# *Can AI Predict the Stock Market? A CNN-BiLSTM Based Analysis of Macroeconomic Indicators' Effect on OMXS30*

Lorin Abdulaziz & Helios Rruci

## Abstract

Forecasting time series within the financial markets is one of the most fundamental subjects in economics. These markets are highly complex due to their nonlinear behavior, where uncertainty and unpredictability play a central role. Fluctuations occur without any clear patterns, resulting in more challenging investment decisions for all the market participants. Based on four domestic macroeconomic indicators, this thesis predicts the return of the OMXS30 index by implementing deep learning approaches, specifically the CNN-BiLSTM model. Analysis reveals that while increasing the model complexity by adding additional variables, the out-of-sample test performance $R^2_{OS}$ improved by 12.1 %. The observed results indicate the model's ability to explain 12.1 % of the variation in the OMXS30 index returns by including all the selected variables. Despite improvements in predictions for both the training and test data, the model still struggles with limited generalization, exhibiting a mild overfitting.

Bachelor's thesis in Economics, 15 credits
Fall Semester 2024
Supervisor: Charles Nadeau

Department of Economics
School of Business, Economics and Law
University of Gothenburg

## Acknowledgments

We sincerely thank our supervisor, Charles Nadeau, for his insightful feedback and guidance, which greatly enhanced the quality of this thesis.

# Table of Contents

# 1. Introduction

## 1.1 Background

The stock market is not just a financial system; it is a reflection of society's ambitions, fears, and progress. As both a driver of profitable economic opportunities and a source of financial volatility, the stock market stands at the center of a paradox in global finance. It is well characterized by uncertainty and unpredictability, but it remains the most recognizable financial market among numerous investors where large sums of money are traded daily. Hence, investors engage in transactions within the stock market, to achieve profitability. However, the profits are not always assured as external factors may affect the stock market performance.

The complex nature of the stock market is often influenced by various macroeconomic indicators as it tends to reflect the aggerate economy. Yet, which specific indicator impacts the most remains uncertain. To uncover the mechanisms underlying the interactions between the stock market and the macroeconomic indicators, numerous approaches have been developed in economics to perform such analyses. They seek to capture complex relationships and patterns within financial data, aiming to predict stock market performance.

Advanced machine learning, one of these methods, is a part of the field of artificial intelligence (AI) whose systems are described as “intelligent agents”. Artificial intelligence develops machines with intelligence comparable to humans, which are designed to achieve desired outcomes (Shiri et al., 2024). Machine learning operates as a framework for data analysis by building advanced models and algorithms to predict future outcomes based on historical data (El-Amir & Hamdy, 2020). In recent years, there has been a significant increase in interest in artificial intelligence (AI), specifically in deep learning, which is a subfield of machine learning. Unlike machine learning, where structured data and predefined algorithms are manually selected by humans to forecast future outcomes, deep learning utilizes artificial neural networks to learn independently from raw data through multiple layers. This in turn creates algorithms to mimic the human brain’s approach to predicting market outcomes (Taye, 2023; El Amir & Hamdy, 2020).

As is well established, predicting the stock market is a challenging task because of its chaotic and non-linear nature. Hence, the application of deep learning, especially in neural networks, has significantly risen in finance, due to its ability to discover such complex and dynamic patterns more efficiently than traditional methods. Neural networks have exhibited their capabilities in predicting stock market prices, enhancing accuracy as well as decision-making in financial markets. By utilizing the power of neural networks in financial analysis, it has improved precision in forecasting stock prices and investment returns (Nelson et al., 2017).

This thesis's main objective is to study the implementation of the deep learning approach, a subfield of machine learning, by applying neural networks to forecast the Swedish stock market performance, measured by returns. In particular, we utilize the Convolutional Neural Networks (CNN) model together with the bidirectional Long Short-Term Memory (BiLSTM) model. Along with the CNN-BiLSTM model, we leverage four macroeconomic indicators: including the Swedish *Inflation Rate, GDP Growth, Exchange Rate, and Unemployment Rate* to explore their relationship with the Swedish stock market, empowering the model to predict the return of the OMXS30 index. The evaluation of this model with the implemented macro indicators provides us valuable insights into the impact of the indicators on predicting the OMXS30 index return. Furthermore, it reveals which one of the indicators contributes the most significant effect on forecasting the Swedish index return.

By combining advanced deep learning models with macroeconomic analysis, this thesis develops the accuracy of the results in forecasting the Swedish stock market. This, in turn, demonstrates the power of applying deep learning neural network models in finance, modeling non-linear relationships, and improving the precision of market predictions. Hence, the findings from the thesis analysis can play a significant role for private investors and companies in decision-making within the stock market.

## 1.2 Purpose

This thesis aims to forecast the return on the Swedish stock market, namely the OMXS30 index return using an advanced deep-learning approach. In the study, an integration model is applied, consisting of the Convolutional Neural Networks (CNN) model along with the Bidirectional Long Short-Term Memory Networks (BiLSTM) model. By incorporating four macroeconomic indicators, including: The Swedish *Inflation Rate, GDP Growth, Exchange Rate, and Unemployment Rate,* the thesis aims to uncover the underlying relationships between the stock market and the domestic macroeconomic indicators, enabling the model to predict the OMXS30 return.

## 1.3 Research Question

To address our aim, the following research questions are introduced:

1. *To what extent can the inclusion of domestic macroeconomic variables explain and predict OMXS30 Index Return?*
2. *Which of the selected variables exhibits the highest predictive power in explaining OMXS30 Index Return within a CNN-BiLSTM-based modeling framework?*

## 1.4 Structure of the Thesis

The thesis is structured as follows: section 2 explores previous studies regarding the power of neural networks in finance and the impact of the macroeconomic indicators on the stock market. In section 3, theories behind the Convolutional Neural Networks (CNN) and the Bidirectional Long Short-Term Memory Neural Networks (BiLSTM) are presented in detail. Section 4 describes the data used in the thesis, and section 5 outlines the methods behind the CNN-BiLSTM model specification. Lastly, the results are presented in section 6, followed by a discussion and a conclusion covered in sections 7 and 8, respectively.

# 2. Literature Review

## 2.1 Neural Networks in Economics and Finance

Neural networks are defined as computational systems inspired by the human brain's way of processing. As is known, the human brain is a complex, non-linear, and parallel computing system. Hence, neural networks mimic the brain's approach to structure components, known as neurons, in order to perform certain tasks such as perception and pattern recognition. With their ability to solve such challenging tasks, like identifying a familiar face within an unfamiliar environment, neural networks demonstrate greater efficiency and speed than most powerful digital computers (Haykin, 2008).

Many earlier implementations of neural networks have been applied in several fields such as medicine, engineering, and psychology. However, they were proven to be equally effective in the field of economics and finance across numerous studies. Chen et al. (2015) conducted an analysis of studying the Chinese stock market. They utilize long short-term memory (LSTM) neural networks, to predict the returns of the stock market in China, based on different learning features, such as trade volume, closing price, and Shanghai securities composite index data. According to their results, the LSTM models significantly outperformed random predictions by improving the accuracy of the stock returns forecasts from 14.3% to 27.2 %.

Additionally, Nelson et al. (2017) analyze a similar study on forecasting the Brazilian stock market price movements by using long short-term memory (LSTM) neural networks. They examine whether the stock price would rise or fall based on technical indicators alongside historical price data. It is concluded that the results of the LSTM models demonstrated an accuracy of approximately 55% in predicting price movements, indicating a higher precision compared to traditional machine learning models.

## 2.2 Macroeconomic Variables in Forecasting

The analysis in this study is based on four macroeconomic indicators, including the Unemployment Rate, GDP growth, Inflation, and the Exchange Rate. These variables have been selected because of their proven influence on the stock market, as demonstrated in various research and studies, which will be further explored in the respective section for each variable.

### 2.2.1 Inflation Rate

Inflation is characterized by an increase in the cost of goods and services, indicating a lower economic purchasing power. It can be measured with different approaches, often by the consumer price index (CPI) that exhibits the price change for domestic consumption. Since 2017, Sweden has utilized the consumer price index with fixed interest (CPIF) to measure inflation, which reflects the same price variations, but excludes the impact of interest rate changes on household mortgages. (Statistics Sweden, 2024). In Sweden, the inflation target is 2 % per year measured in CPIF. Almost all the central banks all around the world have inflation targets to ensure stable and predictable price development, contributing to economic growth (Riksbanken, 2023).

Since the COVID-19 pandemic, Sweden's inflation target has been challenging to stabilize. According to Statistics Sweden (2024), the pandemic contributed to low inflation with an average annual rate of 0.5 %. Throughout 2021, the inflation rate began to reveal signs of recovery and ended up with an average annual rate of 2.2% which was higher than Sweden's inflation target of 2%. Unfortunately, the inflation rate continued to increase during 2022 and 2023, resulting in an average annual rate of 8.5 % in 2023. In December 2022, the inflation rate was at 12.3 %, the highest since October 1980, when it reached 15.5 % (Statistics Sweden, 2024).

Fama (1981) investigates an analysis, studying the negative relationship between inflation and stock market returns. The study is based on a regression analysis, exploring the interactions between stock returns, inflation, real economic variables, and monetary factors. He uncovered the indirect relationship, where high inflation harms economic growth, and since stock returns are positively correlated with real economic activity, they are negatively affected as a result. Likewise, Chen (2009) reveals similar results, where he conducts an analysis of whether macroeconomic indicators, including inflation rates, can predict the stock market by utilizing a predictive regression framework. The evidence exhibited a negative relationship between inflation and stock market returns, caused by a decrease in purchasing power. This implies that when inflation increases, prices rise, which in turn tends to reduce profits, affecting firms negatively. Lastly, Chen concludes that the inflation rate was, among all other indicators, the most reliable predictor of the stock market.

### 2.2.2 GDP Growth

Real gross domestic product (GDP) measures the total value of goods and services produced within a country during a given time period. Hence, GDP growth is the percentage change in a country's GDP from one period to the next, normally quarterly or annually, adjusted for inflation to provide accurate economic growth (OCED, 2024).

Over the last two decades, the timeframe examined in this study, GDP growth has declined sharply twice in Sweden, during the global financial crises and the COVID-19 pandemic. Following the Central Bank Group, GDP growth decreased approximately by – 4.3 % in 2009 and by –2.2 % in 2020. Subsequently, a strong rebound was observed the year after the economic crisis and the COVID-19 pandemic.

Nasseh and Strauss (2000), conduct a study to explore the long-run relationship between the stock market and domestic and international macroeconomic indicators, across six European countries. They investigate whether the stock prices are significantly related to industrial production (a proxy of GDP), and other macroeconomic variables, by utilizing the Johansen cointegration analysis. The researchers concluded that there is a strong, long-term integrating relationship between stock market prices and the selected variables, including industrial production. The cointegration results indicated that output growth influenced stock market prices through its impact on cash flow. Benzoni et al., (2007) reveal similar results regarding the relationship between the stock market and GDP growth, by specifically targeting the cointegration between the labor income and stock market prices, using the cointegration analysis as the primary framework. They concluded that labor income should be cointegrated with stock market prices, as labor income reflects the state of economic activity, therefore it is highly correlated to GDP. When GDP grows, labor income tends to increase, leading to higher stock market prices through enhanced corporate profitability as well as future cash flow.

### 2.2.3 Exchange Rate

The nominal exchange rate indicates how many Swedish SEK are required to purchase one unit of foreign currency. In contrast, the real exchange rate demonstrates the relationship between the Swedish and foreign price levels, which is influenced by the nominal exchange rate as well as the price development in Sweden relative to the rest of the world (Riksbanken, 2019). Hence, the real exchange rate is defined as a measure of the international competitiveness of a country. A

reduction in the value of the real exchange rate leads to higher purchasing power of domestic goods and services related to foreign ones. In such a case, it suggests that Swedish products become cheaper compared to foreign products, which in turn contributes to stronger competitiveness. In contrast, an increase in the value of the real exchange rate diminishes the purchasing power of domestic goods and services relative to foreign, resulting in weaker competitiveness (Riksbanken, 2006).

Numerous studies have demonstrated the relationship between exchange rates and the stock market, with one example being Dr. Tripathy's study. He investigates the causal relationship between selected macroeconomic variables, including the exchange rate and the Indian stock market by using Granger causality tests. Based on empirical evidence, she concludes that there is significant bidirectional causality between the exchange rate and the stock market. She argues that export competitiveness and import costs are influenced by the fluctuations in the exchange rate, where a weaker currency increases the import costs, reduces the probability of companies reliant on imports, and hence lowers their stock prices. This, in turn, indicates that the Indian stock market is influenced by the exchange rate, which therefore can be utilized to predict variations in the stock market prices.

### 2.2.4 Unemployment Rate

The unemployment rate is the percentage of unemployed individuals in the labor force aged between 15 – 75 years. During the past two decades, the unemployment rate increased significantly in Sweden during the financial crisis between 2008 – 2009 (Statistics Sweden, 2024). Following that period, the economy began to recover until the COVID-19 pandemic hit, leading to an increase in the unemployment rate. During the pre-pandemic, Sweden’s labor market rebounded strongly in 2022 with an average rate of 7.5 %, which was a decrease of 1.3% compared to 2021 (Statistics Sweden, 2023).

Previous studies reveal that the relationship between the unemployment rate and the stock market is complex, with no clear consensus among researchers. However, Gonzalo and Taamouti (2014) investigate the reaction of the stock market returns to anticipated and unanticipated unemployment rate announcements, by using nonparametric Ganger Causality tests and quantile regression-based tests. They find that an increase in the anticipated unemployment rate generally has a positive effect on the stock market, explained through monetary policy actions. They

observe that the high unemployment rate is followed by a reduction of the interest rate, which as a result increases the stock market prices. Also, they conclude that the unanticipated unemployment rate does not significantly influence the distribution of stock market returns.

On the other hand, Boyd, Hu, and Jagannathan (2005) utilize the linear regression analysis to observe that the unanticipated unemployment rate is dependent on the state of the economy. Following their findings, they discuss that on average the impact of the unemployment rate is considered as good news during expansion. They argue that rising unemployment throughout an expansion is seen as a positive signal for the future nominal interest rate to decline, improving stock market prices.

### 2.3 Hybrid Models in Financial Forecasting

The journey of forecasting the financial market has developed from simplified linear models such as ARIMA to further advanced deep learning approaches within artificial intelligence. As discussed by Nelson et al., (2017), neural network models based on deep learning have significantly improved the accuracy in predicting the stock market, by identifying complex and non-linear patterns in the stock data. However, despite their power in financial forecasting, these standalone models may struggle to fully address the complexity of the stock market. Hence, hybrid models have evolved by researchers, combining multiple neural network models to leverage higher accuracy in estimating the financial stock market, capturing layered patterns and interactions within the stock data.

Chen et al., 2023 perform an analysis studying 85 journal articles spanning from 2015 to 2023 from *the science citation index expanded database*, to explore how deep learning models have been utilized in financial forecasting. Hence, they conducted a comparison of standalone models such as (CNN) and (LSTM), versus hybrid models including (CNN-LSTM). The researchers examined the models by analyzing two primary tasks: classification and regression. The classification tasks involved predicting trends of whether stock prices would increase or decrease, while the regression tasks included forecasting the accurate values of the stock prices in the future. Results revealed that by combining CNN and LSTM, the hybrid model could effectively capture patterns as well as time series sequences within the financial stock data. Therefore, CNN-LSTM significantly outperformed the standalone models in trend prediction and stock price forecasting.

# 3. Theoretical Framework

## 3.1 Neural Networks and Time Series

As described by the researcher Haykin (2009), human brains are highly complex, nonlinear, and parallel information processing systems, having the capacity to solve advanced tasks. It is established that the brain consists of cells called neurons, which are responsible for performing complex processes such as pattern recognition, perception, and control, faster than powerful digital computers. The human brain has the ability to identify a familiar face within milliseconds, due to the brain's capacity to learn and adapt to new situations through experience. This flexibility enables the brain cells (neurons) to form new connections with each other, and process information efficiently. Hence, computer systems (artificial neural networks) have been designed to mimic the human brain, learning, adapting, and performing connections to enhance their performance on a particular task (Haykin, 2009).

A neural network (NN) is thought of as a machine built up to emulate how the human brain conducts a certain task. NNs are designed by a network of multiple artificial computing units, known as neurons. These neurons are represented as nodes in the network, which are connected and composed of various structured layers. A neuron is defined as a function of input values that generates an output from them, by multiplying each input with respective weight and therefore summing them up to form a linear combination. After including the constant and the bias, the linear combination is expressed as follows (Dreyfus, 2005):

$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b, \tag{1}$$

As observed, $x_1 \, to \, x_n$ are the inputs multiplied by the corresponding weights $w_1 \, to \, w_n$ and $b$ is the bias term. The neurons implement what is referred to as an activation function. This mathematical function transforms the values of the neuron's calculations, known as z, into a useful and desired output format $\hat{y}$. Hence, the neuron's output formula (Michelucci, 2022) is:

$$\hat{y} = f(z) = f(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b), \tag{2}$$

where $f$ is the activation function applied to $z$.

A layer is defined as a collection of nodes, also known as neurons who process the data and transfer it to the next layer, creating sequential patterns. The neural network (NN) contains multiple different layers, including an input layer, one or more hidden layers, and an output layer. Every layer is named to reflect its distinct role in the network (El-Amir and Hamdy, 2020).

The input layer is the first layer, with no neurons included, since the input layer receives the raw data and passes the values to the next layer without performing any calculations on it. Therefore, the input nodes on the first layer are called passive. In the middle, the hidden layers are represented where the neurons are calculated by a set of weighted inputs, combined into *z* values. These values are thereby transformed into neuron's outputs through an activation function. The activated output of each neuron in this hidden layer becomes an input for calculating the neurons in the next layer. The output layer is where the final output of the neural network is produced after taking the input (the activated output) from the neurons in the last hidden layer, calculating the weighted sum of the inputs, and applying the activated function (El-Amir and Hamdy, 2020; Dreyfus, 2005). By utilizing the following layers of neurons, the neural networks learn the relationships between the input and the output values, producing the final output. Once the output has been generated, it can therefore be applied to address classification and forecasting problems (Purkait, 2019). Figure 1 illustrates the framework of a simple neural network.

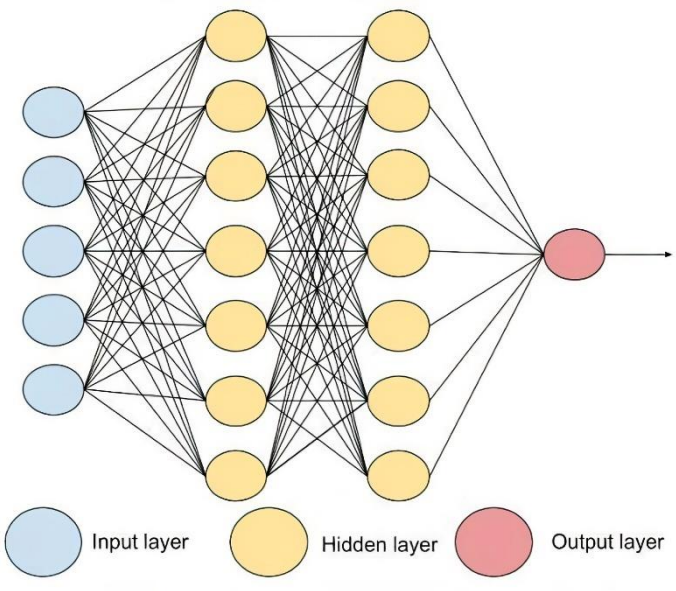


**Figure 1.** A design of a Neural Network with two hidden layers.

**Time series**

In accordance with Box et al., (2016), a time series is defined as a collection of numerical values arranged in a sequence over time. By analyzing historical data patterns, time series can be utilized to predict future values and trends. Common examples of time series include the weather in a specific area, the frequency of vehicle accidents to the value of a stock.

