, 7553, 436–444.

Liu, H. & Long, Z. (2020) An improved deep learning model for predicting stock market price time series. *Digital Signal Processing*, 102. 102741.

London Stock Exchange (2021) FTSE indices. Index Story. A leading measure for UK-listen blue-chip companies since 1984. [www document]. [Accessed 21 May, 2021] Available https://www.londonstockexchange.com/indices?tab=index-story

MathWorks (2021a) Fitcsvm. MathWorks Nordic. [www document]. [Accessed 10 May, 2021] Available https://se.mathworks.com/help/stats/fitcsvm.html

MathWorks (2021b) Long Short-Term Memory Networks. MathWorks Nordic. [www document]. [Accessed 24 April, 2021] Available https://se.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html

MathWorks (2021c) Sequence Classification Using Deep Learning. MathWorks Nordic. [www document]. [Accessed 24 April, 2021] Available https://se.mathworks.com/help/deeplearning/ug/classify-sequence-data-using-lstm-networks.html

MathWorks (2021d) Time Series Forecasting Using Deep Learning. MathWorks Nordic. [www document]. [Accessed 24 April, 2021] Available https://se.mathworks.com/help/deeplearning/ug/time-series-forecasting-using-deep-learning.html

Murty, M. N. & Devi, V. S. (2015) Introduction to pattern recognition and machine learning. World Scientific. New Jersey, World Scientific.

Nasdaq (2021) OMX Helsinki 25 (OMXH25). [www document]. [Accessed 02 March, 2021] Available https://indexes.nasdaqomx.com/Index/Overview/OMXH25

Pai, P. F. & Lin, C. S. (2005) A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega* 33, 6, 497–505.

Provost, F. & Fawcett, T. (2013) Data Science for Business: what you need to know about data mining and data-analytic thinking. Sebastopol, O'REILLY.

Sak, H., Senior, A. & Beaufays, F. (2014) Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv abs: 1402.1128

Samarawickrama, A. J. P. & Fernando, T. G. I. (2017) A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. 2017 IEEE International Conference on Industrial and Information Systems, ICIIS 2017 - Proceedings. Institute of Electrical and Electronics Engineers Inc., 1–6.

Samuel, A. L. (1959) Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3, 3, 210–229.

Sarle, W. S. (1994) Neural Networks and Statistical Models. Proceedings of the Nineteenth Annual SAS Users Group International Conference. SAS Institute, USA, 1538-1550.

Schmidhuber, J. (2015) Deep Learning in neural networks: An overview. *Neural Networks* 61, 85–117.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V.K. & Soman, K. P. (2017) Stock price prediction using LSTM, RNN and CNN-sliding window model. 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017. Institute of Electrical and Electronics Engineers Inc., 1643–1647.

Sezer, O. B., Gudelek, M. U. & Ozbayoglu, A. M. (2020) Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing Journal* 90, 106181.

S&P 500 Dow Jones Indices (2021) S&P 500 ®. Overview. [www document]. [Accessed 24 April, 2021] Available https://www.spglobal.com/spdji/en/indices/equity/sp-500/#

Tay, F. E. H. & Cao, L. (2001) Application of support vector machines in financial time series forecasting. *Omega* 29, 4, 309–317.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M. & Iosifidis, M. (2017) Forecasting stock prices from the limit order book using convolutional neural networks. Proceedings - 2017 IEEE 19th Conference on Business Informatics, CBI 2017. Institute of Electrical and Electronics Engineers Inc., 7–12.

Turing, A. M. (1950) Computing machinery and intelligence. *MIND 59*, 239, 433–460.

Weng, B., Ahmed, M. A. & Megahed, F. M. (2017) Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications* 79, 153–163.

Zhang, L., Aggarwal, C. & Qi, G. J. (2017) Stock price prediction via discovering multi-frequency trading patterns. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: Association for Computing Machinery, 2141–2149.

Zhang, Y., Yan, B. & Aasma, M. (2020) A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM. *Expert Systems with Applications* 159, 113609.

# APPENDICES

Appendix 1. Results of LSTM parameter optimization

| | PARAMETER OPTIMIZATION | | | RESULTS | |
|---|---|---|---|---|---|
| combination | hidden units | sequence length | number of epochs | validation accuracy | training accuracy |
| 1 | 80 | 10 | 10 | 51.64 % | 57.81 % |
| 2 | 100 | 10 | 10 | 52.36 % | 57.03 % |
| 3 | 120 | 10 | 10 | 53.45 % | 53.91 % |
| 4 | 140 | 10 | 10 | 52.73 % | 57.03 % |
| 5 | 80 | 15 | 10 | 50.73 % | 56.25 % |
| 6 | 100 | 15 | 10 | 54.74 % | 52.34 % |
| 7 | 120 | 15 | 10 | 54.01 % | 54.69 % |
| 8 | 140 | 15 | 10 | 52.19 % | 60.16 % |
| 9 | 80 | 20 | 10 | 54.74 % | 53.91 % |
| 10 | 100 | 20 | 10 | 52.19 % | 54.69 % |
| 11 | 120 | 20 | 10 | 53.28 % | 57.03 % |
| 12 | 140 | 20 | 10 | 54.74 % | 59.38 % |
| 13 | 80 | 25 | 10 | 51.65 % | 59.38 % |
| 14 | 100 | 25 | 10 | 54.58 % | 59.38 % |
| 15 | 120 | 25 | 10 | 54.21 % | 58.59 % |
| 16 | 140 | 25 | 10 | 53.48 % | 60.94 % |
| 17 | 80 | 30 | 10 | 52.01 % | 54.69 % |
| 18 | 100 | 30 | 10 | 53.48 % | 54.69 % |
| 19 | 120 | 30 | 10 | 53.48 % | 56.25 % |
| 20 | 140 | 30 | 10 | 51.28 % | 57.03 % |
| 21 | 80 | 35 | 10 | 54.41 % | 57.81 % |
| 22 | 100 | 35 | 10 | 51.84 % | 57.81 % |
| 23 | 120 | 35 | 10 | 54.04 % | 59.38 % |
| 24 | 140 | 35 | 10 | 54.04 % | 58.59 % |
| 25 | 80 | 40 | 10 | 52.94 % | 53.13 % |
| 26 | 100 | 40 | 10 | 52.21 % | 57.03 % |
| 27 | 120 | 40 | 10 | 51.47 % | 57.03 % |
| 28 | 140 | 40 | 10 | 51.47 % | 58.59 % |
| 29 | 80 | 45 | 10 | 52.40 % | 61.72 % |
| 30 | 100 | 45 | 10 | 52.40 % | 58.59 % |

| | | | | | |
|---|---|---|---|---|---|
| 31 | 120 | 45 | 10 | 54.24 % | 53.91 % |
| 32 | 140 | 45 | 10 | 54.61 % | 58.59 % |
| 33 | 80 | 50 | 10 | 50.92 % | 52.34 % |
| 34 | 100 | 50 | 10 | 52.40 % | 58.59 % |
| 35 | 120 | 50 | 10 | 52.03 % | 57.81 % |
| 36 | 140 | 50 | 10 | 54.24 % | 60.94 % |
| 37 | 80 | 10 | 20 | 53.09 % | 60.94 % |
| 38 | 100 | 10 | 20 | 48.73 % | 56.25 % |
| 39 | 120 | 10 | 20 | 50.18 % | 56.25 % |
| 40 | 140 | 10 | 20 | 48.00 % | 52.34 % |
| 41 | 80 | 15 | 20 | 53.65 % | 57.81 % |
| 42 | 100 | 15 | 20 | 51.82 % | 56.25 % |
| 43 | 120 | 15 | 20 | 51.46 % | 59.38 % |
| 44 | 140 | 15 | 20 | 54.01 % | 58.59 % |
| 45 | 80 | 20 | 20 | 47.81 % | 58.59 % |
| 46 | 100 | 20 | 20 | 50.36 % | 60.94 % |
| 47 | 120 | 20 | 20 | 56.20 % | 57.81 % |
| 48 | 140 | 20 | 20 | 53.65 % | 58.59 % |
| 49 | 80 | 25 | 20 | 50.92 % | 62.50 % |
| 50 | 100 | 25 | 20 | 53.85 % | 59.38 % |
| 51 | 120 | 25 | 20 | 53.11 % | 61.72 % |
| 52 | 140 | 25 | 20 | 52.75 % | 62.50 % |
| 53 | 80 | 30 | 20 | 53.85 % | 59.38 % |
| 54 | 100 | 30 | 20 | 54.21 % | 55.47 % |
| 55 | 120 | 30 | 20 | 58.61 % | 53.91 % |
| 56 | 140 | 30 | 20 | 53.11 % | 60.16 % |
| 57 | 80 | 35 | 20 | 50.74 % | 61.72 % |
| 58 | 100 | 35 | 20 | 52.94 % | 53.91 % |
| 59 | 120 | 35 | 20 | 50.37 % | 57.81 % |
| 60 | 140 | 35 | 20 | 51.10 % | 57.81 % |
| 61 | 80 | 40 | 20 | 52.94 % | 54.69 % |
| 62 | 100 | 40 | 20 | 54.04 % | 53.91 % |
| 63 | 120 | 40 | 20 | 53.68 % | 60.94 % |
| 64 | 140 | 40 | 20 | 55.15 % | 63.28 % |
| 65 | 80 | 45 | 20 | 53.14 % | 61.72 % |
| 66 | 100 | 45 | 20 | 53.51 % | 60.94 % |
| 67 | 120 | 45 | 20 | 50.55 % | 53.13 % |
| 68 | 140 | 45 | 20 | 54.24 % | 60.16 % |
| 69 | 80 | 50 | 20 | 53.14 % | 55.47 % |
| 70 | 100 | 50 | 20 | 54.98 % | 58.59 % |

| | | | | | |
|---|---|---|---|---|---|
| 71 | 120 | 50 | 20 | 54.24 % | 58.59 % |
| 72 | 140 | 50 | 20 | 54.98 % | 60.94 % |
| 73 | 80 | 10 | 30 | 53.09 % | 61.72 % |
| 74 | 100 | 10 | 30 | 50.18 % | 68.75 % |
| 75 | 120 | 10 | 30 | 45.82 % | 66.41 % |
| 76 | 140 | 10 | 30 | 51.27 % | 67.97 % |
| 77 | 80 | 15 | 30 | 48.91 % | 64.06 % |
| 78 | 100 | 15 | 30 | 50.00 % | 67.97 % |
| 79 | 120 | 15 | 30 | 57.66 % | 67.19 % |
| 80 | 140 | 15 | 30 | 53.28 % | 69.53 % |
| 81 | 80 | 20 | 30 | 45.26 % | 71.09 % |
| 82 | 100 | 20 | 30 | 52.55 % | 68.75 % |
| 83 | 120 | 20 | 30 | 48.91 % | 68.75 % |
| 84 | 140 | 20 | 30 | 50.73 % | 74.22 % |
| 85 | 80 | 25 | 30 | 50.18 % | 67.19 % |
| 86 | 100 | 25 | 30 | 48.72 % | 65.63 % |
| 87 | 120 | 25 | 30 | 51.28 % | 63.28 % |
| 88 | 140 | 25 | 30 | 49.82 % | 77.34 % |
| 89 | 80 | 30 | 30 | 49.08 % | 62.50 % |
| 90 | 100 | 30 | 30 | 50.55 % | 70.31 % |
| 91 | 120 | 30 | 30 | 52.38 % | 60.94 % |
| 92 | 140 | 30 | 30 | 51.65 % | 64.84 % |
| 93 | 80 | 35 | 30 | 51.10 % | 62.50 % |
| 94 | 100 | 35 | 30 | 56.25 % | 60.94 % |
| 95 | 120 | 35 | 30 | 51.10 % | 61.72 % |
| 96 | 140 | 35 | 30 | 48.90 % | 67.97 % |
| 97 | 80 | 40 | 30 | 50.37 % | 65.63 % |
| 98 | 100 | 40 | 30 | 50.37 % | 66.41 % |
| 99 | 120 | 40 | 30 | 51.84 % | 66.41 % |
| 100 | 140 | 40 | 30 | 52.57 % | 57.03 % |
| 101 | 80 | 45 | 30 | 50.55 % | 55.47 % |
| 102 | 100 | 45 | 30 | 51.66 % | 69.53 % |
| 103 | 120 | 45 | 30 | 55.35 % | 66.41 % |
| 104 | 140 | 45 | 30 | 47.60 % | 67.97 % |
| 105 | 80 | 50 | 30 | 49.45 % | 64.06 % |
| 106 | 100 | 50 | 30 | 54.98 % | 59.38 % |
| 107 | 120 | 50 | 30 | 54.98 % | 61.72 % |
| 108 | 140 | 50 | 30 | 52.77 % | 67.19 % |