### 3.1.1 Feedforward Neural Networks

A feedforward neural network is one of the simplest NNs, where the data transfers in a forward direction from the input layer and through the hidden layers until the data arrives at the output layer. This implies that there are no cycles or loops in the network. Since the feedforward NNs are utilized for supervised learning, where the data is nonsequential and independent of time. As a result, they are commonly used for tasks such as classification and regression. (El-Amir and Hamdy, 2020).

### 3.1.2 Recurrent Neural Networks

Recurrent neural networks (RNN) are more evolving than forward neural networks. An RNN is characterized by its core concept of hidden state, giving it the ability to remember information from earlier timesteps. This implies that RNNs have memory capability, allowing previous information to persist within their network. Hence, RNNs utilize a feedback loop to connect the current input values with the calculated output values from preceding timesteps. Unlike the forward neural network, RNN follows a sequence order of the data and accounts for time dependence. Hence, RNNs are designed to be applied when performing more complex tasks with sequential data, for instance, speech recognition, translation, and time series forecasting (El-Amir and Hamdy, 2020; Witten et al., 2017).

### 3.1.3 The Vanishing and Exploding Gradient Problem

As is known, learning the simplest recurrent neural networks is essential to understanding their capacities in modeling sequential data under time dependence. This ability has underscored the power of RNN in performing complex tasks, establishing it as a wildly useful method today. Despite this, RNNs encounter difficulties in dealing with long data sequences with dependencies which consequently could result in failing to capture prior information. This is due to vanishing

and exploding gradient problems. Therefore, when the distance is short in the sequential data with dependency, RNNs predict outcomes accurately and efficiently, but as the distance becomes wider, RNNs suffer from gradient problems. During backpropagation through time (BPTT), these problems emerge when the recurrent neural networks propagate errors backward to adjust their components (weights and biases). As the gradient values go backward through multiple layers and approach the initial layer, they can either exponentially shrink or grow. The vanishing gradient problem occurs when the gradient values significantly decrease until they diminish close to zero, preventing the model from learning valuable information from earlier timesteps of the sequence. Conversely, the exploding gradient problem arises when the gradient values become extremely large, causing an unstable and ineffective model with undefined outputs (El-Amir and Hamdy, 2020).

### 3.2 Long Short-Term Memory Networks (LSTM)

Long Short-Term Memory (LSTM) is a more advanced Recurrent Neural Network (RNN) invented by Sepp Hochreiter and Jürgen Schmidhuber in 1997. This new model was not disturbed by the vanishing gradient problem while being designed to remember information over long periods, enabling it to capture long-term dependencies in sequence data. The model has been refined and widely used by many in their following work (El-Amir & Hamdy, 2020).

#### 3.2.1 Activation Functions in LSTM

Activation functions in deep learning are mathematical equations that determine the output of each neuron in a neural network. Activation functions play a major role in determining whether a neuron fires or not, which in turn depends on the relevance of the input to the model's prediction. Additionally, activation functions help to both normalize and scale the output from each neuron within specific ranges, such as 0 to 1 or -1 to 1 (El-Amir & Hamdy, 2020).

In LSTM neural networks, two activation functions are used: the sigmoid function (σ), also known as the logistic function, and the hyperbolic tangent (tanh) function. The sigmoid function normalizes the output from each neuron and maps the predicted probabilities as outputs in an interval between 0 and 1 (Zhao et al., 2024). The sigmoid function is used in the different gates of the LSTM cell, which are the input, output, and forget gates (Witten et al., 2017). A predicted probability of 0 from the sigmoid function indicates low importance of the input value to predict

the output, and the neuron will therefore not fire. On the other hand, a predicted value of 1 indicates high relevance of the input value in predicting the output, causing the neuron to fire, and allowing the information to pass on in the model (Zhao et al., 2024; El-Amir & Hamdy, 2020). The sigmoid activation function converts the values in each neuron via the following formula:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{3}$$

where $\sigma$ represents the sigmoid function, $x$ is the input value, and $e$ denotes Euler's number (the base of the natural logarithm).

The tanh function also known as the hyperbolic tangent activation function (HTAF) is similar to the sigmoid activation function but where the output is scaled to a range between -1 to 1 instead of just between 0 and 1. The advantage of this function is that the negative inputs will be negatively scaled and the inputs at zero will be scaled near zero in the tanh graph. (Zhao et al., 2024; El-Amir & Hamdy, 2020). The tanh hyperbolic activation function converts the values in each neuron via the following formula:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \tag{4}$$

where $x$ is the input value, and $e$ denotes Euler's number (the base of the natural logarithm).
Figure 2. below illustrates both the sigmoid and tanh activation functions.

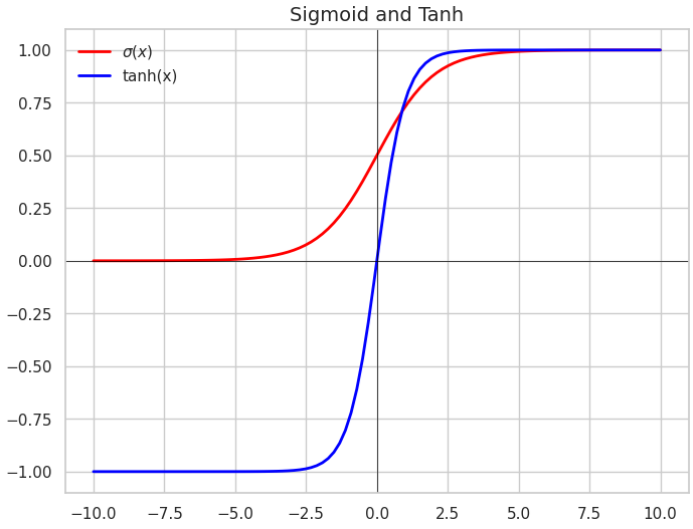


**Figure 2.** Visualization of Sigmoid and Tanh Functions

### 3.2.2 LSTM Cell Structure

The architecture of Long short-term memory (LSTM) is designed to effectively capture long-term dependencies while overcoming the vanishing gradient problem. LSTM models have a more advanced cell structure that includes memory cells and gating mechanisms that constantly regulate information flows to make the networks retain important information and completely forget irrelevant information (Praveenkumar et al., 2024). This cell structure is found in each neuron in the neural network, so each LSTM neuron consists of a memory cell that is responsible for passing data streams through the model (Moghar & Hamiche, 2020).

The structure of an LSTM cell is shown in Figure 3. The cell has three inputs, the first input is $x_t$, where $x$ is the data sample and $t$ is the time index in the specific cell. The second input is $h_{t-1}$ which is also known as the "hidden state" and is calculated in the previous cell and finally, the third input $C_{t-1}$ is what has been preserved and saved from the previous cell called the "cell state" (Peral-García et al., 2024). Additionally, as noted by Peral-García et al. (2024) and Praveenkumar et al. (2024), the cell has three important components (cell gates) which are:

1. **Forget gate ($f_t$):** handles the selection of information from the previous cell, i.e. decides whether to retain or forget information, determined by the sigmoid activation function.

2. **Input gate ($i_t$)**: Regulates new information presented to the current cell via a combination of both sigmoid and tanh activation functions.

3. **Output gate ($o_t$)**: Controls the output data from the current cell state to the hidden state in the cell.

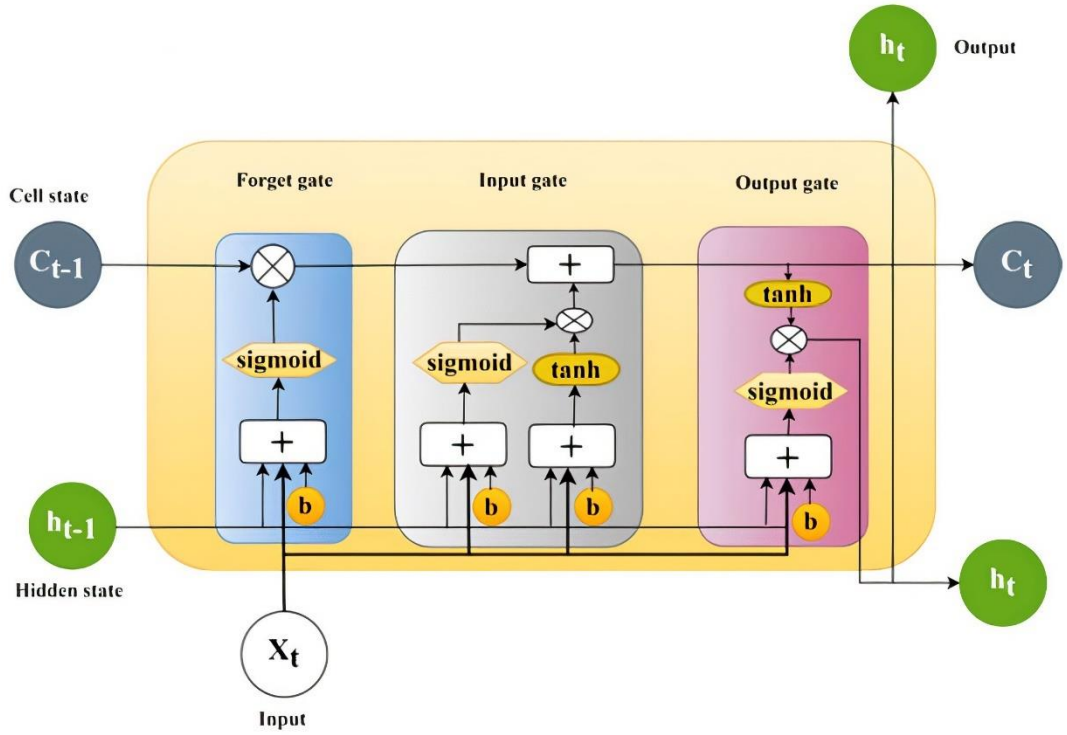


**Figure 3.** Structure of an LSTM cell.

The mathematical functions used for calculations in an LSTM cell are shown below:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{5}$$

$$i_t = \sigma(W_i x_i + U_i h_{t-1} + b_i) \tag{6}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{7}$$

$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{8}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{9}$$

$$h_t = o_t \odot tan\, h(C_t) \tag{10}$$

where *W*, *U*, and *b* (bias term) are the weight matrices that govern the gating mechanisms, and $\odot$ shows element-wise multiplication. The equations (5), (6) and (7) represent the Forget gate, Input gate, and Output gate as described on the previous page. Equation (8) generates new information that will be put into the current cell. Equation (9) updates the state of the cell by

combining the previous equation and the equation for the Input gate to obtain a new memory. Finally, equation (10) carries the output data that is used in the current cell and further in the next step (El-Amir & Hamdy, 2020; Praveenkumar et al., 2024).

### 3.2.3 Bidirectional LSTM

Bidirectional Long Short-Term Memory (BiLSTM) is a more advanced and extended version of the original LSTM. BiLSTM is designed to learn from past and future contexts in sequential data to improve the output result. BiLSTM handles the data in both forward direction (past to future) and backward direction (future to past) instead of just handling the data in a single direction like a standard LSTM. BiLSTM is based on two layers with which it processes the data sequences in both directions, allowing the model to capture a more general understanding of the sequence. This makes the model extremely effective in situations where data sequences contain useful contextual information from both the past and the future, which must be extracted to generate appropriate predictive output (Praveenkumar et al., 2024). A visualization of the bidirectional LSTM architecture is presented in Figure 4.

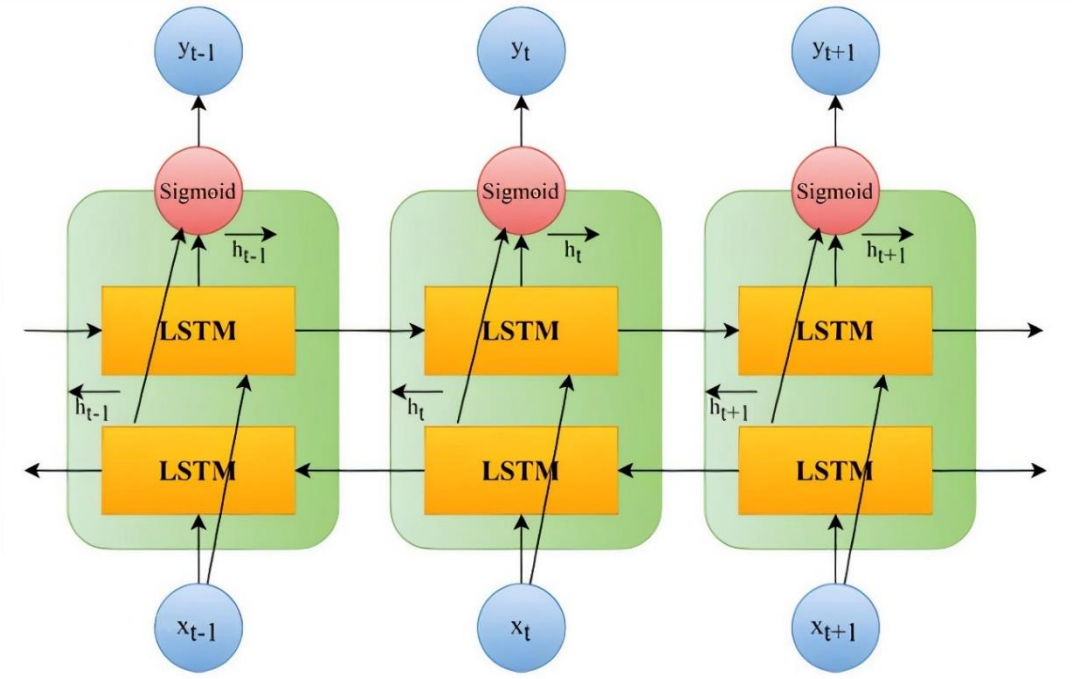


**Figure 4.** Bidirectional long short-term memory (Bi-LSTM) model architecture.

Forward direction LSTM handles the data from left to right (past to future) and backward direction LSTM handles the data from right to left (future to past), as shown in Figure 4. The final step in the bidirectional LSTM network is the concatenation that connects both the hidden states from the forward direction and backward direction LSTM. This allows the capture of long-term patterns from both directions, which in turn can be used in a variety of areas such as time series forecasting, speech recognition and more. Equation (11) shows the calculation that the forward direction LSTM performs at each time step. Equation (12) shows the calculation that the backward direction LSTM performs at each time step and equation (13) shows the connection between both LSTMs and their information from the past and future (Praveenkumar et al., 2024).

$$\overrightarrow{h_t} = LSTM\left(x_t, \overrightarrow{h_{t-1}}, \overrightarrow{C_{t-1}}\right) \tag{11}$$

$$\overleftarrow{h_t} = LSTM\left(x_t, \overleftarrow{h_{t+1}}, \overleftarrow{C_{t+1}}\right) \tag{12}$$

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right] \tag{13}$$

where $x_t$ is the input at time $t$, $\overrightarrow{h_t}$, $\overleftarrow{h_t}$, are the hidden states for the forward and backward LSTMs, and $\overrightarrow{C_{t-1}}$ with $\overleftarrow{C_{t+1}}$ are the cell states for each direction (Praveenkumar et al., 2024).

### 3.3 Convolutional Neural Networks (CNN)

The convolutional neural network (CNN) was created and proposed by Lecun et al. in 1998 (Wang et al., 2021). CNNs are feed-forward neural networks designed to process images, other visual tasks, and language processing. A typical CNN architecture, shown in Figure 5, includes convolutional layers, pooling layers, activation layers, and fully connected layers, where the convolutional layer captures the input data's local patterns and pooling operations reduce dimensionality to preserve the essence (Telmem et al., 2024). The structure of the CNN is inspired by the cooperation between the neurons and the organization of the visual cortex in the human brain. The visual cortex operates by activating individual neurons when something is displayed in the visual field, called the receptive field, these fields overlap and together cover the entire visual field. This is the base inspiration for the technological developments behind advanced models for image recognition such as CNNs (El-Amir & Hamdy, 2020).

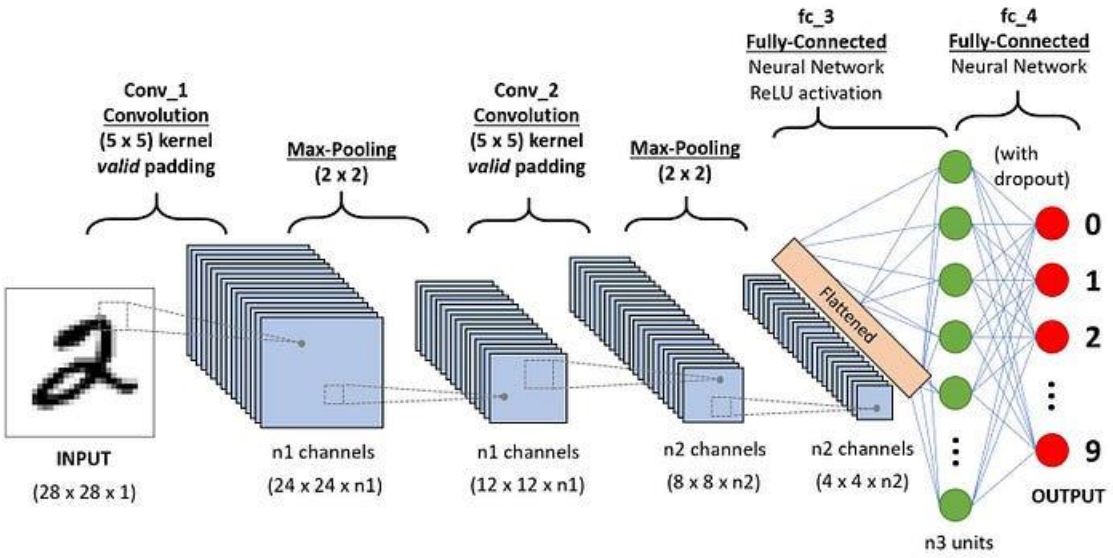


**Figure 5.** Typical architecture of a Convolutional Neural Network.

### 3.3.1 Activation Functions in CNN

The activation function most used in CNN is ReLU which converts the values to positive numbers (Alzubaidi et al., 2021). ReLU (Rectified Linear Unit), is an activation function that is particularly useful for solving the vanishing gradient problem during training. The definition of ReLU is shown in the mathematical function presented in Equation 14 (Mortezapour Shiri et al., 2023).

$$f(x) = max\,(0, x) = \begin{cases} x_i, & if\ x_i \geq 0 \\ 0, & if\ x_i < 0 \end{cases} \tag{14}$$

where *x* represents the input to the neuron in the model (Mortezapour Shiri et al., 2023). Figure 6. below illustrates ReLU activation functions.

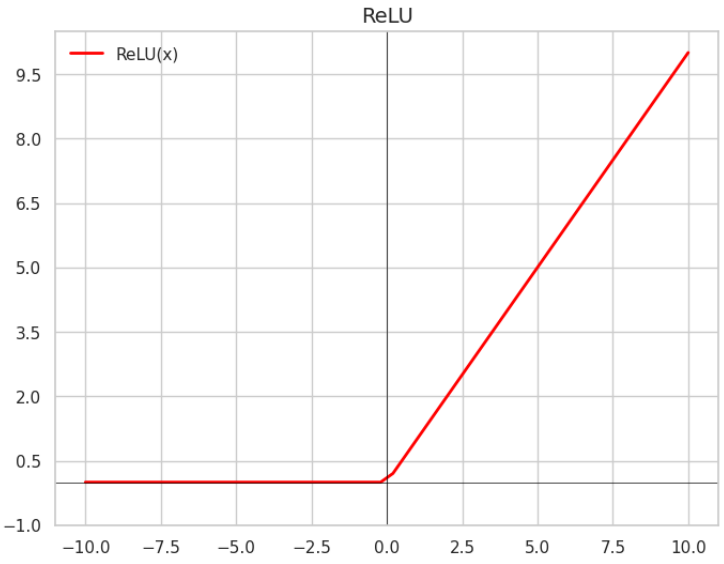


**Figure 6.** Visualization of ReLU Function

### 3.4 Integration of CNN-BiLSTM Architectures for Financial Time Series

Hybrid models combine deep learning and machine learning models in advanced and organized structures to improve predictive performance by leveraging different architectures' unique strengths and characteristics. In this case, the hybrid model with CNN and LSTM layers would potentially increase predictive power, as CNN can extract local/ short-term patterns (short-term fluctuations), while LSTM would be better at capturing long-term dependencies, such as cycles and market trends in the data (Chen et al., 2023).

The CNN-LSTM model is the most widely used hybrid model among researchers in recent times and has now become a state-of-the-art (SOTA) method, i.e. highest level of development, a method/technology in a field within a certain time. CNN is most often followed by LSTM in many cases but at the same time, other versions of this hybrid model are used in different contexts to maximize predictive power. The field of FinTech (financial technology) has made progress in recent years due to the combination of big data and artificial intelligence. Financial time series prediction as part of FinTech has also attracted a great deal of interest in the last decade in both academia and industry where various models and techniques for prediction have been proposed. At the same time, predicting financial time series from historical information is a challenge in itself because of the traits and complexity that characterize financial time series (Chen et al., 2023).

Characteristics that increase the complexity of financial time series include nonlinearity and non-stationarity, which refers to the fact that the data cannot be represented by a straight line and that its statistical properties change over time which may indicate that simple linear methods would not provide a significant predictive result. A characteristic known for financial time series that also increases complexity is short-term and long-term dependencies. These refer to short/fast fluctuations in the market and larger trends, with patterns that last for months or years. By not considering temporal dependencies, it has been proven from previous studies that it results in poorer predictive ability, which shows the importance of temporal dependencies in the predictive ability of the model (Chen et al., 2023). This justifies the choice of models used in this thesis, which is precisely CNN, which as previously mentioned captures short-term fluctuations in the data, and LSTM, which specializes in capturing long-term dependencies such as market trends and patterns.

Another important and well-known property of financial time series is volatility, which is the time series degree of variation around the mean. The behavior and phenomenon of volatility that is noticeable in financial time series is called volatility clustering. It refers to the fact that periods characterized by high volatility tend to be followed by periods of high volatility, as well as periods characterized by low volatility, and a more stable trend tends to be followed by similar periods. Finally, a key characteristic of the financial time series and their evolution is the influence that behavioral and psychological factors have on the dynamics of evolution, while geopolitical fluctuations, corporate reporting, government decisions, and similar events also affect investors' investment decisions, which in turn can cause changes and irrational fluctuations in financial time series (Chen et al., 2023). These characteristics and the complexity of financial time series motivate the decision to use a complex hybrid deep learning model to capture market patterns and enable reliable predictions based on historical data. This will further be described in the Data and Methodology section where the data used in the thesis is analyzed and the model choice is described in detail.

# 4. Data

This section will present and explain several parts, such as the data collection process, variable formatting, data preparation, descriptive statistics, and data visualization. At the same time, we will also explain and justify the Feature Engineering method and its advantages in the model-building process.

## 4.1 Data Collection and Structuring

The data utilized in this thesis consists of a total of 5 variables: Change in Inflation Rate, Seasonally Adjusted GDP Growth Rate, Change in SEK/USD Spot Exchange Rate, Seasonally Adjusted Unemployment Rate, and OMXS30 Index Return. The data for most variables are expressed in percentages and collected monthly from January 2000 to December 2023, providing 23 years of monthly observations for each variable. However, raw data for Inflation (CPI) and SEK/USD Spot Exchange Rate were instead downloaded as units and later reformatted to percentage changes to adapt to the other variables in the data set, which will be explained further below.

The sources of raw data are as follows: Inflation (CPI) and Seasonally Adjusted GDP Growth Rate have been obtained from Statistics Sweden, known as 'Statistiska centralbyrån' in Swedish, (SCB, 2024). The data for the SEK/USD Spot Exchange Rate was retrieved from the Federal Reserve Bank of St. Louis (Board of Governors of the Federal Reserve System, 2024). Data for Seasonally Adjusted Unemployment Rate was sourced from the Federal Reserve Bank of St. Louis, Organization for Economic Co-operation and Development (OECD, 2024). Finally, the raw data for the OMXS30 Index Return Index was collected from Investing.com (2024).

Seasonally adjusted data for both GDP growth and Unemployment rate ensures consistency among variables such as Inflation and Exchange Rate are not affected by seasonal patterns to the same extent. Additionally, this approach allows the analysis to focus on the impact of the underlying economic trends in OMXS30 without interference from seasonal fluctuations that would mislead and confuse the hybrid models employed. The variables above are selected based on previous research that has shown their significance in predicting the stock market. A detailed discussion of the variable selection process is provided in the Literature Review section.

The variables that have been reformatted from the raw data are Inflation (CPI) and SEK/USD Spot Exchange Rate. The reformatting includes calculating the change in the variables instead of using their usual historical values. This calculation has been made for the variables to fit better and to make a greater contribution to the hybrid deep learning model to be able to predict the return in the OMXS30 Index.

The percentage change in Inflation is calculated as:

$$Change\ in\ Inflation\ Rate\ (\%) = \left(\frac{CPI_t - CPI_{t-1}}{CPI_{t-1}}\right) * 100 \qquad (15)$$

where *t* indicates the current time period and *t-1* indicates the previous time period. $CPI_t$ denotes the Consumer Price Index during the current time period, $CPI_{t-1}$ denotes the Consumer Price Index during the previous time period and the multiplication by 100 makes possible the values in percentage instead of decimal form.

The percentage change in SEK/USD Spot Exchange Rate is calculated as:

$$Change\ in\ SEK/USD\ Spot\ Exchange\ Rate\ (\%) = \left(\frac{E_t - E_{t-1}}{E_{t-1}}\right) * 100 \qquad (16)$$

where $E_t$ denotes the Nominal Exchange Rate for Swedish Krona and US Dollar during the current time period, $E_{t-1}$ denotes the Nominal Exchange Rate for Swedish Krona and US Dollar during the previous time period and the multiplication by 100 makes possible the values in percentage instead of decimal form.

The data for all variables have been downloaded and reformatted in an Excel file (.xlsx). The Excel file has then been uploaded to a cloud-based programming platform called Google Collaboratory, which enables programming in Python directly in the web browser. Using the library TensorFlow, the data was processed and analyzed further to construct and train the deep learning models.

### 4.1.1 Feature Engineering

In recent decades, technological developments have changed both the amount of data and the speed with which this data must be processed and analyzed. The data used in machine learning and neural network models tend to be complex in nature with a variety of variables and extracted from various sources, both structured and unstructured. Therefore, it is necessary in some cases to transform the variables to better fit in and contribute more to the models in machine and deep learning. This important step is called Feature Engineering (Verdonck et al., 2021). The goal of Feature Engineering is to transform/modify the variables already present in the dataset or create new variables based on the existing variables to gain new insights about the dataset used to train the models (Deverill et al., 2024).

The feature engineering methods implemented in this thesis create new variables based on existing variables. These methods are lagged functions, moving averages, and moving standard deviations.

- *Lagged variables*

$$X_{t-lag} = X_t - lag \tag{17}$$

where $X_t$ denotes the value of the variable at time t and lag denotes the number of time units backward, in this thesis (1, 2, 3, 6, 12).

- *Moving average*

$$MA_t = \frac{1}{w}\sum_{i=0}^{w-1} X_{t-i} \tag{18}$$

where $MA_t$ denotes the moving average at time *t*, $X_{t-i}$ denotes the value of the variable *X* at time *t-I*, and w indicates the window size, in this thesis (3, 6, 12).

- *Moving Standard Deviation*

$$STD_t = \sqrt{\frac{1}{w}\sum_{i=0}^{w-1}(x_{t-1} - MA_t)^2} \tag{19}$$

where $STD_t$ denotes the moving standard deviation at time *t*, $X_{t-i}$ indicates the value of the variable *X* at time *t-1*, $MA_t$ denotes the moving average at time *t-i* and *w* denotes the window size, in this thesis (3, 6, 12).

The mean (average) and standard deviation are valuable measures that characterize the data. However, these measures are calculated with the same weight for the entire data period, making it difficult to capture the data's short- and long-term trends. The solution to this problem is rolling statistics such as rolling mean (moving average) and rolling standard deviation (moving standard deviation) that extract only a portion of the entire time series for a given window size. Thus, part of the data is extracted consistently, which makes the rolling statistics calculated as a time series. The larger the window size, the greater the smoothing of the mean and standard deviation will be, which shows long-term trends in the data, while smaller window sizes allow the calculation of the mean and standard deviation to capture short-term fluctuations in the data, making the trend sharper (Oh et al., 2024).

The variables that have been transformed with the methods mentioned above are all the variables that have been used to predict the OMXS30 Index Return, which are: Change in Inflation Rate, Change in SEK/USD Spot Exchange Rate, Seasonally Adjusted GDP Growth Rate, and Seasonally Adjusted Unemployment Rate. The window sizes chosen for both the rolling average and the rolling standard deviation are 3, 6, and 12, which means that the calculation of the rolling statistics consider quarterly, semi-annual, and annual time horizons to capture both short- and long-term patterns in the data for the 4 basic variables. Simultaneously lagged variables have been used with previous observations of the variables from 1, 2, 3, 6, and 12 months back in time, these are used for modeling historical information and how this information affects future values in the variables.

It is also important to point out that in this thesis, to ensure isolation of the effect of macroeconomic variables on OMXS30, technical indicators such as volatility and rolling statistics with lagged transformations for OMXS30 have not been used in the deep learning models. This choice is fundamental in the study as the meaning of the study is to find the main effects of the macroeconomic variables on the OMXS30 Index Return without exaggerating or introducing self-reinforcing effects between the OMXS30 and its previous values. By restricting the input data to the four fundamental macroeconomic variables and their derived transformations, the analysis keeps a clearer and more focused isolation of the macroeconomic variables' impact on stock market performance without the influence of internal technical signals.

### 4.2 Data Preparation

The data is processed before being introduced to the models. Following the implementation of feature engineering and the construction of sequence lengths, a new structure for the data has been created, where certain rows include missing values for some variables. Below, the handling of missing values and how the data has been scaled before being introduced to the various model configurations will be explained.

#### 4.2.1 Missing Values and Data Scaling

Feature engineering variables are created by utilizing past data, and as a result, the initial rows of the dataset are excluded from the analysis. Additionally, the sequence lengths in all models are set to 30, meaning that the models use the most recent 30-time steps to predict the 31st-time step. This process also excludes rows from the analysis. Missing data in the dataset is handled using the "df.dropna()" method, which removes entire rows that contain missing values. Another method implemented to manage outliers and rapid fluctuations in the data is RobustScaler, which scales the data down before the model trains it and later inversely transforms the data back to its original scale for evaluation purposes.

### 4.3 Exploratory Data Analysis and Assumptions

**Table 1.** Descriptive Statistics

| Variable | Obs. | Mean (%) | Std.Dev. (%) | Min (%) | Max (%) |
|---|---|---|---|---|---|
| Change in Inflation Rate | 288 | 0.17 | 0.48 | -1.40 | 2.10 |
| SA GDP Growth Rate | 288 | 0.18 | 1.31 | -5.50 | 5.10 |
| Change in SEK/USD Rate | 288 | 0.10 | 2.56 | -6.87 | 11.40 |
| SA Unemployment Rate | 288 | **7.36** | 1.00 | 4.90 | 9.60 |
| OMXS30 Index Return | 288 | 0.39 | **5.36** | **-16.86** | **17.44** |

Note: “Obs.” stands for Observations and “SA" stands for Seasonally Adjusted

The descriptive statistics in Table number 1 show the characteristics of all the variables that are included in the analysis. Change in the Inflation Rate turns out to have a relatively stable fluctuation around a low average where its mean value is 0.17% and standard deviation of 0.48%. The Seasonally Adjusted GDP Growth Rate is also close to the Change in Inflation Rate, with a mean of 0.18%, but it has a slightly higher standard deviation of 1.31%. This indicates relatively weak and variable seasonally adjusted economic growth. Similar to the Change in Inflation Rate, the Change in SEK/USD Rate also has a relatively low mean of 0.10%. However, it exhibits a higher standard deviation of 2.56%, indicating relatively high volatility, which is likely due to financial shocks and external market dynamics. The Seasonally Adjusted Unemployment Rate has the highest mean among the variables, with a value of 7.36%, and a relatively low standard deviation of 1.00%. This reflects a relatively stable but elevated unemployment rate over time. The last variable, OMXS30, which is also the variable we are trying to predict, has a mean of 0.39% and the highest standard deviation among all variables at 5.36%, indicating the greatest volatility among the variables. This high volatility likely reflects financial crises, market recoveries, and changes in market dynamics.

The statistical summary above clearly illustrates the diversification among the variables and their differing characteristics, where some variables are more stable while others are more volatile over time. These traits are critical and necessary for the model's construction, as they underscore the need for advanced and robust methods to capture the dynamic relationship between the macroeconomic variables and OMXS30 returns. The choice and construction of the model will be further explained in the Methodology section.

### 4.3.1 Analysis of Stationarity and Heteroskedasticity

Stationarity is an assumption that needs to be considered when statistical conclusions are drawn about the structure of a stochastic process based on the observation of that process. Stationarity has as its basic idea that the probability rules that govern the behavior of the process do not change over time (Cryer and Chan, 2008). In other words, in a stationary process, the unconditional joint probability is unchanged, where the mean and variance are constant over time, and the covariance depends only on the time distance Wooldridge (2016). To evaluate the variable's stationarity in the dataset, the augmented Dickey-Fuller test is used, where the null hypothesis tells that the time series has a unit root, and if that root is rejected, the test suggests that the variable is stationary. Table A.2, in Appendix A, the Dickey-Fuller test results are

shown, where it can be concluded that all the variables in the data set are stationary, as the hypothesis of having a unit root is rejected for all variables at a 5% significance level.

Homoscedasticity, on the other hand, is an assumption that means that the variance of the residuals remains constant across the entire data set for the variable. Assumptions of homoskedasticity is broken when: var (y | x) ≠ constant, in this case, we expect signs of what is called Heteroskedasticity, which means that the variances of the residuals do not have a constant spread. One way to detect heteroskedasticity is to plot the residuals in graphs for each variable and see if its spread is constant (Brooks, 2008). These graphs have been created and are included in the STL decomposition for each variable described in the paragraph below.

The STL decomposition for each variable has been created and presented in this thesis to visualize both the seasonal patterns underlying the assumption of stationarity and the residual spread underlying the assumption of homoscedasticity for each variable. According to Hyndman and Athanasopoulos (2018), the STL decomposition method is a robust approach for decomposing time series. STL stands for "Seasonal and Trend decomposition using Loess," where Loess is a method for estimating non-linear relationships. STL decomposition offers several advantages, including the ability to handle any type of seasonality, allowing the seasonal component to change over time, and being robust to outliers.

The specific STL decompositions for all variables can be found in Figures A.1, A.2, A.3, A.4, and A.5 in Appendix A. In these decompositions, we can observe that, in general, there are no obvious signs of heteroscedasticity, as the residuals appear to be relatively distributed around zero over the entire period for all the variables. Also, the seasonal patterns align with the results of the Dickey-Fuller test, as there are no significant deviations from the overall patterns that would violate the stationarity of any of the variables.

### 4.3.2 Multicollinearity Analysis

Multicollinearity is a problem that occurs when two or more variables provide similar information. This occurs when two or more variables/indicators have a correlation coefficient close to +1 or -1, meaning they are either perfectly positively correlated or almost perfectly negatively correlated. At the same time, it is important to understand that the relationship between two or more variables does not necessarily mean causation. A variable can be correlated with and used to predict another variable, but this does not mean that the variable causes the one being predicted (Hyndman & Athanasopoulos, 2018).

In this thesis, a correlation matrix heat map has been utilized to visualize the correlation between the macroeconomic indicators themselves, as well as the correlation between the indicators and OMXS30, this is shown in Figure A.6 in Appendix B. The correlation matrix also provides an insight into potential signs of multicollinearity, that would possibly disturb and mislead the neural network as the meaning of the thesis is to isolate the effect of the indicators on OMXS30. In the correlation matrix above, it is evident that the correlation coefficients between the macroeconomic indicators are relatively small, where the correlation coefficients vary between -0.07 to 0.06. Considering the indicators' correlation with OMXS3O, the Seasonally Adjusted Unemployment Rate exhibits the highest correlation of 0.25. Seasonally Adjusted GDP Growth Rate shows a moderately positive correlation coefficient of 0.13 and Change in Inflation Rate with Change in SEK/USD Spot Exchange Rate shows moderate negative correlations of -0.12 and -0.13, respectively.

In this thesis, the Seasonally Adjusted Unemployment Rate exhibits the strongest positive correlation to OMXS30, but that does not necessarily mean that the variable will have the greatest predictive power. What also needs to be considered is that the relationship between the indicators and OMXS30 is not only linear as in the correlation analysis but also non-linear, which can alter the predictive power in explaining OMXS30. The use of feature engineering can also possibly affect the predictive power since it creates new variables, which in turn also affects the predictive power of the variables. Lastly, the scaling of the variables can also affect which of the variables exhibits the greatest predictive power, a topic already presented further up in the subsection Missing Values and Data Scaling.

### 4.3.3 Autocorrelation Analysis

Autocorrelation measures the linear combination between the lagged values in a time series, unlike correlation, which only measures the extent of a linear relationship between two variables (Hyndman & Athanasopoulos, 2018). In other words, autocorrelation occurs when the residuals are correlated with each other over different lagged time periods, thereby violating the assumption of no autocorrelation in the variables (Brooks, 2008).

The assumption of autocorrelation in the residuals for each variable was tested using ACF plots (Autocorrelation Function). The ACF plots for all variables in the data set are presented in Figures A.7, A.8, A.9, A.10, A.11, and A.12 in Appendix B. The ACF plot for the seasonally adjusted unemployment rate and its corresponding PACF (Partial Autocorrelation Function) is shown in Figures A.10 and A.11. This is the only variable with a PCAF as the seasonally adjusted unemployment rate is the only variable in the data set that shows a significant autocorrelation problem. The PACF plot is made specifically for the variable with autocorrelation problems to better understand which lagged values of the seasonally adjusted unemployment rate should be included in the analysis.