Appendix 2. Results of SVM parameter optimization

| | PARAMETER OPTIMIZATION | | RESULTS | | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|---|---|---|---|
| combination | penalty term | gamma | validation accuracy | combination | penalty term | gamma | validation accuracy |
| 1 | 0.1 | 0.1 | 51.61 % | 51 | 52 | 0.2 | 47.31 % |
| 2 | 0.2 | 0.1 | 51.61 % | 52 | 57 | 0.2 | 48.03 % |
| 3 | 0.3 | 0.1 | 51.61 % | 53 | 62 | 0.2 | 48.39 % |
| 4 | 0.4 | 0.1 | 51.61 % | 54 | 67 | 0.2 | 47.67 % |
| 5 | 0.5 | 0.1 | 51.61 % | 55 | 72 | 0.2 | 47.67 % |
| 6 | 0.6 | 0.1 | 50.90 % | 56 | 77 | 0.2 | 47.67 % |
| 7 | 0.7 | 0.1 | 50.18 % | 57 | 82 | 0.2 | 47.67 % |
| 8 | 0.8 | 0.1 | 49.10 % | 58 | 87 | 0.2 | 47.67 % |
| 9 | 0.9 | 0.1 | 48.75 % | 59 | 92 | 0.2 | 48.03 % |
| 10 | 1 | 0.1 | 49.82 % | 60 | 97 | 0.2 | 48.03 % |
| 11 | 2 | 0.1 | 51.25 % | 61 | 0.1 | 0.3 | 51.61 % |
| 12 | 7 | 0.1 | 51.25 % | 62 | 0.2 | 0.3 | 51.97 % |
| 13 | 12 | 0.1 | 51.25 % | 63 | 0.3 | 0.3 | 49.10 % |
| 14 | 17 | 0.1 | 51.25 % | 64 | 0.4 | 0.3 | 48.75 % |
| 15 | 22 | 0.1 | 51.25 % | 65 | 0.5 | 0.3 | 47.67 % |
| 16 | 27 | 0.1 | 51.25 % | 66 | 0.6 | 0.3 | 48.39 % |
| 17 | 32 | 0.1 | 51.25 % | 67 | 0.7 | 0.3 | 48.39 % |
| 18 | 37 | 0.1 | 51.25 % | 68 | 0.8 | 0.3 | 49.10 % |
| 19 | 42 | 0.1 | 51.25 % | 69 | 0.9 | 0.3 | 48.03 % |
| 20 | 47 | 0.1 | 51.25 % | 70 | 1 | 0.3 | 49.46 % |
| 21 | 52 | 0.1 | 51.25 % | 71 | 2 | 0.3 | 49.10 % |
| 22 | 57 | 0.1 | 51.25 % | 72 | 7 | 0.3 | 46.24 % |
| 23 | 62 | 0.1 | 51.25 % | 73 | 12 | 0.3 | 47.67 % |
| 24 | 67 | 0.1 | 51.25 % | 74 | 17 | 0.3 | 48.39 % |
| 25 | 72 | 0.1 | 51.25 % | 75 | 22 | 0.3 | 46.95 % |
| 26 | 77 | 0.1 | 51.25 % | 76 | 27 | 0.3 | 47.67 % |
| 27 | 82 | 0.1 | 51.25 % | 77 | 32 | 0.3 | 47.31 % |
| 28 | 87 | 0.1 | 51.25 % | 78 | 37 | 0.3 | 48.03 % |
| 29 | 92 | 0.1 | 51.25 % | 79 | 42 | 0.3 | 48.75 % |
| 30 | 97 | 0.1 | 51.25 % | 80 | 47 | 0.3 | 47.31 % |
| 31 | 0.1 | 0.2 | 51.61 % | 81 | 52 | 0.3 | 47.67 % |
| 32 | 0.2 | 0.2 | 51.61 % | 82 | 57 | 0.3 | 47.67 % |
| 33 | 0.3 | 0.2 | 51.61 % | 83 | 62 | 0.3 | 47.31 % |
| 34 | 0.4 | 0.2 | 49.82 % | 84 | 67 | 0.3 | 46.24 % |
| 35 | 0.5 | 0.2 | 49.10 % | 85 | 72 | 0.3 | 45.16 % |
| 36 | 0.6 | 0.2 | 47.67 % | 86 | 77 | 0.3 | 44.80 % |
| 37 | 0.7 | 0.2 | 47.67 % | 87 | 82 | 0.3 | 45.16 % |
| 38 | 0.8 | 0.2 | 48.03 % | 88 | 87 | 0.3 | 46.24 % |
| 39 | 0.9 | 0.2 | 49.10 % | 89 | 92 | 0.3 | 45.88 % |
| 40 | 1 | 0.2 | 50.54 % | 90 | 97 | 0.3 | 45.88 % |
| 41 | 2 | 0.2 | 48.75 % | 91 | 0.1 | 0.4 | 51.61 % |
| 42 | 7 | 0.2 | 48.03 % | 92 | 0.2 | 0.4 | 51.61 % |
| 43 | 12 | 0.2 | 48.75 % | 93 | 0.3 | 0.4 | 49.46 % |
| 44 | 17 | 0.2 | 48.75 % | 94 | 0.4 | 0.4 | 49.10 % |
| 45 | 22 | 0.2 | 48.03 % | 95 | 0.5 | 0.4 | 52.33 % |
| 46 | 27 | 0.2 | 48.03 % | 96 | 0.6 | 0.4 | 50.54 % |
| 47 | 32 | 0.2 | 48.03 % | 97 | 0.7 | 0.4 | 50.54 % |
| 48 | 37 | 0.2 | 47.67 % | 98 | 0.8 | 0.4 | 49.46 % |
| 49 | 42 | 0.2 | 47.31 % | 99 | 0.9 | 0.4 | 49.46 % |
| 50 | 47 | 0.2 | 47.31 % | 100 | 1 | 0.4 | 48.75 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 101 | 2 | 0.4 | 48.03 % |
| 102 | 7 | 0.4 | 47.31 % |
| 103 | 12 | 0.4 | 45.88 % |
| 104 | 17 | 0.4 | 45.52 % |
| 105 | 22 | 0.4 | 45.88 % |
| 106 | 27 | 0.4 | 46.59 % |
| 107 | 32 | 0.4 | 46.95 % |
| 108 | 37 | 0.4 | 47.67 % |
| 109 | 42 | 0.4 | 46.95 % |
| 110 | 47 | 0.4 | 48.39 % |
| 111 | 52 | 0.4 | 47.67 % |
| 112 | 57 | 0.4 | 47.31 % |
| 113 | 62 | 0.4 | 48.03 % |
| 114 | 67 | 0.4 | 48.03 % |
| 115 | 72 | 0.4 | 48.39 % |
| 116 | 77 | 0.4 | 48.75 % |
| 117 | 82 | 0.4 | 48.39 % |
| 118 | 87 | 0.4 | 48.75 % |
| 119 | 92 | 0.4 | 48.39 % |
| 120 | 97 | 0.4 | 48.39 % |
| 121 | 0.1 | 0.5 | 51.61 % |
| 122 | 0.2 | 0.5 | 51.25 % |
| 123 | 0.3 | 0.5 | 51.97 % |
| 124 | 0.4 | 0.5 | 52.69 % |
| 125 | 0.5 | 0.5 | 55.20 % |
| 126 | 0.6 | 0.5 | 54.12 % |
| 127 | 0.7 | 0.5 | 52.33 % |
| 128 | 0.8 | 0.5 | 52.69 % |
| 129 | 0.9 | 0.5 | 52.33 % |
| 130 | 1 | 0.5 | 53.41 % |
| 131 | 2 | 0.5 | 50.18 % |
| 132 | 7 | 0.5 | 46.24 % |
| 133 | 12 | 0.5 | 43.37 % |
| 134 | 17 | 0.5 | 44.80 % |
| 135 | 22 | 0.5 | 44.80 % |
| 136 | 27 | 0.5 | 45.88 % |
| 137 | 32 | 0.5 | 47.67 % |
| 138 | 37 | 0.5 | 46.59 % |
| 139 | 42 | 0.5 | 47.31 % |
| 140 | 47 | 0.5 | 46.95 % |
| 141 | 52 | 0.5 | 47.67 % |
| 142 | 57 | 0.5 | 47.31 % |
| 143 | 62 | 0.5 | 46.95 % |
| 144 | 67 | 0.5 | 47.67 % |
| 145 | 72 | 0.5 | 46.95 % |
| 146 | 77 | 0.5 | 46.95 % |
| 147 | 82 | 0.5 | 46.59 % |
| 148 | 87 | 0.5 | 46.59 % |
| 149 | 92 | 0.5 | 46.24 % |
| 150 | 97 | 0.5 | 46.59 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 151 | 0.1 | 0.6 | 51.61 % |
| 152 | 0.2 | 0.6 | 53.05 % |
| 153 | 0.3 | 0.6 | 55.20 % |
| 154 | 0.4 | 0.6 | 55.91 % |
| 155 | 0.5 | 0.6 | 56.99 % |
| 156 | 0.6 | 0.6 | 57.35 % |
| 157 | 0.7 | 0.6 | 56.63 % |
| 158 | 0.8 | 0.6 | 56.63 % |
| 159 | 0.9 | 0.6 | 55.56 % |
| 160 | 1 | 0.6 | 54.84 % |
| 161 | 2 | 0.6 | 54.12 % |
| 162 | 7 | 0.6 | 50.54 % |
| 163 | 12 | 0.6 | 46.59 % |
| 164 | 17 | 0.6 | 47.31 % |
| 165 | 22 | 0.6 | 45.52 % |
| 166 | 27 | 0.6 | 46.24 % |
| 167 | 32 | 0.6 | 45.88 % |
| 168 | 37 | 0.6 | 45.16 % |
| 169 | 42 | 0.6 | 46.24 % |
| 170 | 47 | 0.6 | 45.52 % |
| 171 | 52 | 0.6 | 45.16 % |
| 172 | 57 | 0.6 | 45.16 % |
| 173 | 62 | 0.6 | 45.16 % |
| 174 | 67 | 0.6 | 44.80 % |
| 175 | 72 | 0.6 | 45.52 % |
| 176 | 77 | 0.6 | 46.24 % |
| 177 | 82 | 0.6 | 46.24 % |
| 178 | 87 | 0.6 | 46.59 % |
| 179 | 92 | 0.6 | 46.95 % |
| 180 | 97 | 0.6 | 46.95 % |
| 181 | 0.1 | 0.7 | 51.61 % |
| 182 | 0.2 | 0.7 | 53.41 % |
| 183 | 0.3 | 0.7 | 55.56 % |
| 184 | 0.4 | 0.7 | 57.35 % |
| 185 | 0.5 | 0.7 | 57.71 % |
| 186 | 0.6 | 0.7 | 56.63 % |
| 187 | 0.7 | 0.7 | 58.78 % |
| 188 | 0.8 | 0.7 | 58.06 % |
| 189 | 0.9 | 0.7 | 58.06 % |
| 190 | 1 | 0.7 | 58.78 % |
| 191 | 2 | 0.7 | 56.99 % |
| 192 | 7 | 0.7 | 52.33 % |
| 193 | 12 | 0.7 | 51.97 % |
| 194 | 17 | 0.7 | 50.54 % |
| 195 | 22 | 0.7 | 50.18 % |
| 196 | 27 | 0.7 | 48.03 % |
| 197 | 32 | 0.7 | 48.03 % |
| 198 | 37 | 0.7 | 47.67 % |
| 199 | 42 | 0.7 | 46.24 % |
| 200 | 47 | 0.7 | 46.95 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 201 | 52 | 0.7 | 47.67 % |
| 202 | 57 | 0.7 | 47.31 % |
| 203 | 62 | 0.7 | 46.59 % |
| 204 | 67 | 0.7 | 48.03 % |
| 205 | 72 | 0.7 | 46.95 % |
| 206 | 77 | 0.7 | 46.59 % |
| 207 | 82 | 0.7 | 46.59 % |
| 208 | 87 | 0.7 | 46.59 % |
| 209 | 92 | 0.7 | 47.31 % |
| 210 | 97 | 0.7 | 46.95 % |
| 211 | 0.1 | 0.8 | 52.33 % |
| 212 | 0.2 | 0.8 | 54.12 % |
| 213 | 0.3 | 0.8 | 56.27 % |
| 214 | 0.4 | 0.8 | 56.63 % |
| 215 | 0.5 | 0.8 | 56.27 % |
| 216 | 0.6 | 0.8 | 57.71 % |
| 217 | 0.7 | 0.8 | 57.71 % |
| 218 | 0.8 | 0.8 | 58.78 % |
| 219 | 0.9 | 0.8 | 59.50 % |
| 220 | 1 | 0.8 | 59.86 % |
| 221 | 2 | 0.8 | 58.78 % |
| 222 | 7 | 0.8 | 55.56 % |
| 223 | 12 | 0.8 | 55.91 % |
| 224 | 17 | 0.8 | 55.56 % |
| 225 | 22 | 0.8 | 54.12 % |
| 226 | 27 | 0.8 | 54.84 % |
| 227 | 32 | 0.8 | 55.20 % |
| 228 | 37 | 0.8 | 54.84 % |
| 229 | 42 | 0.8 | 52.33 % |
| 230 | 47 | 0.8 | 51.25 % |
| 231 | 52 | 0.8 | 50.90 % |
| 232 | 57 | 0.8 | 50.18 % |
| 233 | 62 | 0.8 | 49.10 % |
| 234 | 67 | 0.8 | 50.18 % |
| 235 | 72 | 0.8 | 49.82 % |
| 236 | 77 | 0.8 | 48.75 % |
| 237 | 82 | 0.8 | 49.10 % |
| 238 | 87 | 0.8 | 48.75 % |
| 239 | 92 | 0.8 | 48.75 % |
| 240 | 97 | 0.8 | 48.75 % |
| 241 | 0.1 | 0.9 | 51.61 % |
| 242 | 0.2 | 0.9 | 53.76 % |