In the graph shown in Figure A.10, it is evident that lags from 0 to 10 exhibit high residual correlations, indicating strong temporal dependencies over shorter periods, which gradually decrease and approach zero. This suggests that the variable exhibits long-term dependencies and relatively moderate stationarity as we noticed in the Dickey-Fuller test where the Seasonally Adjusted Unemployment Rate had the highest p-value of all variables of 0.0109. However, the p-value is still rejected at the 5% significance level, indicating a low degree of stationarity which does not pose an issue for the analysis. Simultaneously, the graph shown in Figure A.11, shows significant partial correlations for lags 1 to 3, indicating that the analysis should include lagged values of the Seasonally Adjusted Unemployment Rate for periods 1, 2, and 3. This has already been accounted for in the analysis, as lagged variables for all macroeconomic indicators have been created for periods 1, 2, 3, 6, and 12, which automatically addresses the autocorrelation issue.

# 5. Methodology

This section will describe and explain the strategy for model development and the applied methodology used in the thesis. Furthermore, the architecture and hyperparameters of neural networks are discussed and explained. Lastly, the choice behind the selection of evaluation metrics and their advantages in assessing model performance are presented.

## 5.1 Model Development Strategy and Dataset Splitting

The thesis adopts an incremental methodology by developing and comparing four Convolutional-Bidirectional-Long Short-Term Memory Neural Networks (CNN-BiLSTM). The meaning of the incremental method is to gradually introduce the variables to the CNN-BiLSTM model, which in turn systematically reveals the individual contributions of each macroeconomic variable to the OMXS30 index while assessing the final model's highest predictive performance. All four models share the same architecture and sequence lengths to maintain a fair comparison while isolating the individual contributions of each variable to model performance. The models and variable selection for each model are explained a little later in the section. The performance evaluation of each model and the comparison between them is made possible via various metrics that measure the degree of explanation in the training and test data, as well as the size of error produced by each model. These metrics will be further explained later in the methods section.

In machine and deep learning, where advanced models such as neural networks are used, the data is usually divided into training and test data to check how the model generalizes on new/unseen data. The most common distribution is 80% for training data and 20% for test data (Michelucci, 2022). In this thesis, splitting (80-20) has been used for all four models. The training dataset, which constitutes 80% of the total data, has been further split into 60% training data and 20% validation data. The validation dataset, which accounts for only 20% of the training data, is used only for internal purposes, such as monitoring underfitting and overfitting of the models. The graphs illustrating the model's performance will include only the training and test datasets. As the validation dataset is used solely for internal purposes, it will not be displayed in these graphs, which will be presented in the Results section. According to Witten et al. (2017), the validation dataset is used to fine-tune hyperparameters. Therefore, it has been internally used in the thesis but will not be displayed.

The original data set extends from January 2000 to December 2023. Feature engineering and the sequence lengths caused some values to be lost, which were handled by excluding the respective row. The new data set that remains extends from July 2003 to December 2023. The data has then split into a training set (with a validation subset) and a test set as follows:

**Training set**: This includes 80% of the total data used to train the four models, spanning from July 2003 to October 2019.

**Validation set**: This includes 20% of the training data for all four models, spanning from August 2016 to October 2019.

**Test set**: This includes 20% of the remaining data that the model has not previously seen, spanning from November 2019 to December 2023.

The order of the variables added to the models is based on previous studies and their perspectives regarding the variables' impact on the OMXS30 Index Return. As previously mentioned in the Literature Review, earlier studies have concluded that the Inflation Rate is the most reliable variable for predicting the stock market. GDP growth shows a stronger relationship with the stock market in the long run rather than the short run, and the exchange rate generally indicates a relatively strong ability to predict the stock market via changes in exports and imports. Finally, the unemployment rate exhibits a relatively low and insignificant impact on the stock market, partly because the unemployment rate has an indirect effect on stock market performance. Variable orders in each model have been chosen as follows:

**Model 1**: The first model includes only the *Change in Inflation Rate* variable along with its respective variables created through feature engineering to predict the OMXS30 Index Return. This CNN-BiLSTM neural network highlights the direct impact that Changes in Inflation Rate have on predicting the OMXS30 Index Return. This choice is motivated by its reliability in predicting the stock market performance in previous studies.

**Model 2**: The second CNN-BiLSTM neural network now includes *Change in Inflation Rate* and *Seasonally Adjusted GDP Growth Rate*. In this model, one more variable has now been incrementally added, which has been chosen as the second strongest variable based on previous studies to predict stock market performance. The improvement in model performance can now be attributed to the individual contribution of the seasonally adjusted GDP growth rate and its modified values.

**Model 3**: The third CNN-BiLSTM neural network includes Change in Inflation Rate, Seasonally Adjusted GDP Growth Rate, and Change in SEK/USD Spot Exchange Rate. In this model, the third variable has now been incrementally added through conclusions from previous studies and again the differences in the model's performance will be addressed to the Change in SEK/USD Spot Exchange Rate and its modified values.

**Model 4**: The fourth and last CNN-BiLSTM neural network includes all the macroeconomic variables in this thesis to predict the OMXS30 Index Return, i.e. Change in Inflation Rate, Seasonally Adjusted GDP Growth Rate, Change in SEK/USD Spot Exchange Rate, and Seasonally Adjusted Unemployment Rate. The final variable and its derived features have been added last due to its relatively weak and indirect effect on predicting stock market performance, as concluded in previous studies.

### 5.2 Choice of Model Architecture

The neural network used in this thesis is, as previously mentioned, a hybrid deep learning model that combines layers from two different models. This allows both short-term and long-term patterns in the data to be captured effectively. The hybrid neural network is built up of six different types of layers, including the Input layer, Convolutional layers, Bidirectional LSTM layers, Concatenate layer, Dense layer, and Output layer. The convolutional layers are derived from the Convolutional Neural Network (CNN), while the bidirectional LSTM layers originate from the Bidirectional Long Short-Term Memory Neural Network (BiLSTM). This general neural network architecture has been applied to all four models to ensure a fair comparison and isolate the main effects of the variables. However, certain parameters have been adjusted to optimize each model's performance for their respective variables. In the following subsections, the architecture of these models and the hyperparameter adjustments will be described in detail.

### 5.3 Hyperparameters

Machine learning models have both parameters and hyperparameters. Parameters are variables of the model itself that are determined during training, such as the coefficients of linear regression, while hyperparameters are determined before training and are responsible for the

model's learning period and performance in the test data (Arnold et al., 2024).

Hyperparameters in deep learning models used in financial time series predictions play a crucial role in the performance of the model. These hyperparameters in a deep learning model include batch size, number of layers from different models, neurons, and number of epochs. The adjustment of these hyperparameters allows for improvements in the performance of the deep learning model (Chen et al., 2023). In this thesis, the hyperparameters for all four models have been adjusted manually based on tips from previous research and intuition.

### 5.3.1 Batch size

The hyperparameter Batch size represents the number of training samples received and processed by the model before the model updates its parameters; weights and biases (Michelucci, 2022). There is no universal batch size that works perfectly for all models, practically it is about the architecture of the specific model and the characteristics of the data (Khodabakhsh et al., 2020). Out of the batch sizes of 1, 8, 16, 32, and 64, which are also the most common, batch size 32 tends to provide the most robust and stable performance on larger training datasets (Reimers & Gurevych, 2017). In this thesis, batch size 32 has therefore been chosen for models 1, 2, and 4, while for model 3, batch size 30 has been chosen for slightly better performance compared to the batch size used in the previous models.

### 5.3.2 Layers

A neural network is made up of several layers, which are the input layer, the hidden layer, and the output layer. The layers are made up of different neurons, which will be described in a subsection below. A neural network with only one hidden layer is called a Shallow neural network, while a neural network with two or more hidden layers is called a Deep neural network. Neural networks with only one hidden layer are said to perform at least as well as deep neural networks, but still, there is no exact answer to the number of layers that should be used in a neural network. However, it is recommended that deep neural networks, i.e. networks with multiple hidden layers, be used for more complex data (El-Amir & Hamdy, 2020). In this thesis, a deep learning model has been chosen to be used and the same architecture but different configurations of it have been applied with different numbers of input variables. The architecture of the neural network used in this thesis contains 5 hidden layers including two

Convolutional layers, two Bidirectional LSTM layers, and one Dense layer. The rest of the layers in the model are the Input layer which contains the input data, the Concatenate layer which is a single layer where no parameters are calculated but only merges the output data from the Convolutional and Bi-LSTM layers, and the Output layer which contains the output data for the variable OMXS30.

Reimers and Gurevych (2017) suggest that between one, two, or three BiLSTM layers, two BiLSTM layers are the most optimal and robust for use in a neural network. Meanwhile, Raiaa et al. (2024) highlight that the performance of a Convolutional Neural Network (CNN) depends on the number of Convolutional layers. Additionally, dataset size also plays a role where convolutional layers with lower steps achieve higher accuracy and performance in larger datasets, while in smaller datasets there is a risk of overfitting. These hidden layers have therefore been chosen to be used in the thesis: two Convolutional layers and two Bidirectional LSTM chosen for optimal and high performance. Finally, a Dense layer that sums all values from previous neurons with different weights and bias terms.

### 5.3.3 Neurons

The smallest unit of a neural network is an artificial neuron, also sometimes called a perceptron. The artificial neuron in the neural network receives input data, processes it through an activation function (such as sigmoid or tanh), and then passes forward the activated output. This means that the artificial neuron is a linear and binary classifier that calculates the output data *f(x)* via the function $f(x) = \langle w, x \rangle + b$ where *w* is a vector of weights, *x* is calculated to a positive or negative value, and b represents the bias term (El-Amir & Hamdy, 2020). In a neural network, the number of neurons is an important decision to achieve successful training. The number of neurons in the Input layer is determined in advance by the input variables that introduce the raw data to the model, and the neuron number for the Output layer is determined by the variable that the model wants to predict, while for the hidden layers, the number of neurons is not fixed and is adjusted to achieve the highest performance in the neural network. In a scenario where the number of neurons determined in the hidden layers is lower than what is needed in relation to the complexity of the problem, underfitting will occur, while if more neurons are included in the layers, the phenomenon of overfitting will occur (Adil et al., 2020).

According to Bohdan Macukow (2016), the number of neurons in the neural network should be set from the beginning based on theoretical knowledge and then adjusted experimentally based

on the size and complexity of the data. Meanwhile, according to Adil et al. (2020), between 60-100 neurons are recommended for small and medium-sized datasets. Layers with over 100 neurons can lead to longer training times, require more computational power, and can potentially lead to overfitting. Based on previous studies and the complexity of the data in this thesis, it has been chosen to have 64 filters in the first Convolutional layer, and 128 filters in the second layer for all four models. These layers are recommended to capture short-time dependencies in the data and the number of filters (neurons in the convolutional layer) was adjusted for the best performance of the models. The number of neurons for the first Bidirectional layer is 124 neurons for both (past to future) LSTM and (future to past) LSTM for model 1, 64 neurons each for models 2 and 4, and 61 neurons in each direction for model 3. In the second Bidirectional layer, the number of neurons is reduced to 62 neurons in each direction for model 1, 32 neurons in each direction for models 2 and 4, and 30 neurons for model 3. Finally, the last hidden layer, which is the Dense layer, includes 64 neurons in each model. The choice of neurons in each hidden layer is determined for the highest possible performance in each model while preventing overfitting and underfitting.

### 5.3.4 Loss Function

The loss function is an important metric that helps neural networks understand whether learning is in the right direction. This metric is used to measure the performance of the model and how much the model deviates from the predictive target (El-Amir & Hamdy, 2020). The definition of the loss function goes hand in hand with the data type used in the specific context. Two of the most common loss of function are MSE (Mean Square Error) and MAE (Mean Average Error), but in this thesis, a loss function that takes advantage of both is used and it is called Huber Loss. Huber Loss is a functional loss variant of the MAE that automatically becomes MSE when the residuals are small. At a larger difference between prediction and output, errors are linear, which makes Huber Loss less sensitive to outliers while when errors are smaller Huber Loss will follow MSE which makes convergence faster for the model (Ciampiconi et al., 2023). The mathematical function for Huber Loss is shown in Equation 20.

$$L_{Huberloss} = \begin{cases} \frac{1}{2}\left((y_i - f(x_i))^2 \right. & for\ |y_i - f(x_i)| \leq \delta, \\ \delta\left(|y_i - f(x_i)| - \frac{1}{2}\delta\right) & otherwise \end{cases} \tag{20}$$

where $y_i$ is the true value in the data, $f(x_i)$ is the model's prediction and $\delta$ is the threshold value

that determines when the loss function switches between MSE and MAE.

Huber Loss has therefore been chosen to be used in this thesis for all four models, to handle outliers more effectively, especially in financial data where outliers can occur and disrupt the training process. Huber Loss's effectiveness in handling extreme values and outliers can allow the model to focus more on the central trend in the data, which improves performance. In this thesis, TensorFlow's built-in Huber Loss function is used with a threshold value of $\delta = 1.0$ for all four models. Residuals that are less than or equal to 1.0 will not be considered outliers and will thus be treated with MSE while residuals that are greater than 1.0 or less than -1.0 will be considered outliers and thus be handled with MAE.

### 5.3.5 Optimizer

Optimizers are algorithms that minimize given functions. A neural network's training process minimizes the loss function. During training, the optimization algorithm adjusts the model's parameters (weights and bias) to find the minimum value in the loss function. There are a variety of optimizers, and the choice of optimizer depends on the type of problem to be solved. A good start would be Adam (Adaptive Moment estimation) which is considered a fast and robust method for optimizing the loss function (Michelucci, 2022). According to Kingma and Ba (2014), Adam is a stochastic optimization algorithm that combines the advantages of two optimizers, namely AdaGrad (Adaptive Gradient Algorithm) and RMSProp (Root Mean Square Propagation), which makes Adam capable of adapting the learning rate for each parameter and can effectively handle non-stationary data. Additionally, Adam is a suitable optimizer that works well with large data sets and is computationally efficient. Due to the efficiency and robustness of Adam, this optimization algorithm has been chosen to be used in all four models in this thesis.

### 5.3.5 Number of Epochs

An epoch is a hyperparameter that plays an important role in a model's training process (Afaq & Rao, 2020). An epoch is defined as a passage over the dataset and the number of epochs determines how many times the model iterates over the entire dataset during training (Hussein & Shareef, 2024). The number of epochs is not determined by a strict rule, however, the evaluation of training and validation loss for the neural network can be used as a guideline. It

is important to mention that the optimization process for the number of epochs consists of two major problems in machine learning and deep learning which are Underfitting and Overfitting (Afaq & Rao, 2020). Overfitting is a phenomenon that occurs when the model is so flexible that it learns patterns in the data that are due to errors, noise, or irrelevant information (Michelucci, 2022). This causes the model to perform poorly on the test data. In other words, in this case, the model has captured irrelevant and incorrect information in the training data, which reduces the accuracy of the model, which in turn leads to poor performance in the unseen data (Afaq & Rao, 2020). The opposite of overfitting is the phenomenon of underfitting which also occurs when training neural networks. Underfitting occurs when the model fails to capture the underlying patterns in the data and will thus perform poorly on both training and test data (Michelucci, 2022). Hyperparameter adjustment, in this case, the number of epochs, is appropriate to be determined based on the complexity of the data and the nature of the problem. This means that a low number of epochs for complex data reduces the opportunity for the model to learn and thus leads to underfitting. An excessively large number of epochs can cause the model to "overlearn" incorrect information which in turn leads to the model not capturing the underlying patterns and thus overfitting (Afaq & Rao, 2020).

The number of epochs for the four models is chosen precisely in this way to maximize the models' performance in predicting unseen data in OMXS30 while trying to avoid phenomena such as underfitting and overfitting. The number of epochs for all four models is chosen as follows: 200 epochs for model 1 and 300 epochs for models 2, 3, and 4 respectively. To be able to understand if the models exhibit underfitting or overfitting, the Huber Loss function for both the training and validation data is presented in Figures. 11, 12, 13 and 14 below. The graphs below show that the plots for both training and validation data stop earlier than the epochs mentioned here as in this thesis regularization techniques are used which will be explained in the next sub-section.

Figure 7 shows the training and validation plots based on the Huber Loss function for model 1 with only the Change in Inflation Rate variable. Both plots decrease drastically at the same time until they stabilize and converge at a lower value, around 40 epochs. The convergence between the plots shows a robust model that performs and generalizes well in unseen data. There is no sign of overfitting in this case, as the validation plot does not rise above the training plot at higher epoch values. Meanwhile, another point of view regarding the computational time of the model is the convergence at 40 epochs, which means that an earlier stop could be used for model 1 where training would be stopped before 70 epochs to possibly keep similar results.

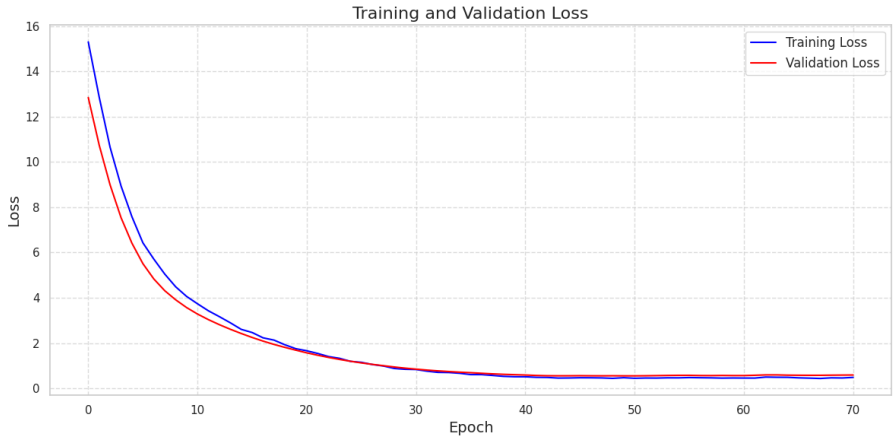


**Figure 7.** Huber Loss Function Across Epochs for Training and Validation Data in Model 1

Figure 8 shows the training and validation plots based on the Huber Loss function for model 2 which includes Change in Inflation Rate and Seasonally Adjusted GDP Growth Rate. Now the plots for training and validation data are gradually decreased and the model learns over the entire 300 epochs with a convergence point around 150 epochs. Both plots generally converge very well but show diminishing returns in loss reduction during the last epochs indicating room for earlier stopping in training. In contrast to model 1, this model shows a higher advantage with more epochs as the model learns over the full 300 epochs, indicating good generalization ability of the model on unseen data and no signs of overfitting.

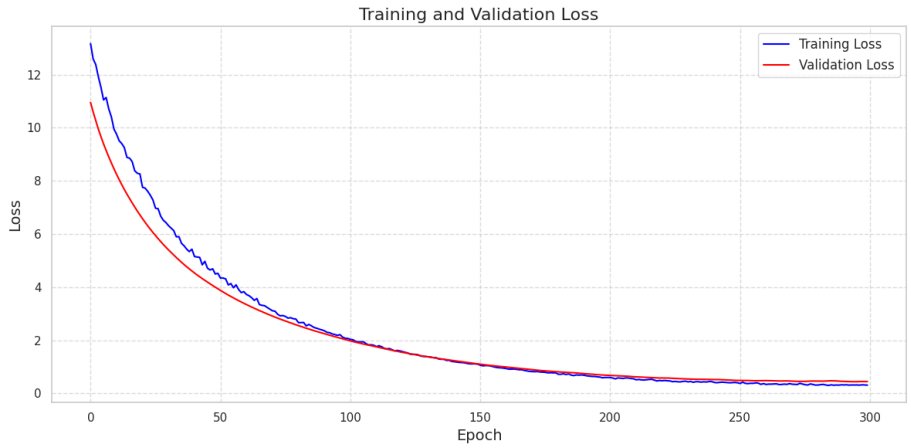


**Figure 8.** Huber Loss Function Across Epochs for Training and Validation Data in Model 2

Figure 9 shows the training and validation plots based on the Huber Loss function for model 3 which now includes Change in Inflation Rate, Seasonally Adjusted GDP Growth Rate, and Change in SEK/USD Spot Exchange Rate. Like model 2, both the plots for training and validation data are shown to decrease gradually in this model. The plots converge relatively well with a convergence point at about 100 epochs and then very little divergence is noticeable between the plots over the rest of the epochs. This indicates a very slight overfitting that model 3 has on the training data and thus does not indicate a strong generalization on unseen data. At the same time, it must be mentioned that in general, the model shows good generalization and not particularly high overfitting which means that the results from this model can be used for relevant conclusions in this thesis.

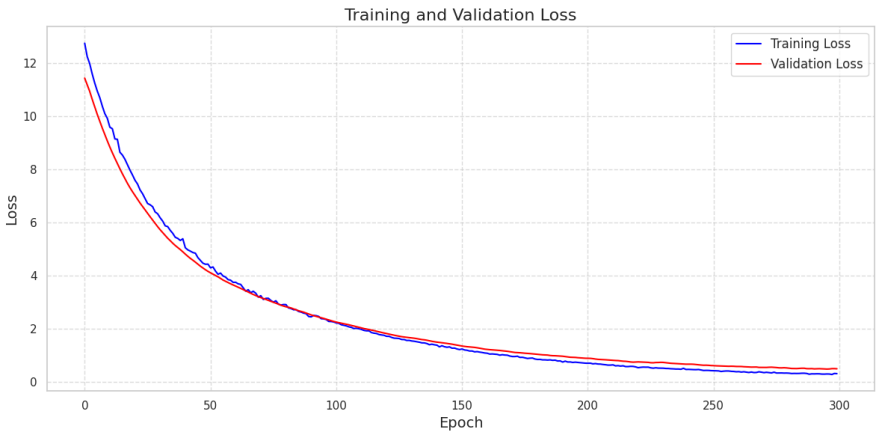


**Figure 9.** Huber Loss Function Across Epochs for Training and Validation Data in Model 3

Figure 10 shows the training and validation plots based on the Huber Loss function for model 4 which now includes all the macroeconomic variables in this thesis. This model unlike models 2 and 3 shows a rapid decrease for both plots with a convergence point of around 40 epochs and stabilization of around 150 epochs. The plots converge quickly and generally show robust performance, but after about 150 epochs to about 220 epochs of training, a slight divergence between the training and validation plots is observed. This divergence indicates a slight overfitting of the model, indicating less generalization to the test data. In conclusion, the last model with all macroeconomics variables has generally shown good generalization and robust performance, but the differences between the explanation of the variation in the training data and the test data indicate a slight overfitting of the model, this will also be shown using graphs and tables in the results section for greater clarity and transparency.

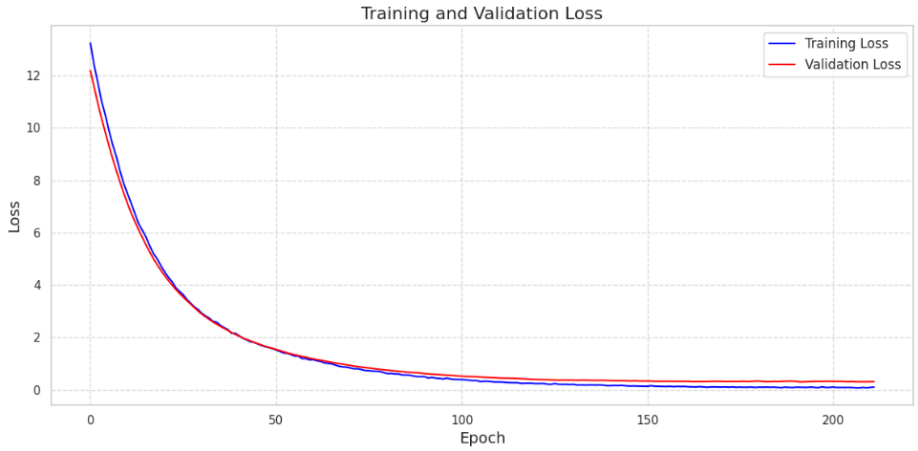


**Figure 10.** Huber Loss Function Across Epochs for Training and Validation Data in Model 4

### 5.3.6 Regularization Techniques

Regularization is a collection of techniques for dealing with overfitting in a neural network. Common regularization techniques used in neural networks and used in this thesis are ℓ2-regularization, Batch Normalization, Dropout, and Early stopping. The ℓ2 method is a common regularization method that is used to add a term in the cost function to reduce the complexity of the model against a complex data set that would potentially increase the risk of overfitting. Batch Normalization is a regularization method for stabilizing and improving the learning speed of the model. The method is applied in the network layers to normalize the output data within each neuron within each mini batch. Dropout is a method with a different approach that involves removing neurons from a layer during the training period. The model is trained with a varying number of random neurons at each batch during the training period to reduce overfitting. Finally, early stopping is a method that is not used directly for overfitting but for stopping a model from learning when the loss function is minimized, which can mitigate overfitting before it becomes too bad (Michelucci, 2022; El-Amir & Hamdy, 2020).

In this thesis, ℓ2-regularization has been used in all models on the second Convolutional layer, both BiLSTM layers and the Dense layer. Batch Normalization is applied as a layer in TensorFlow which we use in this thesis for stabilization and improvement of learning rates. Five Batch Normalization layers have been added to each model, after the Convolutional main branch two layers are added, after the BiLSTM main branch two layers are added and finally after the Dense layer only one layer is added. Dropout, which is also considered a layer of its own in TensorFlow, is applied in all four models in the thesis after the second Convolutional

layer, after each BiLSTM layer, and after the Dense layer. Finally, Early Stopping is applied in each model with a patience of 20, which means that the training is stopped after 20 epochs when the validation loss was last improved.

### 5.4 Model Evaluation Metrics

The models have been evaluated using various metrics to determine and draw conclusions about which model performs the best forecasts in the stock market. The metrics used are MSE (Mean Square Error), MAE (Mean Absolute Error), In-Sample $R^2$ ($R^2_{IS}$), and Out-of-Sample $R^2$ ($R^2_{OS}$). According to Willmott and Matsuura (2005), MSE is sensitive to large errors as this measure squares the errors, the measure is particularly relevant where the data includes larger errors such as in financial data or critical system monitoring. MAE is a much clearer and more direct measure of the magnitude of the average error and is less sensitive to large errors (outliers) compared to MSE. The study concludes that MAE is a more relevant measure of average error magnitude, especially if it's being used to compare model performance.

Campbell and Thompson (2008) analyzed the advantages and disadvantages of using In-Sample $R^2$ as a measure for evaluating models. In-Sample $R^2$ measures the model's fit within the training data, giving an overly optimistic picture of the model's performance since the measure does not reflect the model's performance in the test data (the unseen data). This can make the model's performance hyper-optimistic, and the measure itself is therefore not sufficient to evaluate the models but instead needs to be supplemented with another measure to provide a fairer picture of the evaluation. The measure that must be included in the analysis according to Campbell and Thompson (2008), to complete the evaluation is Out-of-Sample $R^2$, which measures the model's performance over the test data. This can give a fairer picture of the predictive power of the model on the unseen data. This measure is particularly useful in economic and financial contexts where even very small values of $R^2_{OS}$ can be important for investors and portfolio managers. Additionally, according to Hawinkel et al., 2023, $R^2_{OS}$ is an appropriate measure that reflects the differences in predictive power between different models.

To provide a fair and complete picture of the evaluation of the models in this thesis, all four metrics have been chosen to be used. The metrics MSE, MAE, In-Sample $R^2$, and Out-of-Sample $R^2$ complement each other and help us measure the error magnitude and the models' performance in both the training and test data to finally provide a conclusion about which model performs best in predicting the stock market. In the subsections below, all these measures used, and their respective mathematical equations will be presented in Equations 21, 22, 23, and 24.

### 5.4.1 Mean Square Error (MSE)

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2, \tag{21}$$

measures the average squared difference between the observed values and the predicted values where, *n* represents the number of forecasts, $y_t$ represents the observed values and $\hat{y}_t$ indicates the predicted values.

### 5.4.2 Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{t-1}^{n}|y_t - \hat{y}_t|, \tag{22}$$

calculates the average absolute difference between the observed and predicted values where, *n* represents the number of forecasts, $y_t$ represents the observed values and $\hat{y}_t$ indicates the predicted values.

### 5.4.3 In-Sample $R^2$ ($R^2_{IS}$)

$$R^2{}_{IS} = 1 - \frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{n}(y_t - \bar{y}_t)^2}, \tag{23}$$

measures the performance of the model in the training data by comparing the variance explained by the model to the total variance in the data. A $R^2_{IS}$ value closer to 1 indicates a perfect fit, while a value closer to 0 indicates a poor fit. In the formula, $y_t$ represents the observed values, $\hat{y}_t$ indicates the predicted values and $\bar{y}_t$ represents the average of the observed values.

### 5.4.4 Out-of-Sample $R^2$ ($R^2_{OS}$)

$$R^2{}_{OS} = 1 - \frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{n}(y_t - \bar{y}_{train})^2}, \tag{24}$$

measures the model's performance on unseen data (test data) by comparing the predictions with the observed values in the test data while using the average value from the training data as a baseline. In the formula, $y_t$ represents the observed values in the test data, $\hat{y}_t$ indicates the predicted values and $\bar{y}_t$ represents the average of the observed values in the training data.

# 6. Results & Analysis

## 6.1 Model Comparison

In this section, a performance overview of the four CNN-BiLSTM models is presented. The models are represented below in Figures 11, 12, 13, and 14 where each of them incorporates a varying set of variables to predict the OMXS30 index return. As they are progressively added to the model, the results significantly improve and provide deeper insights into the variables influencing the Swedish stock market.

Figure 11 presents the first CNN-BiLSTM model, illustrating the training and test sets. The model contains the change in inflation rate variable combined with its derived variables generated through feature engineering. As observed in figure A.13 in Appendix C, the first model reports the "coefficient of determination" ($R^2$) with a value of 20% for the training set, indicating that the model only explains 20% of the variance in the OMXS30 index return. In comparison to the training set, the test set associated with the model exhibits the "out of sample coefficient of determination" ($R^2_{OS}$) with a value of 4.9% using the provided variables. This implies that the first model captures 4.9% of the variance in OMXS30 returns, but the model appears to have a limited ability to explain higher variation. This can be seen in Figure 11, where the test set predicted values (red line) significantly differ from the actual values (green line), which aligns with the model $R^2_{OS}$ of 4.9% to unseen data. Additionally, the model does not completely overfit the training data, but it demonstrates a relatively low generalization since $R^2_{OS}$ is a positive small value. This indicates that the first model reveals a minor amount of predictive power in the unseen data, reflecting its ability to generalize learned patterns on the new data. Hence, as demonstrated in Figure 11, the test set also reveals that the model struggles to capture the market pattern, neither the extreme nor the steady movements.

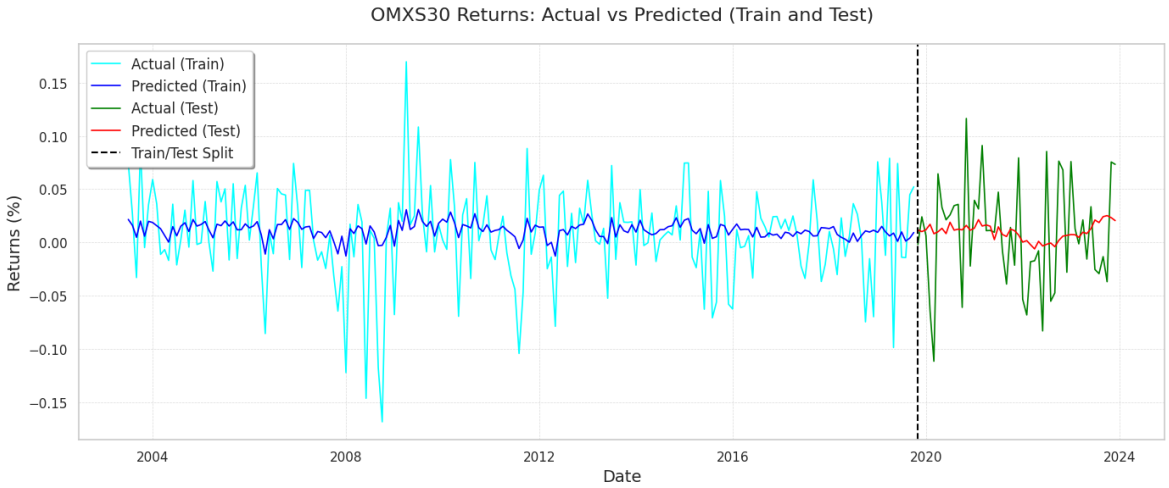


**Figure 11.** Dynamic Evaluation of OMXS30 Index Returns: Actual vs Predicted (Model 1)

Figure 12 presents the second CNN-BiLSTM model, analyzed with both training and test sets. This model includes variables, defined as the change in inflation rate, seasonally adjusted GDP growth rate, and its modified values. According to Figure A.13 in Appendix C, the model demonstrates a higher "coefficient of determination" ($R^2$) in the training set, with a value of 61.3%, compared to the previous model with a value of 20%. This implies that the second model explains 61.3% of the variation in the OMXS30 index return after incorporating an additional variable. As observed, the test set associated with the model exhibits an "out of sample coefficient of determination" ($R^2_{OS}$) with a value of 7.5%, which is 2.6% higher than the first model. Thus, this suggests that the second model captures 7.5% of the variance in OMXS30 return. As provided in Figure 12, the predicted values (red line) markedly differ from the actual values (green line), indicating that the model still struggles to generalize learned patterns on unseen data. In addition, the test set applied to this model exhibits similar limitations to the previous model in capturing the market pattern, especially the extreme movements. As seen in Figure 12, for example, when the actual test return (green line) shows significant peaks and dips, the predicted test return (red line) either flattens out or reacts insufficiently. Despite this, the second model's performance and generalization are slightly improved in the test sets, after adding the GDP growth rate variable with its modified values.

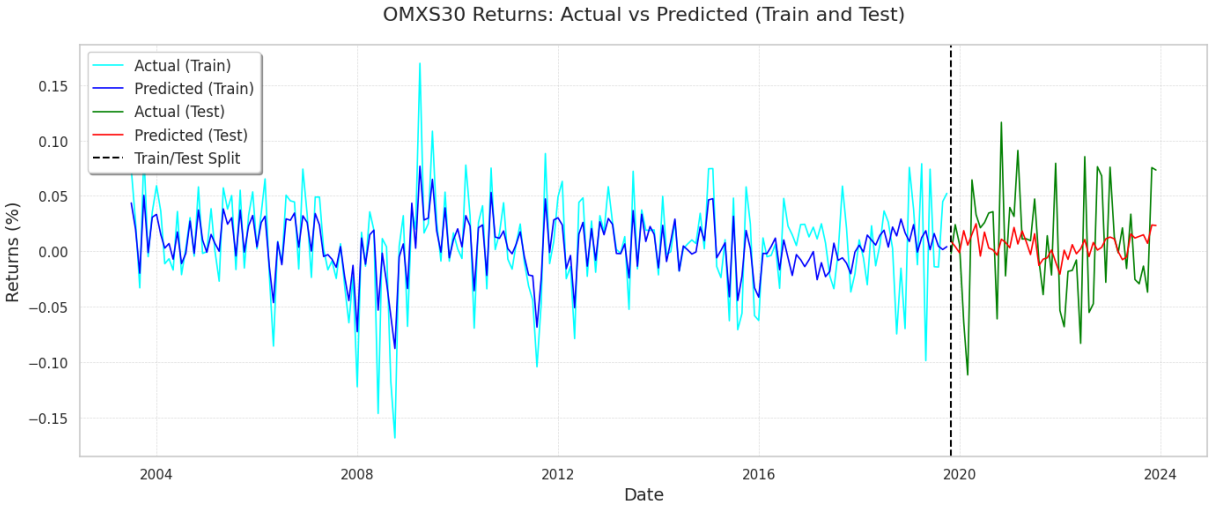


**Figure 12.** Dynamic Evaluation of OMXS30 Index Returns: Actual vs Predicted (Model 2)

The third CNN-BiLSTM model is presented in Figure 13, where the model's training and test sets are illustrated. An additional variable has been added, namely the change in SEK/USD spot exchange rate with its modified values, along with the change in inflation rate and seasonally adjusted GDP growth rate. In accordance with figure A.13 in Appendix C, the third model's training set achieves a significantly higher "coefficient of determination" ($R^2$) than previous models, with a value of 71.2%, explaining 71.2% of the variance in the OMXS30 index returns. Furthermore, the analysis of the third model's test set reveals the "out of sample coefficient of determination"($R^2_{OS}$) with a value of 10.2%, compared to the second model. As incorporating the spot exchange rate variable, the third model could increase $R^2_{OS}$ by 2.7% relative to the previous model, explaining 10.2% of the variance in OMXS30 returns. Despite improvements, generalization to the test set remains weak which in turn limits the model's ability in predicting unseen new data. However, $R^2_{OS}$ is relatively low compared to $R^2$ with value of 71.2%, which points out a potential overfitting problem in the third model and reduced the model's capacity to generalize. Although, as illustrated in Figure 13, the predicted return values (red line) generally follow the direction of the actual return values (green line) during periods of lower volatility. This in turn suggests that the model captures the market pattern, particularly its steady movements. By contrast, it is observed that the model struggles to reflect the extreme movements during higher volatility where the red line reacts weakly to them.

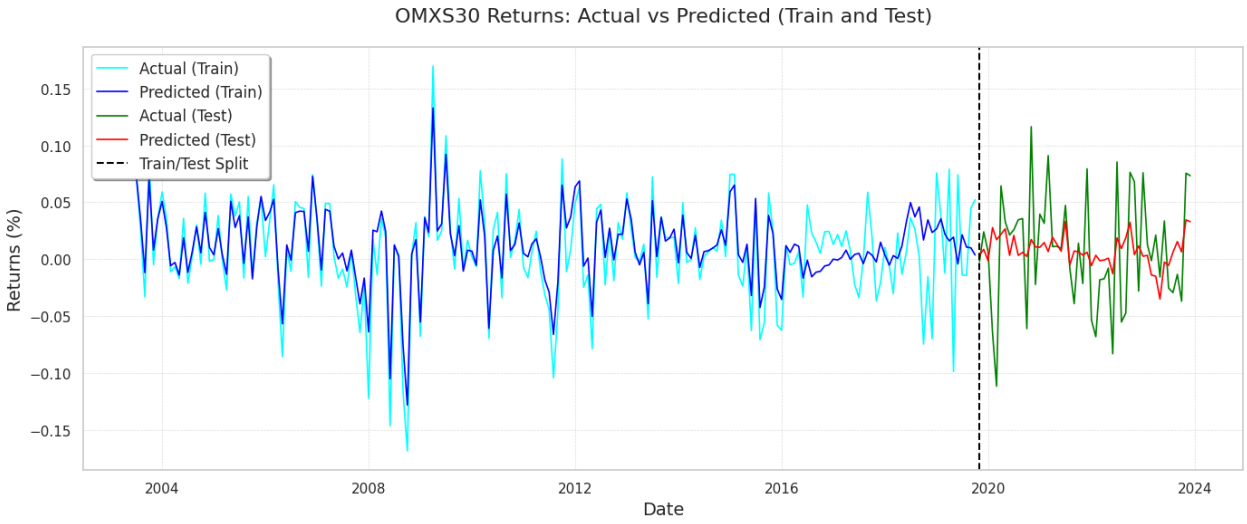


**Figure 13.** Dynamic Evaluation of OMXS30 Index Returns: Actual vs Predicted (Model 3)

Figure 14 represents the fourth CNN-BiLSTM model, illustrating the training and the test sets. This model incorporates all the selected macroeconomic variables for this thesis, which are the change in inflation rate, seasonally adjusted GDP growth rate, the change in SEK/USD spot exchange rate, and seasonally adjusted unemployment rate with its derived features. In figure A.13 in Appendix C, the model's training test demonstrates a slightly higher value of the "coefficient of determination" ($R^2$) than the third model. As observed, the fourth model generates a 72.4% explanation of the variance in the OMXS30 index returns. A high value of $R^2$ reflects the model's ability to learn effectively from the training data. Figure 14 also illustrates the test set evaluated by the fourth model, generating the "out of sample coefficient of determination value" ($R^2_{OS}$) of 12.1% which accounts for the variance in OMXS30 returns. This improvement by 1.9% from the previous model, is due to the additional unemployment rate variable with its derived features, resulting in a greater generalization to unseen data. Despite the enhanced explanatory power, the significant gap between the training set $R^2$ and the test set $R^2_{OS}$ indicates that the model still overfits the training data. Additionally, the model's test set reasonably captures steady market movements during periods with low volatility. Conversely, the model still encounters obstacles in reflecting the extreme movements during periods with higher volatility such as peaks and dips.