| 243 | 0.3 | 0.9 | 55.91 % |
| 244 | 0.4 | 0.9 | 56.27 % |
| 245 | 0.5 | 0.9 | 56.27 % |
| 246 | 0.6 | 0.9 | 56.63 % |
| 247 | 0.7 | 0.9 | 56.99 % |
| 248 | 0.8 | 0.9 | 56.63 % |
| 249 | 0.9 | 0.9 | 57.35 % |
| 250 | 1 | 0.9 | 58.78 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 251 | 2 | 0.9 | 59.50 % |
| 252 | 7 | 0.9 | 58.06 % |
| 253 | 12 | 0.9 | 57.71 % |
| 254 | 17 | 0.9 | 58.42 % |
| 255 | 22 | 0.9 | 59.14 % |
| 256 | 27 | 0.9 | 59.50 % |
| 257 | 32 | 0.9 | 58.42 % |
| 258 | 37 | 0.9 | 58.42 % |
| 259 | 42 | 0.9 | 57.35 % |
| 260 | 47 | 0.9 | 56.27 % |
| 261 | 52 | 0.9 | 56.63 % |
| 262 | 57 | 0.9 | 56.27 % |
| 263 | 62 | 0.9 | 56.63 % |
| 264 | 67 | 0.9 | 56.27 % |
| 265 | 72 | 0.9 | 55.91 % |
| 266 | 77 | 0.9 | 55.20 % |
| 267 | 82 | 0.9 | 55.56 % |
| 268 | 87 | 0.9 | 54.48 % |
| 269 | 92 | 0.9 | 54.12 % |
| 270 | 97 | 0.9 | 53.41 % |
| 271 | 0.1 | 1 | 51.97 % |
| 272 | 0.2 | 1 | 54.12 % |
| 273 | 0.3 | 1 | 55.91 % |
| 274 | 0.4 | 1 | 56.99 % |
| 275 | 0.5 | 1 | 55.91 % |
| 276 | 0.6 | 1 | 55.91 % |
| 277 | 0.7 | 1 | 56.99 % |
| 278 | 0.8 | 1 | 57.35 % |
| 279 | 0.9 | 1 | 57.35 % |
| 280 | 1 | 1 | 56.63 % |
| 281 | 2 | 1 | 60.22 % |
| 282 | 7 | 1 | 58.42 % |
| 283 | 12 | 1 | 56.99 % |
| 284 | 17 | 1 | 56.99 % |
| 285 | 22 | 1 | 57.71 % |
| 286 | 27 | 1 | 58.42 % |
| 287 | 32 | 1 | 57.71 % |
| 288 | 37 | 1 | 56.99 % |
| 289 | 42 | 1 | 56.27 % |
| 290 | 47 | 1 | 56.99 % |
| 291 | 52 | 1 | 58.06 % |
| 292 | 57 | 1 | 58.06 % |
| 293 | 62 | 1 | 58.06 % |
| 294 | 67 | 1 | 57.35 % |
| 295 | 72 | 1 | 57.35 % |
| 296 | 77 | 1 | 56.63 % |
| 297 | 82 | 1 | 56.27 % |
| 298 | 87 | 1 | 56.63 % |
| 299 | 92 | 1 | 56.63 % |
| 300 | 97 | 1 | 56.99 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 301 | 0.1 | 2 | 52.33 % |
| 302 | 0.2 | 2 | 53.41 % |
| 303 | 0.3 | 2 | 54.12 % |
| 304 | 0.4 | 2 | 54.48 % |
| 305 | 0.5 | 2 | 55.20 % |
| 306 | 0.6 | 2 | 55.56 % |
| 307 | 0.7 | 2 | 55.20 % |
| 308 | 0.8 | 2 | 55.20 % |
| 309 | 0.9 | 2 | 55.20 % |
| 310 | 1 | 2 | 55.56 % |
| 311 | 2 | 2 | 55.91 % |
| 312 | 7 | 2 | 56.63 % |
| 313 | 12 | 2 | 56.27 % |
| 314 | 17 | 2 | 56.63 % |
| 315 | 22 | 2 | 56.63 % |
| 316 | 27 | 2 | 58.06 % |
| 317 | 32 | 2 | 58.42 % |
| 318 | 37 | 2 | 59.14 % |
| 319 | 42 | 2 | 59.14 % |
| 320 | 47 | 2 | 59.14 % |
| 321 | 52 | 2 | 59.14 % |
| 322 | 57 | 2 | 59.14 % |
| 323 | 62 | 2 | 59.50 % |
| 324 | 67 | 2 | 59.14 % |
| 325 | 72 | 2 | 59.14 % |
| 326 | 77 | 2 | 59.14 % |
| 327 | 82 | 2 | 59.86 % |
| 328 | 87 | 2 | 60.22 % |
| 329 | 92 | 2 | 60.22 % |
| 330 | 97 | 2 | 59.86 % |
| 331 | 0.1 | 3 | 51.61 % |
| 332 | 0.2 | 3 | 52.33 % |
| 333 | 0.3 | 3 | 53.05 % |
| 334 | 0.4 | 3 | 53.05 % |
| 335 | 0.5 | 3 | 52.69 % |
| 336 | 0.6 | 3 | 52.69 % |
| 337 | 0.7 | 3 | 52.69 % |
| 338 | 0.8 | 3 | 52.69 % |
| 339 | 0.9 | 3 | 51.97 % |
| 340 | 1 | 3 | 52.33 % |
| 341 | 2 | 3 | 54.84 % |
| 342 | 7 | 3 | 55.20 % |
| 343 | 12 | 3 | 55.91 % |
| 344 | 17 | 3 | 56.27 % |
| 345 | 22 | 3 | 55.91 % |
| 346 | 27 | 3 | 56.63 % |
| 347 | 32 | 3 | 56.63 % |
| 348 | 37 | 3 | 56.27 % |
| 349 | 42 | 3 | 56.27 % |
| 350 | 47 | 3 | 56.27 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 351 | 52 | 3 | 55.91 % |
| 352 | 57 | 3 | 55.91 % |
| 353 | 62 | 3 | 55.91 % |
| 354 | 67 | 3 | 55.91 % |
| 355 | 72 | 3 | 55.91 % |
| 356 | 77 | 3 | 56.27 % |
| 357 | 82 | 3 | 56.63 % |
| 358 | 87 | 3 | 56.63 % |
| 359 | 92 | 3 | 56.63 % |
| 360 | 97 | 3 | 56.63 % |
| 361 | 0.1 | 4 | 51.61 % |
| 362 | 0.2 | 4 | 51.61 % |
| 363 | 0.3 | 4 | 52.33 % |
| 364 | 0.4 | 4 | 52.69 % |
| 365 | 0.5 | 4 | 52.33 % |
| 366 | 0.6 | 4 | 52.33 % |
| 367 | 0.7 | 4 | 52.33 % |
| 368 | 0.8 | 4 | 52.33 % |
| 369 | 0.9 | 4 | 52.69 % |
| 370 | 1 | 4 | 52.33 % |
| 371 | 2 | 4 | 53.05 % |
| 372 | 7 | 4 | 52.69 % |
| 373 | 12 | 4 | 54.12 % |
| 374 | 17 | 4 | 54.12 % |
| 375 | 22 | 4 | 54.84 % |
| 376 | 27 | 4 | 55.56 % |
| 377 | 32 | 4 | 55.56 % |
| 378 | 37 | 4 | 55.56 % |
| 379 | 42 | 4 | 55.20 % |
| 380 | 47 | 4 | 54.84 % |
| 381 | 52 | 4 | 54.84 % |
| 382 | 57 | 4 | 54.84 % |
| 383 | 62 | 4 | 54.84 % |
| 384 | 67 | 4 | 55.20 % |
| 385 | 72 | 4 | 55.20 % |
| 386 | 77 | 4 | 55.20 % |
| 387 | 82 | 4 | 55.20 % |
| 388 | 87 | 4 | 55.56 % |
| 389 | 92 | 4 | 55.91 % |
| 390 | 97 | 4 | 55.91 % |
| 391 | 0.1 | 5 | 51.61 % |
| 392 | 0.2 | 5 | 51.61 % |
| 393 | 0.3 | 5 | 51.61 % |
| 394 | 0.4 | 5 | 51.97 % |
| 395 | 0.5 | 5 | 51.97 % |
| 396 | 0.6 | 5 | 51.97 % |
| 397 | 0.7 | 5 | 51.97 % |
| 398 | 0.8 | 5 | 51.97 % |
| 399 | 0.9 | 5 | 52.33 % |
| 400 | 1 | 5 | 52.33 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 401 | 2 | 5 | 52.69 % |
| 402 | 7 | 5 | 53.05 % |
| 403 | 12 | 5 | 52.69 % |
| 404 | 17 | 5 | 52.33 % |
| 405 | 22 | 5 | 51.97 % |
| 406 | 27 | 5 | 51.97 % |
| 407 | 32 | 5 | 51.97 % |
| 408 | 37 | 5 | 51.97 % |
| 409 | 42 | 5 | 52.33 % |
| 410 | 47 | 5 | 53.05 % |
| 411 | 52 | 5 | 53.76 % |
| 412 | 57 | 5 | 54.12 % |
| 413 | 62 | 5 | 53.76 % |
| 414 | 67 | 5 | 53.76 % |
| 415 | 72 | 5 | 53.76 % |
| 416 | 77 | 5 | 53.76 % |
| 417 | 82 | 5 | 54.48 % |
| 418 | 87 | 5 | 54.48 % |
| 419 | 92 | 5 | 54.48 % |
| 420 | 97 | 5 | 54.48 % |
| 421 | 0.1 | 6 | 51.61 % |
| 422 | 0.2 | 6 | 51.61 % |
| 423 | 0.3 | 6 | 51.61 % |
| 424 | 0.4 | 6 | 51.61 % |
| 425 | 0.5 | 6 | 51.61 % |
| 426 | 0.6 | 6 | 51.61 % |
| 427 | 0.7 | 6 | 51.97 % |
| 428 | 0.8 | 6 | 51.97 % |
| 429 | 0.9 | 6 | 51.97 % |
| 430 | 1 | 6 | 51.97 % |
| 431 | 2 | 6 | 51.97 % |
| 432 | 7 | 6 | 52.69 % |
| 433 | 12 | 6 | 52.69 % |
| 434 | 17 | 6 | 52.69 % |
| 435 | 22 | 6 | 53.05 % |
| 436 | 27 | 6 | 53.05 % |
| 437 | 32 | 6 | 52.69 % |
| 438 | 37 | 6 | 52.69 % |
| 439 | 42 | 6 | 52.69 % |
| 440 | 47 | 6 | 52.69 % |
| 441 | 52 | 6 | 52.69 % |
| 442 | 57 | 6 | 52.69 % |
| 443 | 62 | 6 | 52.69 % |
| 444 | 67 | 6 | 52.69 % |
| 445 | 72 | 6 | 52.69 % |
| 446 | 77 | 6 | 52.69 % |
| 447 | 82 | 6 | 52.69 % |
| 448 | 87 | 6 | 52.69 % |
| 449 | 92 | 6 | 52.69 % |
| 450 | 97 | 6 | 52.69 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 451 | 0.1 | 7 | 51.61 % |
| 452 | 0.2 | 7 | 51.61 % |
| 453 | 0.3 | 7 | 51.61 % |
| 454 | 0.4 | 7 | 51.61 % |
| 455 | 0.5 | 7 | 51.61 % |
| 456 | 0.6 | 7 | 51.61 % |
| 457 | 0.7 | 7 | 51.61 % |
| 458 | 0.8 | 7 | 51.61 % |
| 459 | 0.9 | 7 | 51.61 % |
| 460 | 1 | 7 | 51.61 % |
| 461 | 2 | 7 | 51.61 % |
| 462 | 7 | 7 | 52.33 % |
| 463 | 12 | 7 | 52.33 % |
| 464 | 17 | 7 | 52.33 % |
| 465 | 22 | 7 | 52.33 % |
| 466 | 27 | 7 | 52.33 % |
| 467 | 32 | 7 | 52.69 % |
| 468 | 37 | 7 | 52.69 % |
| 469 | 42 | 7 | 52.69 % |
| 470 | 47 | 7 | 52.69 % |
| 471 | 52 | 7 | 52.69 % |
| 472 | 57 | 7 | 52.33 % |
| 473 | 62 | 7 | 52.33 % |
| 474 | 67 | 7 | 52.33 % |
| 475 | 72 | 7 | 52.69 % |
| 476 | 77 | 7 | 52.69 % |
| 477 | 82 | 7 | 52.69 % |
| 478 | 87 | 7 | 52.69 % |
| 479 | 92 | 7 | 52.69 % |
| 480 | 97 | 7 | 52.69 % |
| 481 | 0.1 | 8 | 51.61 % |
| 482 | 0.2 | 8 | 51.61 % |
| 483 | 0.3 | 8 | 51.61 % |
| 484 | 0.4 | 8 | 51.61 % |
| 485 | 0.5 | 8 | 51.61 % |
| 486 | 0.6 | 8 | 51.61 % |
| 487 | 0.7 | 8 | 51.61 % |
| 488 | 0.8 | 8 | 51.61 % |
| 489 | 0.9 | 8 | 51.61 % |
| 490 | 1 | 8 | 51.61 % |
| 491 | 2 | 8 | 51.61 % |
| 492 | 7 | 8 | 51.97 % |
| 493 | 12 | 8 | 52.33 % |
| 494 | 17 | 8 | 52.33 % |
| 495 | 22 | 8 | 52.33 % |
| 496 | 27 | 8 | 52.33 % |
| 497 | 32 | 8 | 52.33 % |
| 498 | 37 | 8 | 52.33 % |
| 499 | 42 | 8 | 52.33 % |
| 500 | 47 | 8 | 52.33 % |