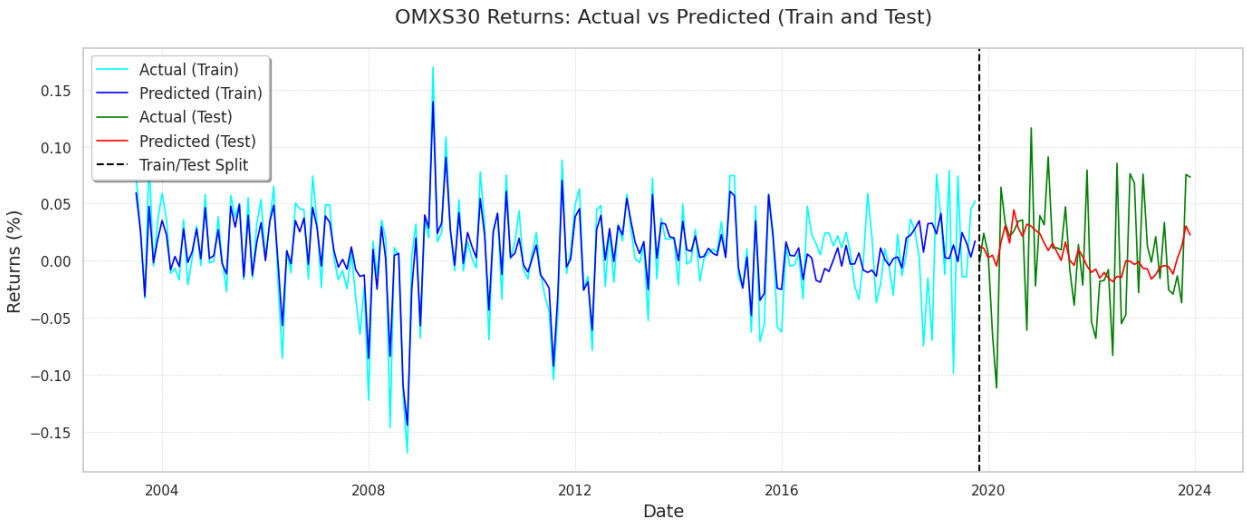


**Figure 14.** Dynamic Evaluation of OMXS30 Index Returns: Actual vs Predicted (Model 4)

## 6.2 Residual Analysis

In this section, a residual overview of the four CNN-BiLSTM models is presented. The residual plots for each model are illustrated below in Figures 15, 16, 17, and 18. These plots compare the predicted values on the y-axis with the actual values on the x-axis for both the training set, marked as blue points and the test set, marked as red points. As shown in these residual plots, a dashed diagonal line is drawn in each of them, representing the ideal fit line where the predicted values match the actual values.

Figure 15 illustrates the residual plot for the first CNN-BiLSTM model, including the change in inflation rate variable combined with its derived variables generated through feature engineering. The first model's residual plot exhibits most of the data points, including the training and test, clustered near the center around 0 on both axes. This indicates that the model generalizes better for smaller return values, with similar deviations from the ideal fit line observed for both the training and test sets, implying that the model is not fully overfitted. Yet, the residual plot exhibits a weak generalization, especially for the larger positive and negative return values. As seen in Figure 15, the model underestimates larger returns and outliers, as the points from test sets are significantly scattered away from the ideal fit line, which aligns with the models $R^2_{OS}$ with a low value of 4.9 %.

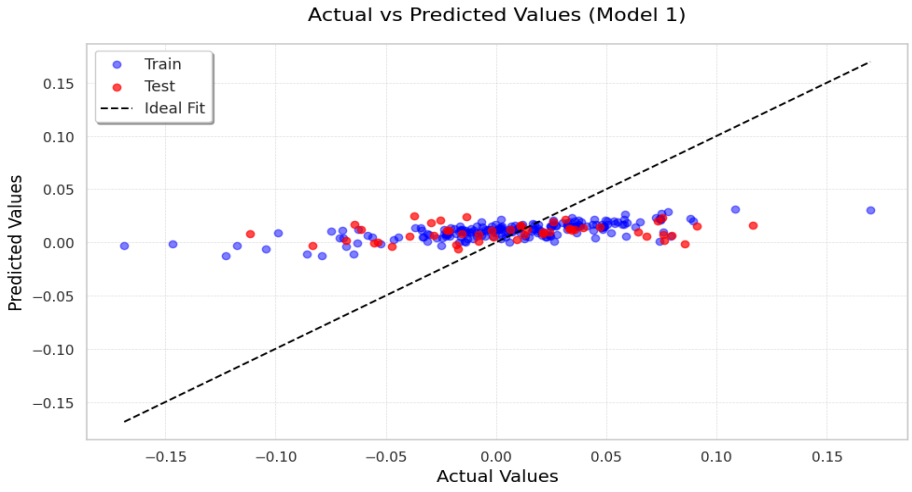


**Figure 15.** Residual Plot of Actual vs Predicted Values for Model 1

Figure 16 represents the residual plot for the second CNN-BiLSTM model, containing the change in inflation rate variable along with the seasonally adjusted GDP growth rate variable and its modified values. As observed in the residual plot, most of the data points from the training and test sets are located near 0 on both axes. This thereby indicates that the model achieves relatively better performance for smaller return values, with less scatter around the ideal fit line compared to the first model. Given that the blue and red points cluster closely around 0 where the test points deviate consistently from the ideal fit line for smaller returns, similar to the training points, it implies that the model is not overfitted. Despite these improvements, the residual plot also demonstrates an underestimation of both large positive and negative return values, as the points from the test set are scattered away from the fit line, limiting the model's ability to generalize.

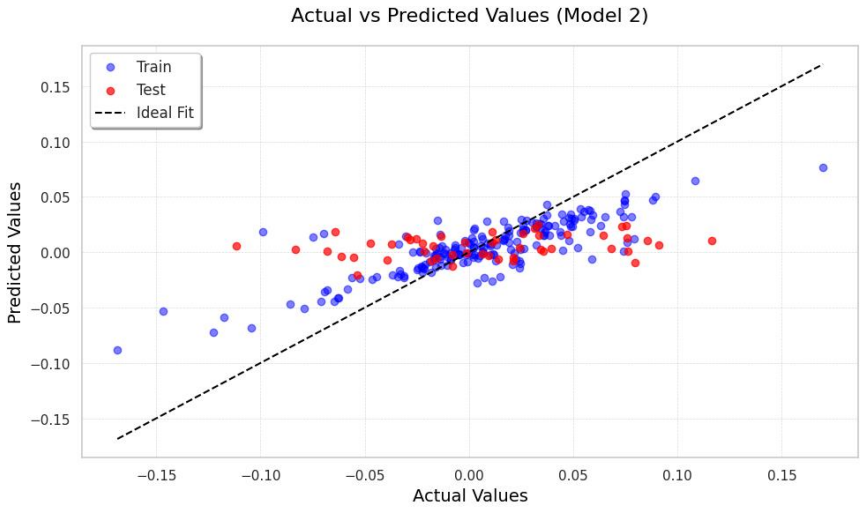


**Figure 16.** Residual Plot of Actual vs Predicted Values for Model 2

Figure 17 presents the residual plot for the third CNN-BiLSTM model, incorporating the variables defined as the change in SEK/USD spot exchange rate with its modified values, combined with the change in inflation rate and seasonally adjusted GDP growth rate. Following Figure 17, the model's residual plot illustrates red and blue points closely aligned near 0 on both axes. The model predicts smaller and steady returns, which contributes to a diminished scatter relative to the fit line, suggesting that this model demonstrates greater alignment with the actual values compared to previous models. As evidenced by Figure 17, the model has therefore enhanced its ability to generalize on smaller returns. However, the generalization is weaker and less accurate for larger values as the test set points deviate significantly from the

line, underestimating the higher positive and negative returns in the model. Furthermore, it should be noted that the third model exhibits a small overfitting problem, as observed in the residual plot in Figure 17, where the test points scatter more inconsistently from the ideal fit line for larger returns than the training points.

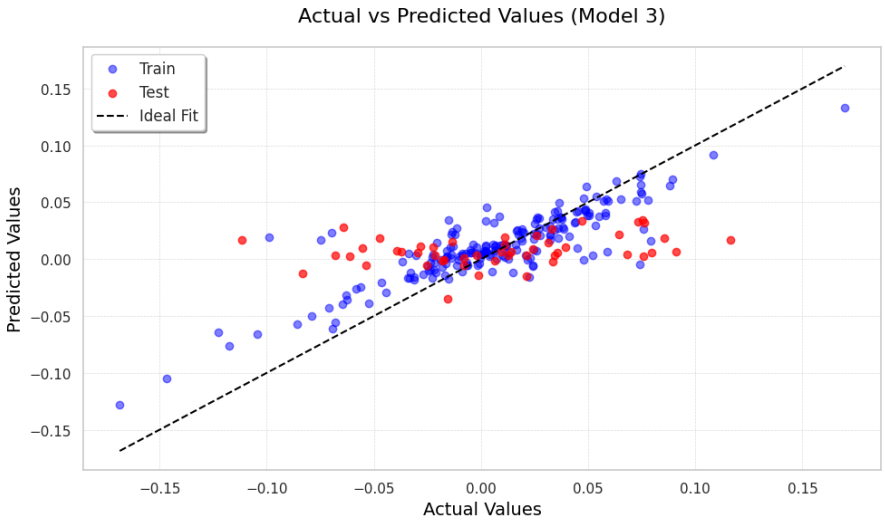


**Figure 17.** Residual Plot of Actual vs Predicted Values for Model 3

Figure 18 demonstrates the fourth model's residual plot, including all the selected macroeconomic variables which are the change in inflation rate, seasonally adjusted GDP growth rate, the change in SEK/USD spot exchange rate, and seasonally adjusted unemployment rate with their respective derived features. The model's residual plot exhibits the red and blue points closer to the ideal fit line for smaller and steady return values, maintaining consistency between the training and test sets. This thereby implies that generalization is improved, since the residual scatter appears less significant for test set points around 0, aligning with the model's $R^2_{OS}$ with a higher value of 12.1 %. In contrast, for larger positive and negative return values, the test set points demonstrate a significant scatter away from the ideal fit line, implying diminished generalization. As additionally observed in Figure 18, the red test points are slightly more deviated from the line compared to the blue training points, particularly for larger and extreme values, which in turn highlights a mild overfitting problem.

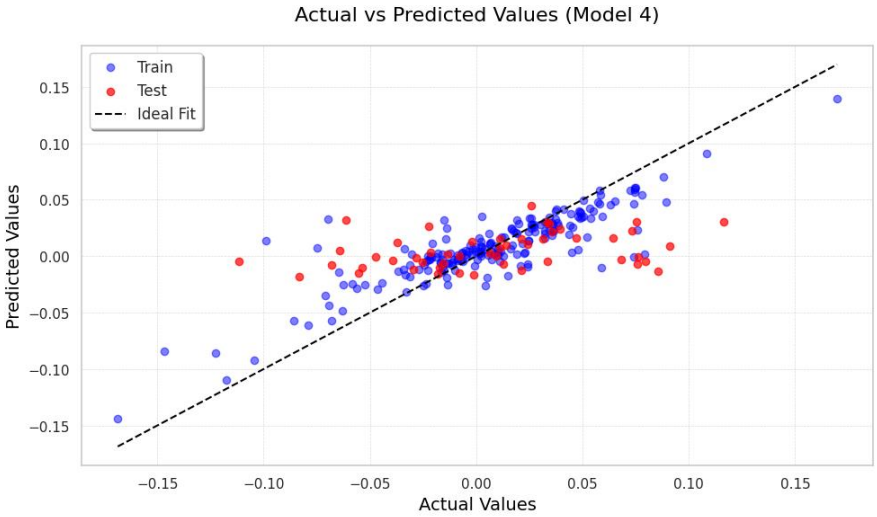


**Figure 18.** Residual Plot of Actual vs Predicted Values for Model 4

Figure A.14 in Appendix C illustrates a comparison overview of the training and test mean squared error (MSE) across the models, to measure each model's average squared difference between its predicted and actual values. According to the analysis in Figure A.14, in the first model, test MSE exhibits a moderately higher value than training MSE which indicates an increased prediction error and a weak generalization. In the second model, the training and test MSEs decreased to 0.0008 and 0.0023, respectively, after adding an additional variable. Although the model captures more of the training data pattern, generalization remains weak. Furthermore, in the third model, the training and test MSEs decrease to 0.0006 and 0.0022, respectively, which exhibits a minimal improvement in generalizing unseen data compared to models 1 and 2. As seen the test MSE does not decrease proportionally to the training MSE as the model improves and incorporates more variables. The test MSE remains instead significantly higher, which suggests that the model has a limited ability to generalize unseen data, highlighting the possibility of overfitting in the third model. In the fourth model, similar results are observed as in the third model where the addition of a new variable to the fourth model does not contribute to more accurate predictions of the new data, indicating a persistent overfitting problem.

Figure A.15 in Appendix C illustrates a comparison overview of the training and test mean absolute error (MAE) across the models, to evaluate each model's average absolute difference between its predicted and actual values. In accordance with Figure A.15, the first model demonstrates the highest training and test MAEs with a value of 0.0300 and 0.0388,

respectively, compared to other models. This indicates that the first model struggles to capture prediction in the training data and accurately generalize the patterns in the test data. Given that the generalization gap, defined as the difference between training and test MAE, is relatively small, it suggests that the model does not fully overfit. In the second model with an additional variable, a disproportionate reduction in both training and test MAE is observed compared to the previous model, which exhibits a significantly greater prediction in the training data but a limited improvement in generalizing unseen data. With the addition of another variable in the third model, the training MAE decreases from 0.0203 to 0.0169, resulting in higher prediction in the training data. Yet, the reduction in the test MAE is less significant, indicating a slight enhancement in generalization. In the fourth model, where all the variables are incorporated, the model generates the lowest training MAE, performing the best prediction in the training data. However, the test MAE does not improve proportionally as the training MAE among all the models. As observed in the fourth model's test MAE, a small decline is demonstrated, still indicating a minor improvement in generalizing unseen data. Also, the figure exhibits a wider generalization gap between the training and test MAE in models 3 and 4, implying overfitting problems. Finally, a general overview of all models and their respective values is given in Table A.2 in Appendix C which illustrates the performance metrics of four CNN-BiLSTM models, including training $R^2$, $R^2_{OS}$, training MSE, test MSE, training MAE, and test MAE, to compare the model's predictive accuracy and error levels.

# 7. Discussion

This section will present various parts, such as the interpretation of the results, the strengths and weaknesses of the models, and the limitations encountered during the thesis process. Finally, this section will conclude by providing suggestions for future research and potential applications of this thesis as a foundation for further studies.

## 7.1 Interpretation of Results

The results represented both via graphical and residual analyses enabled a clear picture of all models built in this thesis. The models were constructed by the same architecture with variables incrementally added one at a time, enabling the capture of the variable's main effects in predicting OMXS30. The first model with only one main variable: Change in Inflation Rate shows a predictive power of 4.9% in the unseen data. The model has a relatively good generalization ability and relatively high performance where the model shows no signs of overfitting. This indicates that the first CNN-BiLSTM model that fed with the inflation change data has been able to extract patterns and predict unseen data in OMXS30 with an accuracy of 4.9%. These results for the first model suggest that 4.9% of the variability in the OMXS30 stock market can be attributed to changes in inflation, as captured by the predictive capacity of the first CNN-BiLSTM model. This indicates that inflation changes contribute to a small but measurable portion of the stock market's volatility.

The second model with two main variables: Change in Inflation Rate and Seasonally Adjusted GDP Growth Rate has a predictive power of 7.5% in the unseen data. The second CNN-BiLSTM model, like the first model, shows no signs of overfitting and has a relatively high performance now with an increase in predictive power of 2.6%. This indicates that this predictive increase is directly attributable to the Seasonally Adjusted GDP Growth Rate. This means that GDP growth accounts for 2.6% of the variability in the OMXS30 stock market, indicating that the new variable added moderately contributes to stock market volatility but not as much as inflation changes. The third model with three main variables: Change in Inflation Rate, Seasonally Adjusted GDP Growth Rate, and Change in SEK/USD Spot Exchange Rate showed further improvement with a predictive power of 10.2%. The third CNN-BiLSTM model has a good generalization ability but as previously mentioned under the subsection Number of Epochs in the Method, this model shows very slight signs of overfitting which may mean a

slightly worse performance during the test data. This may compromise to some extent the predictive increase that would be attributed to the newly added variable Change in SEK/USD Spot Exchange Rate as overfitting to the training data may exaggerate the predictive power in the test data. At the same time, it is important to mention that the signs of overfitting are very small and that the assignment of the predictive power to the change in SEK/USD spot exchange is still relevant and interesting in this context. The relative increase in predictive power is 2.7% which indicates a relatively high proportion of the variability of OMXS30 is assigned to the variable Change in SEK/USD Spot Exchange. While this is slightly larger than the contribution of changes in GDP growth of 2.6%, it remains lower than the contributions from changes in inflation of 4.9%.

The fourth and final model with all the macroeconomic variables in this thesis: Change in Inflation Rate, Seasonally Adjusted GDP Growth Rate, Change in SEK/USD Spot Exchange Rate, and Seasonally Adjusted Unemployment Rate shows the highest predictive power of 12.1% in the unseen data. Making this CNN-BiLSTM model the best and strongest model based on prediction in OMXS30 stock market. The relative increase in predictive power is 1.9% which is assigned to the variable Seasonally Adjusted Unemployment Rate. What must be mentioned is that this model also exhibits signs of overfitting, which is slightly larger than the overfitting in model 3. This may indicate a compromise in the predictive power assigned to the unemployment rate. This means that the predictive power assigned to the last variable, which was also the lowest among all variables, should be viewed with some critical eye due to the slight overfitting that may affect the isolation of the main effect.

The last model, which includes all the macroeconomic variables, exhibits the highest predictive power of 12.1% on the unseen data in the OMXS30 stock market. A prediction of 12.1% in a volatile market that shows sensitivity to a variety of economic, political and psychological factors is a great achievement. The largest proportion of the total predictive power of 12.1% is distributed to the Change in Inflation Rate with an explanation of the unseen data of 4.9%. The second most important variable is the Change in SEK/USD Spot Exchange Rate with a predictive power of 2.7%. The third most important variable is the Seasonally Adjusted GDP Growth Rate with a predictive power of 2.6% and lastly the Seasonally Adjusted Unemployment Rate with a predictive power of only 1.9%. These results are compatible with previous research as previously mentioned in the study which has shown that inflation, inflation changes and inflation expectations tend to reflect themselves in stock markets and thus have high predictive power in these markets. While changes in exchange rate and GDP growth have

a relevant but relatively less predictive power compared to changes in inflation. Finally, the unemployment rate shows the least predictive power, which is also compatible with previous research that argues that the unemployment rate can have an indirect effect on the stock market and thus does not show a stronger and more efficient relationship to these types of markets.

### 7.2 Overfitting and Generalization

As previously mentioned, the generalization of the models and phenomena such as under- and overfitting are very important to discuss in the context of machine learning and deep learning. The results of all models in this thesis have shown that the first two models do not exhibit signs of overfitting and therefore show high generalization ability but at the same time lower predictive power. On the other hand, the last two models show slightly overfitting and relatively lower generalization ability but at the same time higher predictive power compared to the first two models. This means that when the models are introduced to a larger amount of data, the complexity becomes higher and thus the models are more exposed to problems with overfitting. Although regularization techniques have been used to counteract overfitting, overfitting could not be counteracted up to 100% in the case of the last two models.

Apart from the size and complexity of the data, the complexity of the architecture of the models following the complexity of the data can also create problems with overfitting. In this thesis, different configurations of hybrid deep learning models with convolutional layers and bidirectional LSTM layers are used to effectively learn the patterns in the OMXS30 stock market. This indicates that the model is very advanced and when this type of model is presented with a more complex dataset as in the last two models, the total complexity can become too high and thus lead to overfitting.

Additionally, from another perspective, the course of events during the test period can also affect the model's performance and hence the predictive power. The test set for all four models extends from November 2019 to December 2023 and this period coincides with the Coronavirus pandemic (Covid 19) that started in early 2020. The pandemic led to unusual and unprecedented market dynamics that could potentially have disrupted the relationship between macroeconomic variables and the return in OMXS30. This disruption between the macroeconomic variables incrementally fed into the models and the OMXS30 index leads to difficulties for the hybrid model to predict the test period because of the new more unusual patterns and trends that the models have not seen before. This shows, to some extent, the weaknesses of these predictive

models in deep learning. The models are very dependent on the data they are trained on, which means that if the data the model is trained on does not include similar patterns in the test data, the model will potentially have greater difficulty predicting the test data. This also indicates that these models cannot predict stock market shocks that are created due to unforeseen events. In a scenario where the model could be made to control unforeseen events in the economy and finance, the predictive power would possibly be much higher. Finally, it can be concluded that the hybrid models have had a slight problem with overfitting in the last two cases and that these models may have difficulty continuously predicting the stock market due to a variety of factors mentioned above. On the other hand, these models are not useless, on the contrary, these models can be used to gain a deeper understanding of how stock markets behave and what relationships macroeconomic indicators can have with stock markets.

### 7.3 Limitations

During the thesis, certain limitations have been encountered that could potentially have affected the results of the models. An important limitation that has had a direct effect on the models is data limitation. Advanced deep learning models require large amounts of data to be trained on to make accurate predictions. In this thesis, only 23 years of monthly data could be collected for all macroeconomic variables and the return of OMXS30. Feature Engineering and modifications of the data led to the total length of the data being only 20 years and 6 months. Based on this length, the data has been divided into an 80% training set and a 20% test set, which indicates that the model has not had a particularly large amount of data to train on which could potentially have affected the results. Under the scenario of having a larger amount of data, which was impossible for some macroeconomic variables, the model would certainly perform better. The data has been collected in monthly frequency which made it even more difficult to find data before the year 2000. The frequency of the data is also important where instead of monthly frequency as used in this thesis, daily frequency or weekly frequency would be more effective in teaching the model the patterns in the data. However, this was also very difficult as the data frequencies for the macroeconomic variables are often only in annual or monthly frequency for this data length.

Another limitation that is not as serious as data limitation, but which also turned out to have a negative effect from a practical point of view is the computer's computational power. Advanced deep learning models require powerful computers to perform the training on, which can also be

energy and time-consuming. At the same time, adjusting the hyperparameters requires training multiple models to see which values fit in this context. During this thesis, not much time was available, and the computer used for training the models did not have a particularly high computational power, which led to the training of the models requiring even longer time, which complicated the time-consuming situation in this thesis even more.

### 7.4 Suggestions for Future Research

Research incorporating deep learning and finance requires a good understanding of the dynamics of financial markets as well as programming skills for building and evaluating models. This thesis contributes a foundational understanding of the extent to which domestic macroeconomic variables can predict the stock market. The new foundational knowledge discovered in this thesis is of course built on the enormous knowledge pyramid of deep learning in economics and finance. This enormous knowledge pyramid has gained enormous momentum in recent times, driven by major advancements in AI (Artificial Intelligence), which means that this knowledge can in turn be used to build on the pyramid and thereby come up with new theorems that can better explain the dynamics of the financial market. One way this thesis can be used as a basis for future research is to use models with a similar architecture in other financial markets such as the S&P 500 and instead of only domestic macroeconomic variables also include other global variables such as commodity prices and global stock indices. The domestic and global macroeconomic variables can also be combined with sentimental data from news articles, social media, or reviews that reflect people's feelings, thoughts, and expectations for the financial markets. This would potentially increase the accuracy of the predictive models in this context as financial markets are very complex and highly affected by several different factors at the same time which can lead to large and irrational shifts in the data.

From another perspective, this analysis can be developed by adding more domestic variables such as interest rates and energy prices, as well as introducing interaction variables to capture their combined effects. This would increase the understanding of how macroeconomic indicators affect stock returns. Additionally, the model built in this thesis can be the basis for further experimentation with the architecture of the model where you can add Attention mechanisms for very long-time dependencies or completely try Transformer models to see how the performance changes in predicting the stock market. Advanced models also require larger

amounts of data as previously mentioned, which means that techniques such as data augmentation or synthetic data are recommended for future research to both retain all data points in the dataset and augment the data artificially. As well as methods such as Cross-Validation can in turn be used to evaluate the model's performance and generalization ability.

Finally, another interesting component that could be added to this thesis and that is very recommended for future research is explainable AI (XAI) methods. According to Chen et al. (2023), hybrid models in deep learning are very powerful methods for predicting the future in financial markets but at the same time, their complicated structure makes them difficult to interpret. The interpretation of the results can be enabled via XAI methods such as Shapley Additive Explanations (SHAP) and Layer-wise Relevance Propagation (LRP) to address the model's results back to the input variables. This would explain the results of the model more deeply while at the same time allowing for further improvements by understanding which variables contribute most to the model's performance.

# 8. Conclusion

This section summarizes the meaning and findings of the entire thesis and briefly shows what implications these findings may have in economics and finance.

## 8.1 Summary of Main Findings

The purpose of this thesis was to gain an understanding of the extent to which domestic macroeconomic indicators can explain stock market variability and to see which of the variables exhibits the highest predictive power in explaining OMXS30 stock market performance. The fourth and final CNN-BiLSTM model that includes all the macroeconomic variables in the thesis shows the highest predictive power of 12.1% on the unseen data. The most important variable that exhibits the highest predictive power among all other variables is the Change in Inflation Rate, contributing 4.9% to the overall predictive ability. However, the last two models showed slight signs of overfitting even when regularization techniques were used, which may question the generalizability of the models and the exaggeration to some extent of the predictive power. The hybrid deep learning models built in this thesis show interesting and relevant results but at the same time, these models unfortunately have nothing magical and thus have difficulties in continuously predicting the future in the financial market. The knowledge of financial time series and deep learning is constantly growing and with this thesis, the aim has been to further show a new aspect of how these areas can be combined to understand phenomena within the economic and financial world.

## 8.2 Implications for Finance and Economics

The results presented in this thesis and the model used with its respective architecture have implications in several different areas which also shows the importance of further research in the field of financial technology. Predictive accuracy about the future enables us to make more informed and effective decisions in the present. These decisions can be practically used by states for more concise policymaking, hedge funds, and investment banks for optimization problems, and technical analysts to strengthen their strategies for assessing market movements.

## References


Adil, M., Ullah, R., Noor, S., & Gohar, N. (2022). Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design. *Neural Comput & Applic, 34*, 8355–8363. https://doi.org/10.1007/s00521-020-05305-8

Afaq, S., & Rao, S. (2020). Significance of epochs on training a neural network. *International Journal of Scientific & Technology Research, 9*(6), 485. Retrieved 2020-06-01, from: https://mail.ijstr.org/final-print/jun2020/Significance-Of-Epochs-On-Training-A-Neural-Network.pdf

Alsterlind, J. (2006). *Sambandet mellan KPI och underliggande inflation i Sverige*. Sveriges Riksbank. Retrieved 2024-11-23, from: https://www.riksbank.se/globalassets/media/rapporter/pov/filer-fore2017/artiklar/jan_alsterlind_060421_sve.pdf

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data, 8*(1), Article 53. https://doi.org/10.1186/s40537-021-00444-8

Arnold, C., Biedebach, L., Küpfer, A., & Neunhoeffer, M. (2024). The role of hyperparameters in machine learning models and how to tune them. *Political Science Research and Methods, 12*(4), 841–848. https://doi.org/10.1017/psrm.2023.61

Benzoni, L., Collin-Dufresne, P., & Goldstein, R. (2007). Portfolio choice over the life cycle when stock and labor markets are cointegrated. *The Journal of Finance, 62*(5), 2123–2163. Retrieved 2024-11-24, from: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2007.01271.x

Board of Governors of the Federal Reserve System (US). (2024). Swedish Kronor to U.S. Dollar Spot Exchange Rate. Retrieved 2024-11-07, from:

https://fred.stlouisfed.org/series/EXSDUS

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th ed.). John Wiley & Sons. Retrieved 2024-12-06, from: https://www.wiley.com/enus/Time+Series+Analysis%3A+Forecasting+and+Control%2C+5th+Edition-p-9781118675021

Boyd, J. H., Hu, J., & Jagannathan, R. (2005). The stock market's reaction to unemployment news: Why bad news is usually good for stocks. *The Journal of Finance, 60*(2), 649–671. Retrieved 2024-11-16, from: https://www.jstor.org/stable/3694763

Brooks, C. (2008). *Introductory econometrics for finance* (2nd ed.). Cambridge University Press.

Campbell, J. Y., & Thompson, S. B. (2008). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies, 21*(4), 1509–1531. https://www.jstor.org/stable/40056860

Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In *Proceedings of the 2015 IEEE International Conference on Big Data* (pp. 2823–2824). IEEE. Retrieved 2024-12-01, from: https://ieeexplore.ieee.org/document/7364089

Chen, S.-S. (2009). Predicting the bear stock market: Macroeconomic variables as leading indicators. *Journal of Banking & Finance, 33*(2), 211–223. Retrieved 2024-11-23, from: https://www.sciencedirect.com/science/article/abs/pii/S0378426608001544

Chen, W., Hussain, W., Cauteruccio, F., & Zhang, X. (2023). Deep learning for financial time series prediction: A state-of-the-art review of standalone and hybrid models. *Journal of Financial Data Science, 15*(4), 123–156. https://doi.org/10.32604/cmes.2023.031388

Chen, W., Hussain, W., Cauteruccio, F., & Zhang, X. (2024). Deep learning for financial time series prediction: A state-of-the-art review of standalone and hybrid models. *Computer Modeling in Engineering & Sciences*, *139*(1), 189–217. https://doi.org/10.32604/cmes.2023.031388

Cheung, Y.-W., & Ng, L. K. (1998). International evidence on the stock market and aggregate economic activity. *Journal of Empirical Finance, 5*(3), 281–295. Retrieved 2024-11-24, from: https://people.ucsc.edu/~cheung/WorkingPapers/EquityEconFunda_JEF98.pdf

Ciampiconi, L., Elwood, A., Leonardi, M., Mohamed, A., & Rozza, A. (2023). A survey and taxonomy of loss functions in machine learning. *arXiv preprint*. https://doi.org/10.48550/arXiv.2301.05579

Cryer, J. D., & Chan, K. (2008). *Time series analysis: With applications in R* (2nd ed.). Springer.

Deverill, H., Scherer, W., Porter, M., & Stam, A. (2024). The utility of machine learning applied to military assessment and selection. *Military Operations Research, 29*(2), 53–94. https://www.jstor.org/stable/10.2307/27317896

Dreyfus, G. (2005). *Neural networks: Methodology and applications.* Springer. Retrieved 2024-12-07, from: https://link.springer.com/book/10.1007/3-540-28847-3

EL-Amir, H., & Hamdy, M. (2020). *Deep learning pipeline: Building a deep learning model with TensorFlow.* Apress. https://doi.org/10.1007/978-1-4842-5349-6

Fama, E. F. (1981). Stock returns, real activity, inflation, and money. *The American Economic Review, 71*(4), 545–565. Retrieved 2024-11-24, from: https://www.jstor.org/stable/1806180

Gonzalo, J., & Taamouti, A. (2014). The reaction of stock market returns to anticipated unemployment. Retrieved 2024-11-14, from: https://www.eco.uc3m.es/~jgonzalo/UnemploymentRateFinancialMarket.pdf

Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Pearson Education, Inc.

Hawinkel, S., Waegeman, W., & Maere, S. (2023). The out-of-sample $R^2$: Estimation and inference. *arXiv*. https://doi.org/10.48550/arXiv.2302.05131

Hussein, B. M., & Shareef, S. M. (2024). An empirical study on the correlation between early stopping patience and epochs in deep learning. *ITM Web of Conferences, 64*(01003). https://doi.org/10.1051/itmconf/20246401003

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). Monash University.

Investing.com. (2024). OMX Stockholm 30 (OMXS30). Retrieved 2024-11-07, from: https://www.investing.com/indices/omx-stockholm-30-historical-data

Khodabakhsh, A., Ari, I., Bakır, M., & Alagoz, S. M. (2020). Forecasting multivariate time-series data using LSTM and mini-batches. In M. Bohlouli, B. Sadeghi Bigham, Z. Narimani, M. Vasighi, & E. Ansari (Eds.), *Data Science: From Research to Application. CiDaS 2019. Lecture Notes on Data Engineering and Communications Technologies* (Vol. 45). Springer, Cham. https://doi.org/10.1007/978-3-030-37309-2_10

Kingma, D. P., & Ba, J. L. (2014). ADAM: A method for stochastic optimization. *arXiv preprint*. https://doi.org/10.48550/arXiv.1412.6980

Macukow, B. (2016). Neural networks – state of art, brief history, basic models and architecture. In *Computer Information Systems and Industrial Management. CISIM 2016. Lecture Notes in Computer Science (Vol. 9842)* (pp. 3–14). Springer, Cham. https://doi.org/10.1007/978-3-319-45378-1_1

Michelucci, U. (2022). *Applied deep learning with TensorFlow 2: Learn to implement advanced deep learning techniques with Python* (2nd ed.). Apress.

Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science, 170*, 1168–1173. https://www.sciencedirect.com/science/article/pii/S1877050920304865

Mortezapour Shiri, F., Perumal, T., Mustapha, N., & Mohamed, R. (2024). A comprehensive overview and comparative analysis on deep learning models. *Journal on Artificial Intelligence, 6*(4), 301–360. https://doi.org/10.32604/jai.2024.054314

Nelson, D. M. Q., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. In *Proceedings of the IEEE Symposium on Computational Intelligence* (pp. 1419–1426). IEEE. Retrieved 2024-11-28, from: https://ieeexplore.ieee.org/document/7966019

OECD. (2024). Real gross domestic product (GDP). Retrieved 2024-11-20, from: https://www.oecd.org/en/data/indicators/real-gross-domestic-product-gdp.html

Oh, M., Kim, C. K., Kim, B., & Kim, H.-G. (2024). A novel model to estimate regional differences in time-series solar and wind forecast predictability across small regions: A case study in South Korea. *Energy, 291*, 130284. https://doi.org/10.1016/j.energy.2023.130284

Organization for Economic Co-operation and Development (OECD). (2024). Infra-annual labor statistics: Monthly unemployment rate total: 15 years or over for Sweden. Retrieved 2024-11-07, from: https://fred.stlouisfed.org/series/LRHUTTTTSEM156S

Peral-García, D., Cruz-Benito, J., & García-Peñalvo, F. J. (2024). Systematic literature review: Quantum machine learning and its applications. *Computer Science Review, 51*, Article 100619. https://doi.org/10.1016/j.cosrev.2024.100619

Praveenkumar, A., Jha, G. K., Madival, S. D., Lama, A., & Kumar, R. R. (2024). Deep learning approaches for potato price forecasting: Comparative analysis of LSTM, Bi LSTM, and AM LSTM models. *Potato Research*. https://doi.org/10.1007/s11540-024-09823-z

Purkait, N. (2019). *Hands-on neural networks with Keras: Design and create neural networks using deep learning and artificial intelligence principles* (1st ed.). Packt Publishing.