| | PARAMETER OPTIMIZATION | | RESULTS |
|---|---|---|---|
| combination | penalty term | gamma | validation accuracy |
| 501 | 52 | 8 | 52.33 % |
| 502 | 57 | 8 | 52.33 % |
| 503 | 62 | 8 | 52.33 % |
| 504 | 67 | 8 | 52.33 % |
| 505 | 72 | 8 | 52.33 % |
| 506 | 77 | 8 | 52.33 % |
| 507 | 82 | 8 | 52.33 % |
| 508 | 87 | 8 | 52.33 % |
| 509 | 92 | 8 | 52.33 % |
| 510 | 97 | 8 | 52.69 % |
| 511 | 0.1 | 9 | 51.61 % |
| 512 | 0.2 | 9 | 51.61 % |
| 513 | 0.3 | 9 | 51.61 % |
| 514 | 0.4 | 9 | 51.61 % |
| 515 | 0.5 | 9 | 51.61 % |
| 516 | 0.6 | 9 | 51.61 % |
| 517 | 0.7 | 9 | 51.61 % |
| 518 | 0.8 | 9 | 51.61 % |
| 519 | 0.9 | 9 | 51.61 % |
| 520 | 1 | 9 | 51.61 % |
| 521 | 2 | 9 | 51.61 % |
| 522 | 7 | 9 | 51.61 % |
| 523 | 12 | 9 | 51.61 % |
| 524 | 17 | 9 | 52.33 % |
| 525 | 22 | 9 | 51.97 % |
| 526 | 27 | 9 | 52.33 % |
| 527 | 32 | 9 | 52.33 % |
| 528 | 37 | 9 | 52.33 % |
| 529 | 42 | 9 | 52.33 % |
| 530 | 47 | 9 | 52.33 % |
| 531 | 52 | 9 | 51.97 % |
| 532 | 57 | 9 | 51.97 % |
| 533 | 62 | 9 | 51.97 % |
| 534 | 67 | 9 | 51.97 % |
| 535 | 72 | 9 | 51.97 % |
| 536 | 77 | 9 | 51.97 % |
| 537 | 82 | 9 | 52.33 % |
| 538 | 87 | 9 | 52.33 % |
| 539 | 92 | 9 | 52.33 % |
| 540 | 97 | 9 | 52.33 % |
| 541 | 0.1 | 10 | 51.61 % |
| 542 | 0.2 | 10 | 51.61 % |
| 543 | 0.3 | 10 | 51.61 % |
| 544 | 0.4 | 10 | 51.61 % |
| 545 | 0.5 | 10 | 51.61 % |
| 546 | 0.6 | 10 | 51.61 % |
| 547 | 0.7 | 10 | 51.61 % |
| 548 | 0.8 | 10 | 51.61 % |
| 549 | 0.9 | 10 | 51.61 % |
| 550 | 1 | 10 | 51.61 % |
| 551 | 2 | 10 | 51.61 % |
| 552 | 7 | 10 | 51.61 % |
| 553 | 12 | 10 | 51.61 % |
| 554 | 17 | 10 | 51.61 % |
| 555 | 22 | 10 | 51.97 % |
| 556 | 27 | 10 | 52.33 % |
| 557 | 32 | 10 | 52.33 % |
| 558 | 37 | 10 | 52.33 % |
| 559 | 42 | 10 | 51.97 % |
| 560 | 47 | 10 | 52.33 % |
| 561 | 52 | 10 | 52.33 % |
| 562 | 57 | 10 | 52.33 % |
| 563 | 62 | 10 | 51.97 % |
| 564 | 67 | 10 | 51.97 % |
| 565 | 72 | 10 | 51.61 % |
| 566 | 77 | 10 | 51.61 % |
| 567 | 82 | 10 | 51.61 % |
| 568 | 87 | 10 | 51.61 % |
| 569 | 92 | 10 | 51.97 % |
| 570 | 97 | 10 | 51.97 % |

Appendix 3. Comparison of predictions for the OMXH25 data

| Day | Returns | Actual | LSTM | SVM | Random guess | Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.21 % | 1 | 0 | 0 | 1 | **41** | -1.21 % | 0 | 0 | 1 | 0 |
| **2** | 0.60 % | 1 | 1 | 1 | 0 | **42** | 0.79 % | 1 | 0 | 0 | 1 |
| **3** | -0.95 % | 0 | 1 | 0 | 0 | **43** | -0.36 % | 0 | 1 | 1 | 1 |
| **4** | -0.07 % | 0 | 1 | 0 | 0 | **44** | -1.00 % | 0 | 0 | 1 | 1 |
| **5** | 0.22 % | 1 | 1 | 1 | 1 | **45** | 0.33 % | 1 | 0 | 1 | 1 |
| **6** | -0.49 % | 0 | 0 | 1 | 0 | **46** | 0.56 % | 1 | 1 | 1 | 0 |
| **7** | 2.39 % | 1 | 1 | 1 | 1 | **47** | 1.43 % | 1 | 1 | 1 | 1 |
| **8** | -1.84 % | 0 | 1 | 0 | 1 | **48** | 1.21 % | 1 | 1 | 0 | 1 |
| **9** | -0.86 % | 0 | 0 | 0 | 1 | **49** | -1.02 % | 0 | 1 | 0 | 1 |
| **10** | -1.32 % | 0 | 1 | 1 | 1 | **50** | -0.50 % | 0 | 0 | 0 | 1 |
| **11** | -1.68 % | 0 | 1 | 1 | 1 | **51** | 0.76 % | 1 | 0 | 1 | 1 |
| **12** | 1.73 % | 1 | 1 | 1 | 0 | **52** | -0.30 % | 0 | 0 | 1 | 0 |
| **13** | 2.82 % | 1 | 1 | 1 | 0 | **53** | 0.54 % | 1 | 1 | 1 | 1 |
| **14** | -0.76 % | 0 | 1 | 1 | 1 | **54** | -0.04 % | 0 | 1 | 1 | 1 |
| **15** | 0.10 % | 1 | 1 | 1 | 1 | **55** | 0.30 % | 1 | 0 | 1 | 0 |
| **16** | -0.80 % | 0 | 0 | 1 | 1 | **56** | 0.57 % | 1 | 0 | 1 | 0 |
| **17** | -0.43 % | 0 | 0 | 0 | 0 | **57** | -1.06 % | 0 | 0 | 1 | 0 |
| **18** | 1.00 % | 1 | 0 | 1 | 0 | **58** | 0.44 % | 1 | 0 | 0 | 0 |
| **19** | -2.27 % | 0 | 0 | 1 | 0 | **59** | -1.14 % | 0 | 1 | 1 | 0 |
| **20** | -0.67 % | 0 | 0 | 0 | 0 | **60** | 0.11 % | 1 | 0 | 0 | 1 |
| **21** | -1.33 % | 0 | 1 | 1 | 0 | **61** | 0.73 % | 1 | 1 | 1 | 1 |
| **22** | 1.22 % | 1 | 0 | 1 | 1 | **62** | 0.67 % | 1 | 0 | 1 | 0 |
| **23** | 0.79 % | 1 | 1 | 1 | 0 | **63** | -0.03 % | 0 | 1 | 1 | 1 |
| **24** | -0.66 % | 0 | 1 | 1 | 1 | **64** | -0.81 % | 0 | 1 | 1 | 0 |
| **25** | 3.21 % | 1 | 1 | 0 | 0 | **65** | 2.15 % | 1 | 0 | 1 | 1 |
| **26** | 0.10 % | 1 | 1 | 0 | 1 | **66** | -0.09 % | 0 | 0 | 0 | 0 |
| **27** | 1.41 % | 1 | 1 | 1 | 0 | **67** | -0.36 % | 0 | 1 | 1 | 0 |
| **28** | 0.87 % | 1 | 0 | 1 | 0 | **68** | -0.55 % | 0 | 1 | 1 | 0 |
| **29** | 0.06 % | 1 | 0 | 1 | 1 | **69** | -0.80 % | 0 | 0 | 1 | 1 |
| **30** | -0.26 % | 0 | 1 | 1 | 0 | **70** | -0.47 % | 0 | 0 | 1 | 1 |
| **31** | -0.45 % | 0 | 0 | 1 | 0 | **71** | 1.25 % | 1 | 1 | 1 | 1 |
| **32** | 0.52 % | 1 | 1 | 1 | 1 | **72** | 0.33 % | 1 | 1 | 0 | 1 |
| **33** | 1.22 % | 1 | 0 | 0 | 1 | **73** | -0.20 % | 0 | 1 | 1 | 1 |
| **34** | 0.37 % | 1 | 1 | 1 | 0 | **74** | 1.72 % | 1 | 0 | 1 | 0 |
| **35** | 1.22 % | 1 | 1 | 1 | 1 | **75** | 0.76 % | 1 | 0 | 1 | 1 |
| **36** | 0.59 % | 1 | 0 | 0 | 0 | **76** | 0.62 % | 1 | 1 | 1 | 1 |
| **37** | -0.42 % | 0 | 0 | 1 | 0 | **77** | 0.81 % | 1 | 1 | 1 | 1 |
| **38** | -0.28 % | 0 | 0 | 1 | 0 | **78** | -1.25 % | 0 | 0 | 1 | 1 |
| **39** | 0.64 % | 1 | 0 | 1 | 0 | **79** | -0.04 % | 0 | 0 | 0 | 0 |
| **40** | 0.75 % | 1 | 0 | 1 | 1 | **80** | -2.25 % | 0 | 1 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **81** | -0.70 % | 0 | 1 | 0 | 0 |
| **82** | 0.16 % | 1 | 1 | 1 | 0 |
| **83** | 0.33 % | 1 | 0 | 1 | 0 |
| **84** | -0.98 % | 0 | 1 | 1 | 0 |
| **85** | 0.40 % | 1 | 1 | 1 | 0 |
| **86** | 2.37 % | 1 | 1 | 0 | 1 |
| **87** | 1.11 % | 1 | 1 | 0 | 1 |
| **88** | 1.63 % | 1 | 1 | 1 | 0 |
| **89** | 0.10 % | 1 | 1 | 0 | 1 |
| **90** | 0.83 % | 1 | 1 | 1 | 0 |
| **91** | -0.71 % | 0 | 0 | 1 | 0 |
| **92** | -0.70 % | 0 | 0 | 1 | 1 |
| **93** | 0.55 % | 1 | 1 | 1 | 0 |
| **94** | 0.05 % | 1 | 1 | 1 | 1 |
| **95** | 0.34 % | 1 | 1 | 1 | 1 |
| **96** | -0.49 % | 0 | 1 | 1 | 1 |
| **97** | -0.53 % | 0 | 0 | 1 | 1 |
| **98** | -0.68 % | 0 | 0 | 1 | 1 |
| **99** | 0.81 % | 1 | 1 | 1 | 1 |
| **100** | 0.38 % | 1 | 1 | 1 | 0 |
| **101** | -0.67 % | 0 | 1 | 1 | 1 |
| **102** | -0.09 % | 0 | 0 | 1 | 0 |
| **103** | -0.32 % | 0 | 1 | 1 | 1 |
| **104** | -0.74 % | 0 | 0 | 0 | 0 |
| **105** | -1.36 % | 0 | 1 | 1 | 1 |
| **106** | -0.72 % | 0 | 1 | 1 | 0 |
| **107** | 0.29 % | 1 | 1 | 1 | 1 |
| **108** | -1.11 % | 0 | 1 | 1 | 1 |
| **109** | -1.02 % | 0 | 1 | 1 | 1 |
| **110** | -0.53 % | 0 | 1 | 1 | 1 |
| **111** | -1.47 % | 0 | 1 | 1 | 1 |
| **112** | 1.10 % | 1 | 1 | 0 | 0 |
| **113** | -2.24 % | 0 | 1 | 1 | 1 |
| **114** | 1.15 % | 1 | 0 | 0 | 1 |
| **115** | 0.14 % | 1 | 1 | 1 | 0 |
| **116** | 2.05 % | 1 | 1 | 1 | 0 |
| **117** | 0.35 % | 1 | 1 | 0 | 0 |
| **118** | -0.81 % | 0 | 1 | 1 | 0 |
| **119** | 1.36 % | 1 | 1 | 1 | 1 |
| **120** | -0.11 % | 0 | 0 | 1 | 1 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **121** | -2.32 % | 0 | 0 | 1 | 1 |
| **122** | 0.38 % | 1 | 1 | 0 | 0 |
| **123** | 0.55 % | 1 | 1 | 1 | 0 |
| **124** | -0.73 % | 0 | 1 | 1 | 1 |
| **125** | -1.51 % | 0 | 0 | 1 | 1 |
| **126** | -0.48 % | 0 | 0 | 0 | 1 |
| **127** | 0.27 % | 1 | 1 | 1 | 1 |
| **128** | 0.73 % | 1 | 1 | 1 | 0 |
| **129** | -0.48 % | 0 | 1 | 0 | 1 |
| **130** | -0.20 % | 0 | 0 | 1 | 0 |
| **131** | 1.17 % | 1 | 1 | 1 | 1 |
| **132** | 0.10 % | 1 | 1 | 1 | 1 |
| **133** | 2.13 % | 1 | 1 | 1 | 0 |
| **134** | -0.65 % | 0 | 1 | 0 | 1 |
| **135** | 0.51 % | 1 | 1 | 0 | 0 |
| **136** | -0.89 % | 0 | 0 | 1 | 1 |
| **137** | -0.72 % | 0 | 0 | 0 | 1 |
| **138** | 1.70 % | 1 | 1 | 1 | 1 |
| **139** | 0.11 % | 1 | 0 | 1 | 0 |
| **140** | 0.02 % | 1 | 1 | 1 | 1 |
| **141** | -0.89 % | 0 | 1 | 1 | 1 |
| **142** | 0.31 % | 1 | 0 | 0 | 0 |
| **143** | 0.97 % | 1 | 0 | 1 | 1 |
| **144** | -0.14 % | 0 | 0 | 1 | 0 |
| **145** | 0.11 % | 1 | 1 | 1 | 1 |
| **146** | 1.16 % | 1 | 1 | 1 | 0 |
| **147** | -0.20 % | 0 | 0 | 1 | 0 |
| **148** | 0.64 % | 1 | 0 | 1 | 1 |
| **149** | 0.44 % | 1 | 1 | 1 | 0 |
| **150** | -0.48 % | 0 | 0 | 0 | 0 |
| **151** | -0.49 % | 0 | 1 | 1 | 1 |
| **152** | -1.19 % | 0 | 0 | 1 | 1 |
| **153** | -0.01 % | 0 | 1 | 0 | 0 |
| **154** | -0.02 % | 0 | 1 | 1 | 1 |
| **155** | 0.41 % | 1 | 1 | 1 | 1 |
| **156** | 0.63 % | 1 | 1 | 1 | 1 |
| **157** | 0.44 % | 1 | 1 | 1 | 1 |
| **158** | -0.99 % | 0 | 1 | 1 | 0 |
| **159** | -2.15 % | 0 | 0 | 1 | 1 |
| **160** | 0.66 % | 1 | 0 | 1 | 1 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| 161 | 0.09 % | 1 | 0 | 1 | 0 |
| 162 | 1.83 % | 1 | 0 | 1 | 0 |
| 163 | 0.15 % | 1 | 1 | 1 | 0 |
| 164 | 0.05 % | 1 | 1 | 1 | 0 |
| 165 | -0.72 % | 0 | 0 | 0 | 1 |
| 166 | -0.22 % | 0 | 0 | 0 | 1 |
| 167 | -1.29 % | 0 | 0 | 1 | 1 |
| 168 | 0.18 % | 1 | 1 | 0 | 1 |
| 169 | 0.12 % | 1 | 1 | 1 | 1 |
| 170 | -1.32 % | 0 | 1 | 1 | 1 |
| 171 | -1.16 % | 0 | 0 | 0 | 0 |
| 172 | -0.06 % | 0 | 1 | 1 | 0 |
| 173 | -0.39 % | 0 | 1 | 1 | 0 |
| 174 | 1.72 % | 1 | 1 | 1 | 0 |
| 175 | 0.09 % | 1 | 1 | 0 | 1 |
| 176 | -0.73 % | 0 | 1 | 1 | 1 |
| 177 | 0.72 % | 1 | 1 | 1 | 1 |
| 178 | -2.66 % | 0 | 1 | 1 | 0 |
| 179 | -0.35 % | 0 | 0 | 0 | 0 |
| 180 | 1.54 % | 1 | 1 | 1 | 0 |
| 181 | 1.49 % | 1 | 0 | 0 | 1 |
| 182 | -1.57 % | 0 | 1 | 0 | 0 |
| 183 | 1.10 % | 1 | 0 | 0 | 0 |
| 184 | -0.23 % | 0 | 0 | 1 | 0 |
| 185 | -1.26 % | 0 | 0 | 1 | 1 |
| 186 | -0.36 % | 0 | 0 | 0 | 0 |
| 187 | 0.47 % | 1 | 1 | 1 | 1 |
| 188 | -1.10 % | 0 | 0 | 1 | 1 |
| 189 | 1.78 % | 1 | 0 | 0 | 1 |
| 190 | 0.76 % | 1 | 1 | 1 | 1 |
| 191 | 0.15 % | 1 | 1 | 1 | 0 |
| 192 | -0.96 % | 0 | 0 | 1 | 0 |
| 193 | 1.59 % | 1 | 0 | 1 | 1 |
| 194 | 1.53 % | 1 | 0 | 1 | 0 |
| 195 | 0.68 % | 1 | 0 | 1 | 1 |
| 196 | 0.69 % | 1 | 1 | 1 | 1 |
| 197 | 0.44 % | 1 | 1 | 1 | 1 |
| 198 | 0.00 % | 1 | 0 | 1 | 1 |
| 199 | 1.33 % | 1 | 0 | 1 | 0 |
| 200 | 0.25 % | 1 | 1 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| 201 | 0.13 % | 1 | 1 | 1 | 0 |
| 202 | -0.70 % | 0 | 1 | 1 | 1 |
| 203 | -0.83 % | 0 | 0 | 1 | 0 |
| 204 | 0.58 % | 1 | 0 | 1 | 1 |
| 205 | 0.02 % | 1 | 1 | 1 | 1 |
| 206 | -1.10 % | 0 | 0 | 1 | 1 |
| 207 | -0.44 % | 0 | 0 | 0 | 1 |
| 208 | -0.50 % | 0 | 1 | 1 | 0 |
| 209 | 0.28 % | 1 | 1 | 1 | 0 |
| 210 | 1.18 % | 1 | 1 | 1 | 0 |
| 211 | 0.66 % | 1 | 1 | 1 | 0 |
| 212 | -1.20 % | 0 | 1 | 1 | 0 |
| 213 | -2.21 % | 0 | 0 | 0 | 1 |
| 214 | -1.08 % | 0 | 0 | 1 | 1 |
| 215 | 0.70 % | 1 | 1 | 0 | 1 |
| 216 | 1.03 % | 1 | 0 | 1 | 1 |
| 217 | -0.59 % | 0 | 1 | 1 | 1 |
| 218 | 0.52 % | 1 | 1 | 1 | 0 |
| 219 | 1.03 % | 1 | 1 | 1 | 1 |
| 220 | 1.93 % | 1 | 0 | 1 | 0 |
| 221 | -0.75 % | 0 | 1 | 1 | 1 |
| 222 | 0.92 % | 1 | 1 | 0 | 0 |
| 223 | 0.08 % | 1 | 1 | 1 | 1 |
| 224 | 0.51 % | 1 | 0 | 1 | 1 |
| 225 | 0.07 % | 1 | 0 | 0 | 1 |
| 226 | 0.86 % | 1 | 1 | 1 | 0 |
| 227 | 0.22 % | 1 | 0 | 1 | 1 |
| 228 | 0.62 % | 1 | 0 | 1 | 1 |
| 229 | -0.78 % | 0 | 1 | 1 | 1 |
| 230 | 0.34 % | 1 | 0 | 1 | 1 |
| 231 | 0.62 % | 1 | 0 | 1 | 1 |
| 232 | -0.50 % | 0 | 0 | 1 | 0 |
| 233 | 0.34 % | 1 | 1 | 1 | 0 |
| 234 | -1.00 % | 0 | 1 | 1 | 1 |
| 235 | 0.52 % | 1 | 0 | 1 | 1 |
| 236 | 1.30 % | 1 | 1 | 1 | 0 |
| 237 | 0.08 % | 1 | 0 | 1 | 0 |
| 238 | -0.30 % | 0 | 1 | 1 | 1 |
| 239 | 0.25 % | 1 | 1 | 1 | 1 |
| 240 | -0.23 % | 0 | 0 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **241** | -0.15 % | 0 | 0 | 1 | 0 |
| **242** | 0.38 % | 1 | 1 | 1 | 1 |
| **243** | -0.85 % | 0 | 1 | 1 | 1 |
| **244** | -1.06 % | 0 | 0 | 1 | 1 |
| **245** | 0.81 % | 1 | 0 | 0 | 1 |
| **246** | -0.45 % | 0 | 1 | 1 | 0 |
| **247** | -0.46 % | 0 | 0 | 1 | 1 |
| **248** | -0.09 % | 0 | 1 | 1 | 0 |
| **249** | -0.56 % | 0 | 1 | 1 | 0 |
| **250** | 0.19 % | 1 | 0 | 1 | 1 |
| **251** | 0.50 % | 1 | 1 | 1 | 0 |
| **252** | -0.09 % | 0 | 1 | 1 | 0 |
| **253** | 0.03 % | 1 | 1 | 1 | 0 |
| **254** | -0.15 % | 0 | 0 | 1 | 0 |
| **255** | -0.43 % | 0 | 1 | 1 | 0 |
| **256** | -1.20 % | 0 | 1 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **257** | -0.50 % | 0 | 1 | 0 | 0 |
| **258** | 1.13 % | 1 | 1 | 1 | 0 |
| **259** | -0.25 % | 0 | 0 | 1 | 1 |
| **260** | 0.55 % | 1 | 1 | 1 | 0 |
| **261** | -0.32 % | 0 | 1 | 1 | 0 |
| **262** | 0.11 % | 1 | 0 | 1 | 0 |
| **263** | 0.60 % | 1 | 0 | 1 | 0 |
| **264** | 0.86 % | 1 | 0 | 1 | 1 |
| **265** | 0.86 % | 1 | 1 | 0 | 0 |
| **266** | 0.75 % | 1 | 1 | 1 | 1 |
| **267** | 0.24 % | 1 | 0 | 0 | 0 |
| **268** | 0.50 % | 1 | 0 | 1 | 1 |
| **269** | 0.96 % | 1 | 0 | 1 | 0 |
| **270** | 0.28 % | 1 | 0 | 1 | 0 |
| **271** | 0.17 % | 1 | 1 | 1 | 1 |
| **272** | -0.79 % | 0 | 1 | 1 | 0 |

Appendix 4. Comparison of predictions for the S&P 500 data

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **1** | 0.84 % | 1 | 1 | 1 | 1 |
| **2** | 1.09 % | 1 | 0 | 1 | 1 |
| **3** | -3.23 % | 0 | 0 | 1 | 1 |
| **4** | -0.12 % | 0 | 0 | 0 | 1 |
| **5** | -2.32 % | 0 | 1 | 1 | 1 |
| **6** | 0.18 % | 1 | 1 | 0 | 1 |
| **7** | -0.03 % | 0 | 1 | 0 | 0 |
| **8** | 0.54 % | 1 | 1 | 1 | 1 |
| **9** | 0.00 % | 0 | 1 | 0 | 0 |
| **10** | -1.90 % | 0 | 0 | 1 | 1 |
| **11** | -2.07 % | 0 | 0 | 1 | 0 |
| **12** | 0.02 % | 1 | 1 | 0 | 1 |
| **13** | -1.53 % | 0 | 1 | 1 | 0 |
| **14** | -1.57 % | 0 | 1 | 1 | 1 |
| **15** | -2.06 % | 0 | 0 | 1 | 0 |
| **16** | -2.70 % | 0 | 0 | 1 | 1 |
| **17** | 4.96 % | 1 | 0 | 0 | 1 |
| **18** | 0.87 % | 1 | 0 | 1 | 0 |
| **19** | -0.11 % | 0 | 1 | 1 | 0 |
| **20** | 0.86 % | 1 | 1 | 1 | 1 |
| **21** | 0.13 % | 1 | 0 | 1 | 1 |
| **22** | -2.45 % | 0 | 0 | 1 | 1 |
| **23** | 3.43 % | 1 | 1 | 1 | 0 |
| **24** | 0.70 % | 1 | 1 | 1 | 0 |
| **25** | 0.97 % | 1 | 0 | 0 | 1 |
| **26** | 0.44 % | 1 | 0 | 0 | 1 |
| **27** | 0.45 % | 1 | 0 | 0 | 1 |
| **28** | -0.01 % | 0 | 0 | 1 | 0 |
| **29** | -0.51 % | 0 | 0 | 1 | 1 |
| **30** | 1.07 % | 1 | 0 | 1 | 1 |
| **31** | 0.22 % | 1 | 0 | 1 | 0 |
| **32** | 0.77 % | 1 | 0 | 1 | 1 |
| **33** | 1.32 % | 1 | 0 | 1 | 0 |
| **34** | -1.41 % | 0 | 0 | 1 | 0 |
| **35** | 0.22 % | 1 | 0 | 0 | 1 |
| **36** | 0.14 % | 1 | 0 | 1 | 1 |
| **37** | 0.85 % | 1 | 0 | 1 | 0 |
| **38** | -0.78 % | 0 | 0 | 0 | 1 |
| **39** | -0.14 % | 0 | 1 | 1 | 0 |
| **40** | 1.57 % | 1 | 1 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **41** | 0.88 % | 1 | 1 | 0 | 1 |
| **42** | 0.10 % | 1 | 1 | 0 | 0 |
| **43** | 0.68 % | 1 | 0 | 1 | 1 |
| **44** | 0.47 % | 1 | 0 | 1 | 1 |
| **45** | -0.21 % | 0 | 0 | 0 | 0 |
| **46** | -0.92 % | 0 | 1 | 1 | 1 |
| **47** | 0.10 % | 1 | 1 | 1 | 0 |
| **48** | 0.07 % | 1 | 1 | 1 | 0 |
| **49** | 1.30 % | 1 | 0 | 1 | 0 |
| **50** | 0.31 % | 1 | 1 | 0 | 0 |
| **51** | -0.23 % | 0 | 1 | 1 | 0 |
| **52** | 1.10 % | 1 | 0 | 1 | 1 |
| **53** | 0.16 % | 1 | 0 | 0 | 0 |
| **54** | 0.20 % | 1 | 0 | 1 | 0 |
| **55** | -0.34 % | 0 | 1 | 1 | 0 |
| **56** | 0.64 % | 1 | 1 | 1 | 1 |
| **57** | 0.14 % | 1 | 0 | 1 | 0 |
| **58** | -0.08 % | 0 | 0 | 1 | 0 |
| **59** | -0.04 % | 0 | 1 | 1 | 1 |
| **60** | -0.25 % | 0 | 1 | 1 | 0 |
| **61** | 0.70 % | 1 | 0 | 1 | 1 |
| **62** | -0.39 % | 0 | 0 | 1 | 0 |
| **63** | -0.11 % | 0 | 0 | 0 | 0 |
| **64** | -0.65 % | 0 | 1 | 1 | 0 |
| **65** | -0.79 % | 0 | 0 | 1 | 0 |
| **66** | -0.20 % | 0 | 1 | 1 | 0 |
| **67** | 1.47 % | 1 | 1 | 1 | 1 |
| **68** | 0.30 % | 1 | 0 | 1 | 0 |
| **69** | 0.70 % | 1 | 1 | 1 | 1 |
| **70** | -0.05 % | 0 | 1 | 1 | 0 |
| **71** | 0.50 % | 1 | 0 | 1 | 0 |
| **72** | 0.37 % | 1 | 0 | 1 | 0 |
| **73** | -0.01 % | 0 | 0 | 1 | 0 |
| **74** | -0.29 % | 0 | 1 | 1 | 1 |
| **75** | 1.09 % | 1 | 1 | 1 | 1 |
| **76** | -1.89 % | 0 | 0 | 1 | 0 |
| **77** | -0.08 % | 0 | 0 | 0 | 1 |
| **78** | 0.72 % | 1 | 1 | 1 | 0 |
| **79** | -0.46 % | 0 | 0 | 0 | 1 |
| **80** | 0.37 % | 1 | 1 | 1 | 1 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **81** | 0.68 % | 1 | 1 | 1 | 1 |
| **82** | 1.16 % | 1 | 0 | 1 | 0 |
| **83** | 0.01 % | 1 | 1 | 0 | 0 |
| **84** | 0.21 % | 1 | 1 | 1 | 0 |
| **85** | 0.23 % | 1 | 1 | 1 | 1 |
| **86** | 0.46 % | 1 | 0 | 1 | 0 |
| **87** | 0.11 % | 1 | 0 | 1 | 0 |
| **88** | -0.58 % | 0 | 1 | 1 | 0 |
| **89** | 0.36 % | 1 | 1 | 1 | 0 |
| **90** | 0.01 % | 1 | 1 | 1 | 0 |
| **91** | 0.67 % | 1 | 0 | 1 | 1 |
| **92** | -0.06 % | 0 | 0 | 1 | 0 |
| **93** | 0.05 % | 1 | 0 | 1 | 1 |
| **94** | -0.22 % | 0 | 0 | 1 | 1 |
| **95** | 0.16 % | 1 | 0 | 1 | 0 |
| **96** | 0.10 % | 1 | 1 | 1 | 0 |
| **97** | 0.89 % | 1 | 0 | 1 | 1 |
| **98** | -0.22 % | 0 | 1 | 1 | 1 |
| **99** | -0.04 % | 0 | 1 | 1 | 0 |
| **100** | 0.47 % | 1 | 1 | 1 | 0 |
| **101** | 0.11 % | 1 | 0 | 1 | 1 |
| **102** | 0.10 % | 1 | 0 | 0 | 0 |
| **103** | -0.75 % | 0 | 1 | 1 | 0 |
| **104** | -0.21 % | 0 | 0 | 1 | 1 |
| **105** | 0.97 % | 1 | 1 | 1 | 1 |
| **106** | -0.44 % | 0 | 0 | 0 | 0 |
| **107** | -1.65 % | 0 | 1 | 1 | 0 |
| **108** | -0.16 % | 0 | 0 | 1 | 0 |
| **109** | -0.27 % | 0 | 1 | 1 | 1 |
| **110** | 0.41 % | 1 | 1 | 1 | 1 |
| **111** | -2.41 % | 0 | 1 | 1 | 0 |
| **112** | 0.81 % | 1 | 0 | 1 | 1 |
| **113** | 0.60 % | 1 | 1 | 1 | 1 |
| **114** | 0.92 % | 1 | 1 | 1 | 1 |
| **115** | -0.57 % | 0 | 1 | 1 | 0 |
| **116** | -0.67 % | 0 | 1 | 1 | 0 |
| **117** | 0.85 % | 1 | 1 | 0 | 1 |
| **118** | -0.28 % | 0 | 0 | 0 | 0 |
| **119** | -1.18 % | 0 | 1 | 1 | 0 |
| **120** | 0.15 % | 1 | 1 | 1 | 1 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **121** | -0.84 % | 0 | 1 | 0 | 0 |
| **122** | -0.69 % | 0 | 0 | 1 | 1 |
| **123** | 0.22 % | 1 | 1 | 1 | 0 |
| **124** | -1.30 % | 0 | 1 | 1 | 0 |
| **125** | -0.28 % | 0 | 0 | 0 | 0 |
| **126** | 2.15 % | 1 | 1 | 1 | 1 |
| **127** | 0.83 % | 1 | 1 | 1 | 1 |
| **128** | 0.64 % | 1 | 1 | 1 | 1 |
| **129** | 1.06 % | 1 | 1 | 0 | 1 |
| **130** | 0.47 % | 1 | 0 | 0 | 1 |
| **131** | -0.03 % | 0 | 0 | 0 | 1 |
| **132** | -0.20 % | 0 | 1 | 1 | 1 |
| **133** | 0.44 % | 1 | 1 | 1 | 0 |
| **134** | -0.15 % | 0 | 0 | 1 | 0 |
| **135** | 0.09 % | 1 | 1 | 1 | 0 |
| **136** | 0.97 % | 1 | 1 | 1 | 1 |
| **137** | 0.30 % | 1 | 1 | 0 | 1 |
| **138** | 0.96 % | 1 | 1 | 1 | 1 |
| **139** | -0.12 % | 0 | 0 | 0 | 0 |
| **140** | -0.17 % | 0 | 0 | 0 | 0 |
| **141** | -0.95 % | 0 | 0 | 1 | 1 |
| **142** | -0.12 % | 0 | 0 | 0 | 0 |
| **143** | 0.40 % | 1 | 1 | 1 | 1 |
| **144** | 0.58 % | 1 | 0 | 1 | 0 |
| **145** | 0.77 % | 1 | 1 | 1 | 0 |
| **146** | 0.30 % | 1 | 1 | 0 | 1 |
| **147** | 0.79 % | 1 | 1 | 1 | 0 |
| **148** | -0.17 % | 0 | 0 | 0 | 1 |
| **149** | -0.48 % | 0 | 0 | 1 | 1 |
| **150** | 0.15 % | 1 | 0 | 0 | 0 |
| **151** | 0.45 % | 1 | 1 | 1 | 1 |
| **152** | 0.23 % | 1 | 0 | 0 | 0 |
| **153** | 0.47 % | 1 | 1 | 1 | 0 |
| **154** | 0.02 % | 1 | 1 | 1 | 1 |
| **155** | -0.34 % | 0 | 0 | 1 | 0 |
| **156** | -0.65 % | 0 | 0 | 1 | 0 |
| **157** | 0.37 % | 1 | 0 | 1 | 0 |
| **158** | -0.61 % | 0 | 1 | 1 | 0 |
| **159** | 0.29 % | 1 | 0 | 1 | 1 |
| **160** | 0.69 % | 1 | 1 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **161** | 0.47 % | 1 | 1 | 1 | 1 |
| **162** | -0.53 % | 0 | 1 | 1 | 0 |
| **163** | 0.74 % | 1 | 0 | 1 | 1 |
| **164** | -0.16 % | 0 | 0 | 1 | 0 |
| **165** | -0.25 % | 0 | 0 | 1 | 1 |
| **166** | -1.09 % | 0 | 1 | 1 | 0 |
| **167** | -0.89 % | 0 | 1 | 0 | 1 |
| **168** | -0.72 % | 0 | 0 | 1 | 1 |
| **169** | -2.97 % | 0 | 1 | 1 | 1 |
| **170** | 1.31 % | 1 | 1 | 1 | 0 |
| **171** | 0.08 % | 1 | 0 | 1 | 1 |
| **172** | 1.90 % | 1 | 1 | 1 | 0 |
| **173** | -0.65 % | 0 | 0 | 1 | 1 |
| **174** | -1.18 % | 0 | 1 | 0 | 1 |
| **175** | 1.48 % | 1 | 0 | 0 | 0 |
| **176** | -2.90 % | 0 | 0 | 1 | 0 |
| **177** | 0.27 % | 1 | 0 | 1 | 0 |
| **178** | 1.46 % | 1 | 1 | 0 | 0 |
| **179** | 1.22 % | 1 | 0 | 1 | 1 |
| **180** | -0.78 % | 0 | 1 | 1 | 0 |
| **181** | 0.83 % | 1 | 1 | 0 | 0 |
| **182** | -0.05 % | 0 | 0 | 1 | 0 |
| **183** | -2.59 % | 0 | 0 | 1 | 0 |
| **184** | 1.11 % | 1 | 1 | 0 | 1 |
| **185** | -0.32 % | 0 | 1 | 1 | 0 |
| **186** | 0.66 % | 1 | 0 | 1 | 1 |
| **187** | 1.29 % | 1 | 1 | 1 | 1 |
| **188** | 0.08 % | 1 | 1 | 1 | 1 |
| **189** | -0.68 % | 0 | 1 | 1 | 1 |
| **190** | 1.09 % | 1 | 0 | 1 | 1 |
| **191** | 1.32 % | 1 | 0 | 0 | 0 |
| **192** | 0.09 % | 1 | 0 | 1 | 1 |
| **193** | 0.00 % | 0 | 0 | 1 | 0 |
| **194** | 0.04 % | 1 | 1 | 1 | 0 |
| **195** | 0.73 % | 1 | 0 | 1 | 0 |
| **196** | 0.30 % | 1 | 0 | 0 | 0 |
| **197** | -0.04 % | 0 | 0 | 0 | 1 |
| **198** | -0.31 % | 0 | 1 | 1 | 1 |
| **199** | 0.26 % | 1 | 0 | 1 | 0 |
| **200** | 0.03 % | 1 | 0 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **201** | 0.01 % | 1 | 1 | 1 | 0 |
| **202** | -0.48 % | 0 | 0 | 0 | 0 |
| **203** | -0.01 % | 0 | 0 | 1 | 0 |
| **204** | -0.83 % | 0 | 1 | 1 | 1 |
| **205** | 0.62 % | 1 | 0 | 0 | 0 |
| **206** | -0.24 % | 0 | 1 | 1 | 0 |
| **207** | -0.52 % | 0 | 0 | 1 | 0 |
| **208** | 0.51 % | 1 | 1 | 1 | 1 |
| **209** | -1.22 % | 0 | 1 | 1 | 0 |
| **210** | -1.79 % | 0 | 0 | 0 | 1 |
| **211** | 0.82 % | 1 | 1 | 1 | 0 |
| **212** | 1.42 % | 1 | 1 | 1 | 0 |
| **213** | -0.45 % | 0 | 0 | 1 | 1 |
| **214** | -1.55 % | 0 | 1 | 0 | 1 |
| **215** | 0.94 % | 1 | 0 | 1 | 0 |
| **216** | 0.64 % | 1 | 1 | 1 | 1 |
| **217** | 1.10 % | 1 | 0 | 1 | 1 |
| **218** | -0.14 % | 0 | 1 | 1 | 1 |
| **219** | 1.00 % | 1 | 1 | 0 | 0 |
| **220** | -0.19 % | 0 | 1 | 1 | 0 |
| **221** | 0.28 % | 1 | 0 | 1 | 0 |
| **222** | -0.39 % | 0 | 0 | 1 | 0 |
| **223** | 0.69 % | 1 | 1 | 1 | 1 |
| **224** | -0.35 % | 0 | 0 | 1 | 0 |
| **225** | 0.29 % | 1 | 0 | 1 | 1 |
| **226** | 0.19 % | 1 | 1 | 1 | 1 |
| **227** | 0.41 % | 1 | 0 | 1 | 0 |
| **228** | 0.56 % | 1 | 0 | 1 | 0 |
| **229** | -0.08 % | 0 | 0 | 0 | 1 |
| **230** | 0.33 % | 1 | 0 | 1 | 1 |
| **231** | -0.30 % | 0 | 0 | 1 | 0 |
| **232** | 0.98 % | 1 | 0 | 1 | 1 |
| **233** | 0.37 % | 1 | 0 | 0 | 1 |
| **234** | -0.12 % | 0 | 0 | 1 | 0 |
| **235** | 0.08 % | 1 | 0 | 1 | 0 |
| **236** | 0.32 % | 1 | 1 | 1 | 0 |
| **237** | 0.27 % | 1 | 0 | 0 | 0 |
| **238** | -0.20 % | 0 | 1 | 1 | 0 |
| **239** | 0.16 % | 1 | 1 | 1 | 0 |
| **240** | 0.07 % | 1 | 0 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **241** | 0.11 % | 1 | 1 | 1 | 1 |
| **242** | 0.79 % | 1 | 0 | 1 | 1 |
| **243** | 0.05 % | 1 | 0 | 0 | 0 |
| **244** | -0.05 % | 0 | 1 | 1 | 0 |
| **245** | -0.36 % | 0 | 0 | 1 | 1 |
| **246** | -0.15 % | 0 | 0 | 1 | 1 |
| **247** | 0.22 % | 1 | 0 | 1 | 1 |
| **248** | 0.76 % | 1 | 0 | 1 | 1 |
| **249** | 0.22 % | 1 | 1 | 1 | 1 |
| **250** | 0.43 % | 1 | 1 | 1 | 1 |
| **251** | -0.38 % | 0 | 0 | 1 | 1 |
| **252** | -0.86 % | 0 | 0 | 1 | 0 |
| **253** | -0.66 % | 0 | 0 | 0 | 0 |
| **254** | 0.64 % | 1 | 1 | 0 | 1 |
| **255** | 0.18 % | 1 | 0 | 0 | 1 |
| **256** | 0.92 % | 1 | 1 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **257** | -0.31 % | 0 | 1 | 0 | 1 |
| **258** | -0.11 % | 0 | 0 | 1 | 0 |
| **259** | 0.29 % | 1 | 0 | 1 | 1 |
| **260** | 0.87 % | 1 | 0 | 1 | 0 |
| **261** | 0.02 % | 1 | 1 | 1 | 1 |
| **262** | 0.72 % | 1 | 1 | 1 | 0 |
| **263** | 0.03 % | 1 | 0 | 0 | 0 |
| **264** | -0.03 % | 0 | 0 | 1 | 0 |
| **265** | 0.45 % | 1 | 0 | 1 | 0 |
| **266** | 0.50 % | 1 | 1 | 1 | 0 |
| **267** | 0.09 % | 1 | 1 | 1 | 1 |
| **268** | -0.01 % | 0 | 0 | 1 | 1 |
| **269** | 0.52 % | 1 | 0 | 1 | 0 |
| **270** | 0.00 % | 1 | 0 | 1 | 1 |
| **271** | -0.56 % | 0 | 0 | 1 | 0 |
| **272** | 0.30 % | 1 | 0 | 1 | 1 |

Appendix 5. Comparison of predictions for the FTSE 100 data

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| 1 | 1.18 % | 1 | 0 | 0 | 0 |
| 2 | -0.56 % | 0 | 0 | 1 | 0 |
| 3 | -1.44 % | 0 | 1 | 0 | 1 |
| 4 | -3.14 % | 0 | 0 | 0 | 1 |
| 5 | 1.10 % | 1 | 0 | 1 | 0 |
| 6 | -0.83 % | 0 | 1 | 1 | 0 |
| 7 | 1.27 % | 1 | 0 | 0 | 1 |
| 8 | 1.08 % | 1 | 0 | 1 | 0 |
| 9 | -0.02 % | 0 | 1 | 0 | 1 |
| 10 | -0.47 % | 0 | 1 | 1 | 1 |
| 11 | -1.05 % | 0 | 0 | 1 | 0 |
| 12 | -1.06 % | 0 | 0 | 0 | 1 |
| 13 | 0.96 % | 1 | 1 | 1 | 0 |
| 14 | -0.79 % | 0 | 0 | 1 | 0 |
| 15 | 0.14 % | 1 | 1 | 0 | 0 |
| 16 | -0.52 % | 0 | 1 | 1 | 1 |
| 17 | -1.43 % | 0 | 0 | 0 | 0 |
| 18 | 2.27 % | 1 | 0 | 1 | 1 |
| 19 | -0.09 % | 0 | 1 | 1 | 0 |
| 20 | 0.09 % | 1 | 0 | 1 | 0 |
| 21 | -0.61 % | 0 | 1 | 1 | 0 |
| 22 | 2.16 % | 1 | 0 | 0 | 1 |
| 23 | -0.39 % | 0 | 0 | 0 | 0 |
| 24 | 0.74 % | 1 | 0 | 1 | 0 |
| 25 | 0.66 % | 1 | 1 | 1 | 0 |
| 26 | 0.52 % | 1 | 1 | 0 | 0 |
| 27 | -0.36 % | 0 | 0 | 1 | 1 |
| 28 | -0.91 % | 0 | 0 | 0 | 0 |
| 29 | 0.58 % | 1 | 1 | 0 | 0 |
| 30 | -0.47 % | 0 | 1 | 1 | 0 |
| 31 | -0.36 % | 0 | 0 | 1 | 0 |
| 32 | 1.95 % | 1 | 1 | 1 | 1 |
| 33 | 0.03 % | 1 | 1 | 0 | 0 |
| 34 | -0.99 % | 0 | 1 | 1 | 1 |
| 35 | -0.85 % | 0 | 0 | 0 | 0 |
| 36 | -0.35 % | 0 | 0 | 1 | 0 |
| 37 | -0.14 % | 0 | 0 | 1 | 1 |
| 38 | -0.91 % | 0 | 0 | 1 | 0 |
| 39 | 1.29 % | 1 | 1 | 0 | 1 |
| 40 | 1.58 % | 1 | 1 | 1 | 0 |
| 41 | 0.39 % | 1 | 1 | 1 | 1 |
| 42 | 0.74 % | 1 | 1 | 1 | 1 |
| 43 | 0.20 % | 1 | 0 | 1 | 0 |
| 44 | 2.04 % | 1 | 0 | 1 | 0 |
| 45 | -0.06 % | 0 | 0 | 0 | 1 |
| 46 | -1.10 % | 0 | 1 | 1 | 1 |
| 47 | -0.32 % | 0 | 0 | 0 | 0 |
| 48 | 0.82 % | 1 | 0 | 1 | 0 |
| 49 | 0.06 % | 1 | 0 | 1 | 0 |
| 50 | 0.82 % | 1 | 1 | 1 | 0 |
| 51 | 0.35 % | 1 | 1 | 1 | 0 |
| 52 | 0.55 % | 1 | 0 | 1 | 1 |
| 53 | -0.24 % | 0 | 0 | 1 | 0 |
| 54 | -0.56 % | 0 | 0 | 1 | 1 |
| 55 | 0.69 % | 1 | 0 | 1 | 1 |
| 56 | -0.57 % | 0 | 1 | 1 | 0 |
| 57 | 0.16 % | 1 | 0 | 1 | 1 |
| 58 | 0.07 % | 1 | 1 | 1 | 1 |
| 59 | -0.45 % | 0 | 1 | 1 | 1 |
| 60 | -0.61 % | 0 | 1 | 1 | 0 |
| 61 | -0.26 % | 0 | 1 | 1 | 1 |
| 62 | 0.45 % | 1 | 1 | 1 | 0 |
| 63 | 0.39 % | 1 | 1 | 1 | 0 |
| 64 | 0.69 % | 1 | 1 | 1 | 0 |
| 65 | 0.18 % | 1 | 1 | 1 | 0 |
| 66 | -0.35 % | 0 | 1 | 1 | 0 |
| 67 | -0.74 % | 0 | 1 | 1 | 0 |
| 68 | 0.37 % | 1 | 0 | 1 | 0 |
| 69 | 0.29 % | 1 | 0 | 1 | 1 |
| 70 | 0.11 % | 1 | 1 | 1 | 1 |
| 71 | 0.42 % | 1 | 1 | 1 | 1 |
| 72 | 0.60 % | 1 | 1 | 1 | 1 |
| 73 | 0.98 % | 1 | 0 | 1 | 0 |
| 74 | 0.34 % | 1 | 1 | 1 | 1 |
| 75 | -0.45 % | 0 | 1 | 0 | 1 |
| 76 | 0.97 % | 1 | 0 | 1 | 0 |
| 77 | -2.01 % | 0 | 0 | 1 | 1 |
| 78 | -0.42 % | 0 | 0 | 0 | 0 |
| 79 | 0.26 % | 1 | 1 | 1 | 1 |
| 80 | -0.03 % | 0 | 0 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess | Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **81** | 0.62 % | 1 | 1 | 1 | 1 | **121** | -1.15 % | 0 | 0 | 1 | 1 |
| **82** | 0.62 % | 1 | 1 | 1 | 0 | **122** | 0.53 % | 1 | 0 | 1 | 1 |
| **83** | 0.52 % | 1 | 1 | 1 | 0 | **123** | -0.78 % | 0 | 1 | 1 | 0 |
| **84** | 1.01 % | 1 | 0 | 0 | 0 | **124** | 0.32 % | 1 | 0 | 0 | 1 |
| **85** | 0.37 % | 1 | 1 | 1 | 0 | **125** | 0.41 % | 1 | 1 | 1 | 1 |
| **86** | -0.08 % | 0 | 1 | 1 | 1 | **126** | 0.08 % | 1 | 1 | 0 | 0 |
| **87** | 0.61 % | 1 | 1 | 1 | 0 | **127** | 0.65 % | 1 | 1 | 0 | 0 |
| **88** | 0.07 % | 1 | 0 | 1 | 0 | **128** | 0.99 % | 1 | 1 | 1 | 1 |
| **89** | -0.35 % | 0 | 0 | 1 | 1 | **129** | 0.59 % | 1 | 1 | 0 | 0 |
| **90** | -0.05 % | 0 | 1 | 1 | 0 | **130** | 0.31 % | 1 | 1 | 0 | 0 |
| **91** | 0.05 % | 1 | 1 | 1 | 1 | **131** | -0.42 % | 0 | 0 | 1 | 1 |
| **92** | 0.26 % | 1 | 0 | 1 | 0 | **132** | 0.07 % | 1 | 1 | 1 | 1 |
| **93** | 0.00 % | 0 | 0 | 1 | 1 | **133** | -0.31 % | 0 | 0 | 1 | 0 |
| **94** | 0.44 % | 1 | 1 | 1 | 0 | **134** | 0.16 % | 1 | 1 | 1 | 0 |
| **95** | 0.02 % | 1 | 1 | 1 | 1 | **135** | 1.17 % | 1 | 1 | 1 | 0 |
| **96** | -0.10 % | 0 | 0 | 1 | 0 | **136** | -0.53 % | 0 | 1 | 1 | 1 |
| **97** | 0.85 % | 1 | 0 | 1 | 0 | **137** | 0.31 % | 1 | 0 | 0 | 0 |
| **98** | -0.68 % | 0 | 0 | 1 | 0 | **138** | -0.23 % | 0 | 0 | 1 | 1 |
| **99** | -0.40 % | 0 | 0 | 0 | 0 | **139** | 0.12 % | 1 | 0 | 1 | 1 |
| **100** | -0.08 % | 0 | 1 | 1 | 1 | **140** | 0.08 % | 1 | 1 | 1 | 1 |
| **101** | 0.17 % | 1 | 0 | 1 | 1 | **141** | -0.08 % | 0 | 1 | 1 | 1 |
| **102** | -0.30 % | 0 | 0 | 1 | 1 | **142** | -0.10 % | 0 | 0 | 1 | 0 |
| **103** | -0.44 % | 0 | 1 | 1 | 1 | **143** | 0.31 % | 1 | 1 | 1 | 1 |
| **104** | -0.40 % | 0 | 1 | 1 | 1 | **144** | 0.97 % | 1 | 1 | 1 | 0 |
| **105** | 0.40 % | 1 | 1 | 1 | 1 | **145** | 0.82 % | 1 | 0 | 1 | 0 |
| **106** | -1.63 % | 0 | 1 | 1 | 0 | **146** | 0.66 % | 1 | 1 | 0 | 1 |
| **107** | 0.15 % | 1 | 0 | 0 | 1 | **147** | -0.05 % | 0 | 1 | 0 | 0 |
| **108** | -0.75 % | 0 | 1 | 0 | 1 | **148** | -0.66 % | 0 | 0 | 1 | 1 |
| **109** | -0.06 % | 0 | 0 | 1 | 0 | **149** | -0.05 % | 0 | 0 | 1 | 1 |
| **110** | -0.55 % | 0 | 1 | 1 | 1 | **150** | -0.17 % | 0 | 1 | 1 | 1 |
| **111** | 1.09 % | 1 | 1 | 0 | 0 | **151** | -0.08 % | 0 | 1 | 1 | 1 |
| **112** | 0.76 % | 1 | 1 | 1 | 0 | **152** | -0.28 % | 0 | 1 | 1 | 1 |
| **113** | 1.09 % | 1 | 1 | 0 | 0 | **153** | -0.05 % | 0 | 1 | 1 | 0 |
| **114** | -0.07 % | 0 | 1 | 1 | 1 | **154** | 0.34 % | 1 | 1 | 1 | 1 |
| **115** | -0.51 % | 0 | 1 | 1 | 0 | **155** | 0.60 % | 1 | 1 | 1 | 1 |
| **116** | 0.25 % | 1 | 0 | 1 | 1 | **156** | -0.55 % | 0 | 1 | 1 | 0 |
| **117** | 0.07 % | 1 | 0 | 1 | 1 | **157** | -0.56 % | 0 | 0 | 1 | 0 |
| **118** | -1.37 % | 0 | 1 | 1 | 1 | **158** | 0.21 % | 1 | 0 | 1 | 0 |
| **119** | 0.65 % | 1 | 1 | 0 | 0 | **159** | 0.08 % | 1 | 1 | 1 | 0 |
| **120** | -0.12 % | 0 | 1 | 1 | 0 | **160** | 0.56 % | 1 | 1 | 1 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **161** | -0.73 % | 0 | 1 | 1 | 1 |
| **162** | -0.13 % | 0 | 0 | 0 | 0 |
| **163** | 0.80 % | 1 | 0 | 1 | 0 |
| **164** | 1.82 % | 1 | 0 | 1 | 1 |
| **165** | -0.52 % | 0 | 1 | 1 | 0 |
| **166** | -0.78 % | 0 | 1 | 1 | 1 |
| **167** | -0.01 % | 0 | 0 | 0 | 0 |
| **168** | -2.34 % | 0 | 0 | 1 | 1 |
| **169** | -2.47 % | 0 | 1 | 0 | 0 |
| **170** | -0.72 % | 0 | 1 | 1 | 1 |
| **171** | 0.38 % | 1 | 1 | 0 | 0 |
| **172** | 1.71 % | 1 | 0 | 1 | 1 |
| **173** | -0.44 % | 0 | 1 | 0 | 0 |
| **174** | -0.37 % | 0 | 1 | 0 | 1 |
| **175** | 0.33 % | 1 | 0 | 1 | 0 |
| **176** | -1.42 % | 0 | 0 | 1 | 0 |
| **177** | -0.76 % | 0 | 0 | 0 | 0 |
| **178** | 0.71 % | 1 | 1 | 0 | 1 |
| **179** | 1.02 % | 1 | 0 | 1 | 0 |
| **180** | -0.90 % | 0 | 1 | 0 | 0 |
| **181** | 1.11 % | 1 | 1 | 0 | 0 |
| **182** | -0.97 % | 0 | 1 | 1 | 1 |
| **183** | -0.47 % | 0 | 0 | 0 | 0 |
| **184** | -0.08 % | 0 | 0 | 1 | 0 |
| **185** | 0.35 % | 1 | 0 | 1 | 0 |
| **186** | 0.99 % | 1 | 0 | 1 | 1 |
| **187** | 0.32 % | 1 | 1 | 0 | 1 |
| **188** | 1.04 % | 1 | 1 | 0 | 0 |
| **189** | -0.19 % | 0 | 0 | 1 | 1 |
| **190** | 0.59 % | 1 | 0 | 1 | 0 |
| **191** | -0.39 % | 0 | 0 | 1 | 1 |
| **192** | 0.15 % | 1 | 0 | 1 | 1 |
| **193** | -0.64 % | 0 | 0 | 1 | 1 |
| **194** | 0.44 % | 1 | 0 | 0 | 0 |
| **195** | 0.96 % | 1 | 1 | 1 | 1 |
| **196** | 0.09 % | 1 | 0 | 0 | 0 |
| **197** | 0.31 % | 1 | 1 | 1 | 1 |
| **198** | -0.62 % | 0 | 1 | 1 | 0 |
| **199** | -0.01 % | 0 | 0 | 1 | 1 |
| **200** | -0.09 % | 0 | 0 | 1 | 1 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **201** | 0.58 % | 1 | 0 | 1 | 0 |
| **202** | -0.16 % | 0 | 1 | 1 | 1 |
| **203** | -0.26 % | 0 | 1 | 1 | 0 |
| **204** | -0.47 % | 0 | 0 | 1 | 0 |
| **205** | -0.02 % | 0 | 0 | 1 | 1 |
| **206** | 0.85 % | 1 | 1 | 1 | 0 |
| **207** | 1.02 % | 1 | 0 | 0 | 1 |
| **208** | -0.24 % | 0 | 1 | 0 | 0 |
| **209** | -0.65 % | 0 | 1 | 1 | 1 |
| **210** | -3.23 % | 0 | 0 | 1 | 0 |
| **211** | -0.52 % | 0 | 0 | 0 | 0 |
| **212** | 1.10 % | 1 | 1 | 1 | 0 |
| **213** | 0.59 % | 1 | 0 | 0 | 0 |
| **214** | -0.76 % | 0 | 1 | 0 | 1 |
| **215** | 0.33 % | 1 | 1 | 0 | 0 |
| **216** | 0.40 % | 1 | 1 | 1 | 1 |
| **217** | 0.84 % | 1 | 0 | 1 | 1 |
| **218** | -0.46 % | 0 | 0 | 1 | 1 |
| **219** | -0.03 % | 0 | 0 | 0 | 0 |
| **220** | -0.61 % | 0 | 1 | 1 | 0 |
| **221** | 0.22 % | 1 | 0 | 1 | 1 |
| **222** | -0.44 % | 0 | 0 | 1 | 0 |
| **223** | 0.18 % | 1 | 0 | 0 | 0 |
| **224** | 0.68 % | 1 | 0 | 1 | 1 |
| **225** | 0.67 % | 1 | 1 | 1 | 0 |
| **226** | 0.95 % | 1 | 1 | 1 | 0 |
| **227** | -0.05 % | 0 | 0 | 0 | 1 |
| **228** | 0.09 % | 1 | 0 | 1 | 1 |
| **229** | -0.34 % | 0 | 0 | 1 | 1 |
| **230** | 0.34 % | 1 | 0 | 1 | 1 |
| **231** | -1.10 % | 0 | 1 | 1 | 1 |
| **232** | 0.75 % | 1 | 0 | 0 | 1 |
| **233** | 0.92 % | 1 | 1 | 1 | 0 |
| **234** | 0.25 % | 1 | 0 | 0 | 1 |
| **235** | 0.12 % | 1 | 1 | 1 | 1 |
| **236** | 0.22 % | 1 | 1 | 1 | 0 |
| **237** | -0.64 % | 0 | 0 | 0 | 1 |
| **238** | -0.42 % | 0 | 0 | 1 | 0 |
| **239** | 0.50 % | 1 | 1 | 1 | 0 |
| **240** | -0.19 % | 0 | 0 | 0 | 0 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **241** | -0.59 % | 0 | 1 | 1 | 0 |
| **242** | 0.14 % | 1 | 0 | 1 | 1 |
| **243** | 0.07 % | 1 | 1 | 1 | 0 |
| **244** | 0.22 % | 1 | 0 | 1 | 1 |
| **245** | -0.84 % | 0 | 0 | 0 | 1 |
| **246** | -0.28 % | 0 | 0 | 1 | 0 |
| **247** | 1.22 % | 1 | 1 | 1 | 1 |
| **248** | 0.95 % | 1 | 0 | 1 | 1 |
| **249** | 0.09 % | 1 | 1 | 0 | 1 |
| **250** | 0.36 % | 1 | 1 | 1 | 0 |
| **251** | -0.07 % | 0 | 0 | 1 | 1 |
| **252** | -0.94 % | 0 | 0 | 1 | 0 |
| **253** | -0.82 % | 0 | 1 | 1 | 1 |
| **254** | -1.75 % | 0 | 1 | 1 | 0 |
| **255** | 0.42 % | 1 | 1 | 1 | 1 |
| **256** | -0.70 % | 0 | 1 | 1 | 1 |

| Day | Returns | Actual | LSTM | SVM | Random guess |
|---|---|---|---|---|---|
| **257** | 1.43 % | 1 | 0 | 1 | 0 |
| **258** | -0.08 % | 0 | 1 | 1 | 1 |
| **259** | -0.28 % | 0 | 1 | 1 | 0 |
| **260** | 0.03 % | 1 | 1 | 1 | 1 |
| **261** | 0.81 % | 1 | 0 | 0 | 0 |
| **262** | 1.10 % | 1 | 0 | 1 | 0 |
| **263** | 2.25 % | 1 | 1 | 1 | 1 |
| **264** | 0.08 % | 1 | 1 | 1 | 1 |
| **265** | 0.21 % | 1 | 1 | 1 | 0 |
| **266** | 0.45 % | 1 | 0 | 1 | 1 |
| **267** | 0.11 % | 1 | 0 | 1 | 0 |
| **268** | 0.54 % | 1 | 1 | 1 | 0 |
| **269** | 0.20 % | 1 | 1 | 1 | 1 |
| **270** | 0.17 % | 1 | 1 | 1 | 0 |
| **271** | -0.76 % | 0 | 0 | 1 | 1 |
| **272** | -0.59 % | 0 | 0 | 1 | 0 |