Raiaa, M. A. K., Sakib, S., Fahad, N. M., Al Mamun, A., Rahman, M. A., Shatabda, S., & Mukta, M. S. H. (2024). A systematic review of hyperparameter optimization techniques in convolutional neural networks. *Decision Analytics Journal, 11*, Article 100470. https://doi.org/10.1016/j.dajour.2024.100470

Reimers, N., & Gurevych, I. (2017). Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks. *arXiv preprint*. https://doi.org/10.48550/arXiv.1707.06799

Statistics Sweden. (2022). Recovery in the labour market in 2022. Retrieved 2024-11-10, from: https://www.scb.se/en/finding-statistics/statistics-by-subject-area/labour-market/labour-force-surveys/labour-force-surveys-lfs/pong/statistical-news/labour-force-surveys-lfs-2022/

Statistics Sweden. (2024a). Time series on the unemployment rate, persons 15–74 years. Retrieved 2024-11-10, from: https://www.scb.se/en/finding-statistics/statistics-by-subject-area/labour-market/labour-force-surveys/labour-force-surveys-lfs/pong/tables-and-graphs/seasonally-adjusted-data/time-series-on-the-unemployment-rate-persons-15-74-years/

Statistics Sweden. (2024b). *Inflation i Sverige*. Retrieved 2024-11-18, from: https://www.scb.se/hitta-statistik/sverige-i-siffror/samhallets-ekonomi/inflation/

Statistics Sweden. (2024c). *Inflation in Sweden*. Retrieved 2024-11-20, from: https://www.scb.se/en/finding-statistics/statistics-by-subject-area/prices-and-consumption/consumer-price-index/consumer-price-index-cpi/pong/tables-and-graphs/consumer-price-index-cpi/inflation-in-sweden/

Statistics Sweden. (2024d). CPI: Annual changes (inflation rate). Retrieved 2024-11-21, from: https://www.scb.se/en/finding-statistics/statistics-by-subject-area/prices-and-consumption/consumer-price-index/consumer-price-index-cpi/pong/tables-and-graphs/consumer-price-index-cpi/cpi-annual-changes-inflation-rate/

Statistics Sweden. (2024e). National accounts, GDP indicator (ENS2010). Retrieved 2024-11-07, from: https://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START__NR__NR9999__NR9999A/NR9999ENS2010BNPIndN/table/tableViewLayout1/

Statistics Sweden. (2024f). CPI, fixed index numbers by month. Retrieved 2024-11-07, from: https://www.statistikdatabasen.scb.se/pxweb/en/ssd/START__PR__PR0101__PR0101A/KPItotM/table/tableViewLayout1/

Sveriges Riksbank. (n.d.). Inflationsmålet. Retrieved 2024-11-20, from: https://www.riksbank.se/sv/penningpolitik/inflationsmalet/

Sveriges Riksbank. (2020). Att analysera växelkurser – några centrala begrepp: Fördjupning i redogörelse för penningpolitiken 2019. Retrieved 2024-11-24, from: https://www.riksbank.se/globalassets/media/rapporter/rpp/svenska/2020/att-analysera-vaxelkurser--nagra-centrala-begrepp-fordjupning-i-redogorelse-for-penningpolitiken-2019.pdf

Taye, M. M. (2023). Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation*, *11*(3), 1. https://doi.org/10.3390/computation11030052

Telmem, M., Laaidi, N., Ghanou, Y., Hamiane, S., & Satori, H. (2024). Comparative study of CNN, LSTM and hybrid CNN LSTM model in Amazigh speech recognition using spectrogram feature extraction and different gender and age dataset. *Springer Science+Business Media*.https://doi.org/10.1007/s10772-024-10154-0

Tripathy, N. (2011). Causal relationship between macro-economic indicators and stock market in India. *Asian Journal of Finance & Accounting, 3*(1), E13. Retrieved 2024-11-24, from: http://dx.doi.org/10.5296/ajfa.v3i1.633

Verdonck, T., Baesens, B., Óskarsdóttir, M., & vanden Broucke, S. (2021). Special issue on feature engineering editorial. *Machine Learning, 113*(10), 3917–3928. https://doi.org/10.1007/s10994-021-06042-2

Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q., & Wang, J. (2021). A stock closing price prediction model based on CNN-BiSLSTM. *Complexity, 2021*, Article 5360828. https://doi.org/10.1155/2021/5360828

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research, 30*(1), 79–82. https://www.jstor.org/stable/24869236

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data mining: Practical machine learning tools and techniques* (4th ed.). Morgan Kaufmann.

Wooldridge, J. M. (2016). *Introductory econometrics: A modern approach.* Cengage Learning.

World Bank. (2023). GDP growth (annual %) – Sweden. Retrieved 2024-11-17, from: https://data.worldbank.org/indicator/NY.GDP.MKTP.KD.ZG?locations=SE

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review, 57*, 99. https://doi.org/10.1007/s10462-024-10721-6

# Appendices

## Appendix A: Dickey-Fuller Test and STL Decompositions

**Table A.1** Dickey-Fuller Test For Unit Root

| Variable | Test Statistic | 1% Critical Value | 5% Critical Value | 10% Critical Value | P-value |
|---|---|---|---|---|---|
| Change in Inflation Rate | -3.427 | -3.455 | -2.872 | -2.572 | 0.0101 |
| SA GDP Growth Rate | -15.220 | -3.453 | -2.872 | -2.572 | 0.0000 |
| Change in SEK/USD Rate | -6.591 | -3.454 | -2.872 | -2.572 | 0.0000 |
| SA Unemployment Rate | -3.402 | -3.454 | -2.872 | -2.572 | 0.0109 |
| OMXS30 Index Return | -8.327 | -3.454 | -2.872 | -2.572 | 0.0000 |

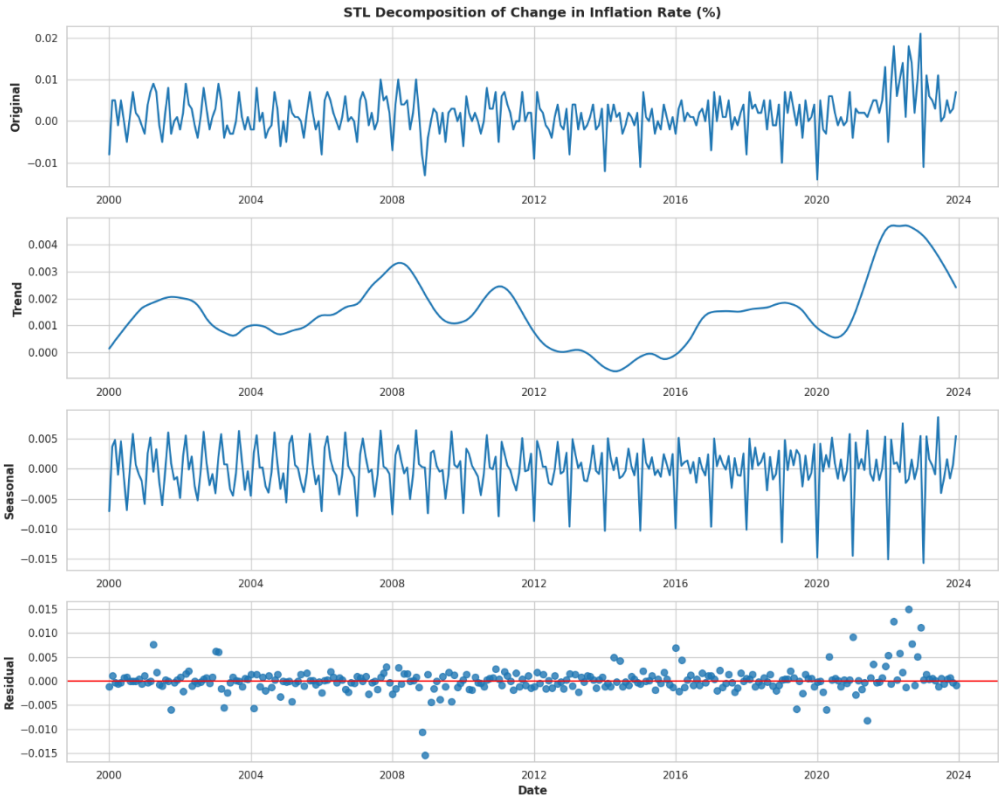


**Figure A.1** STL Decomposition of Change in Inflation Rate

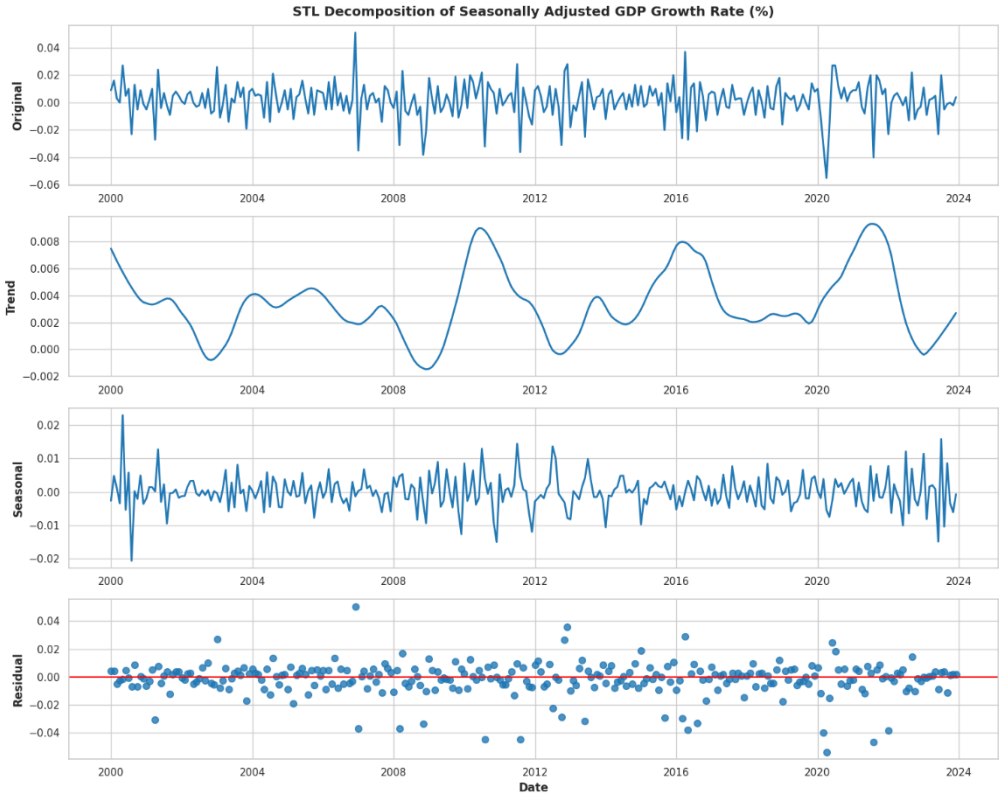


**Figure A.2** STL Decomposition of Seasonally Adjusted GDP Growth Rate

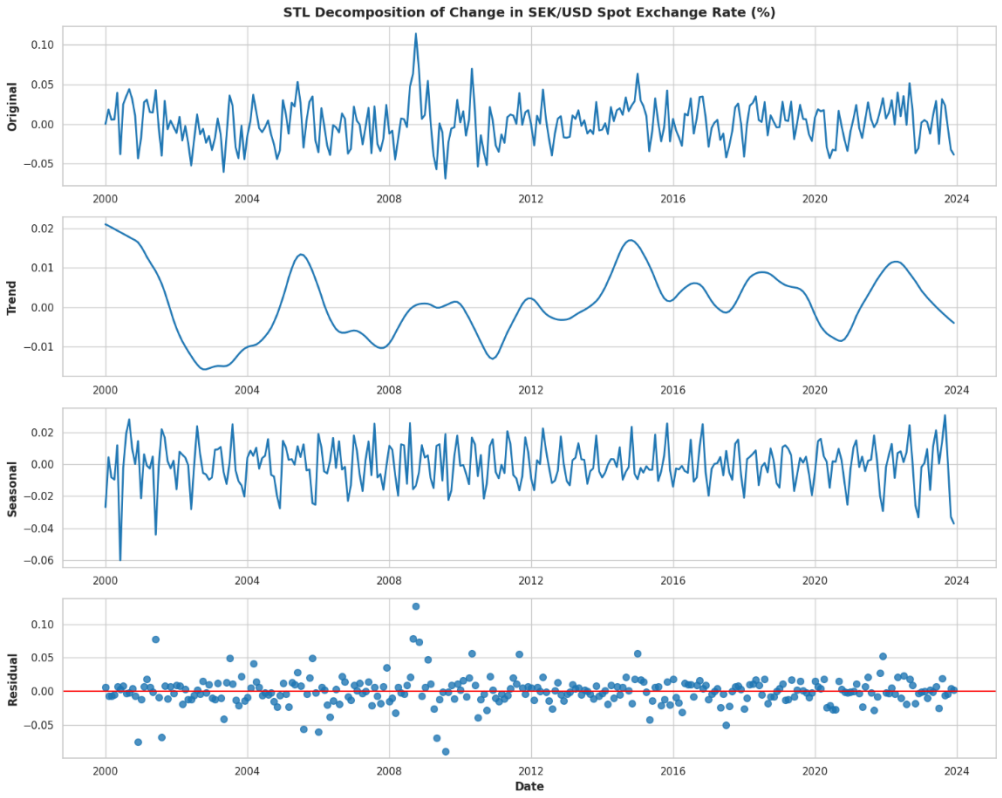


**Figure A.3** STL Decomposition of Change in SEK/USD Spot Exchange Rate

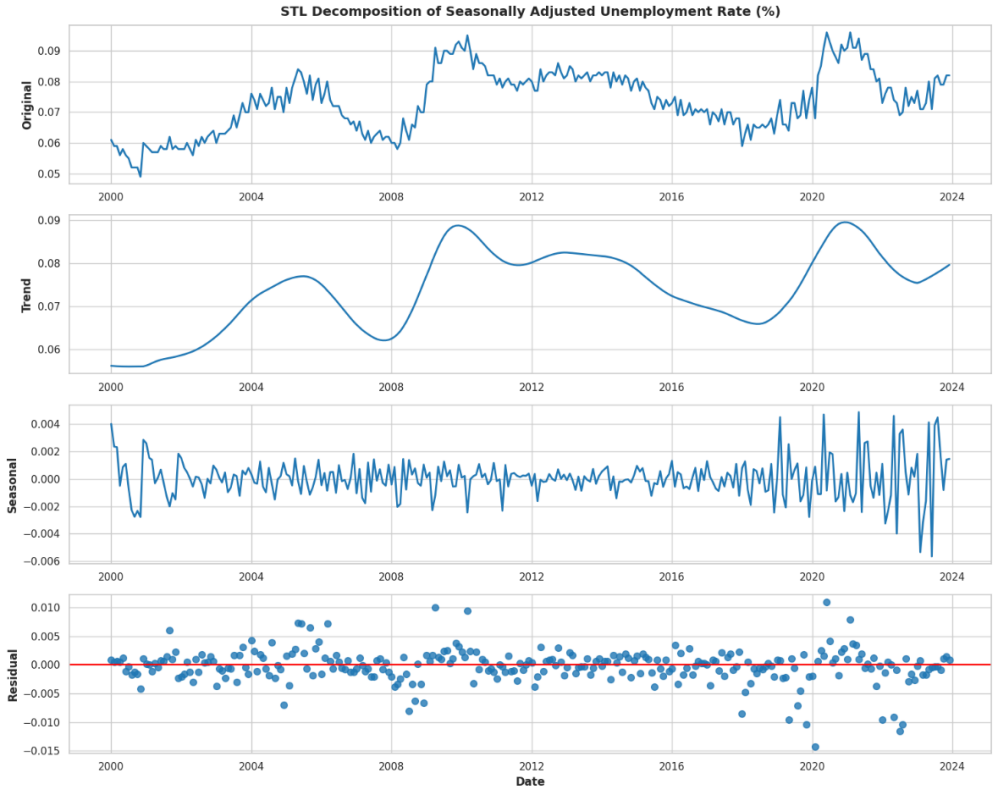


**Figure A.4** STL Decomposition of Seasonally Adjusted Unemployment Rate

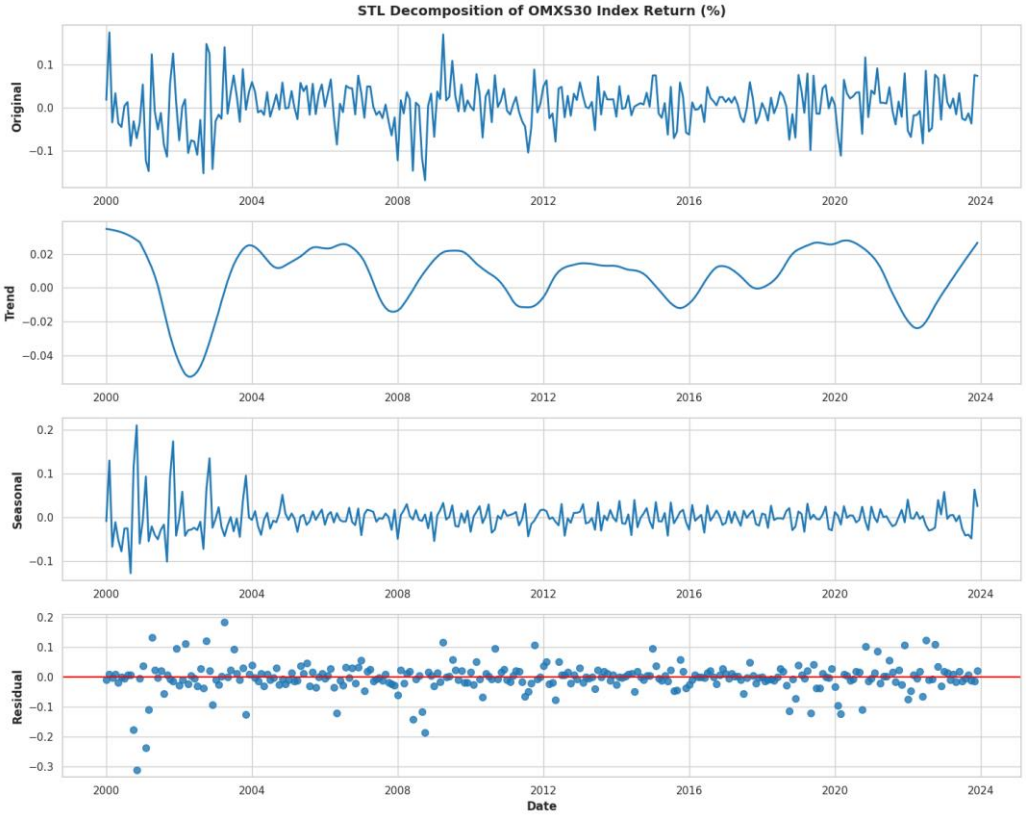


**Figure A.5** STL Decomposition of OMXS30 Index Return

## Appendix B: Correlation Matrix and ACF/PACF Graphs

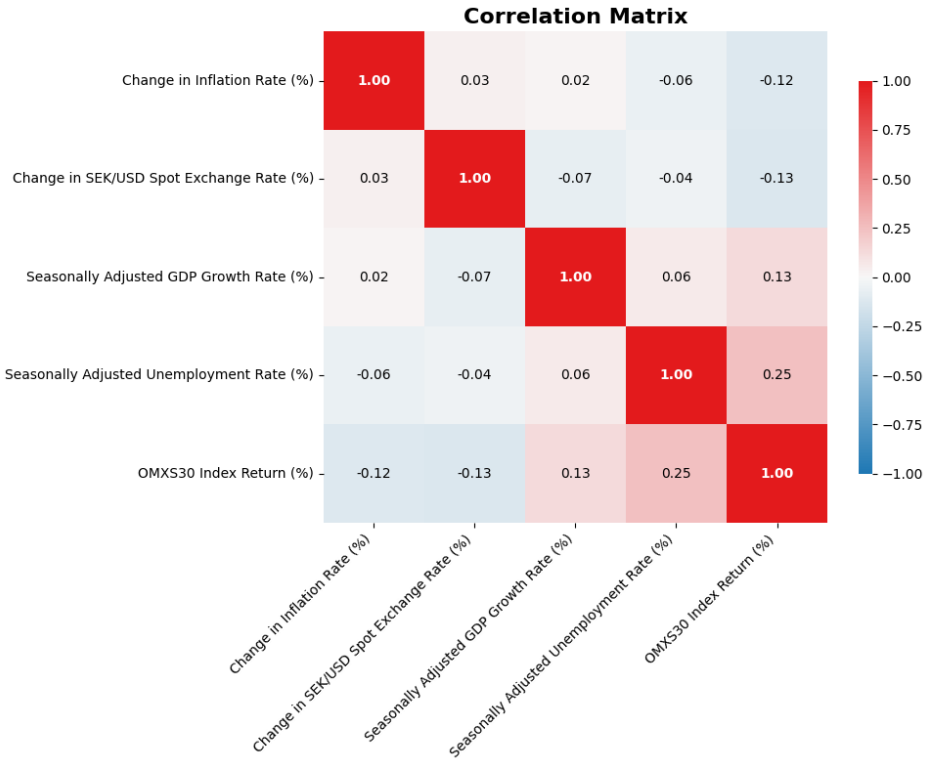


**Figure A.6** Correlation Matrix of Macroeconomic Variables and OMXS30 Index Returns

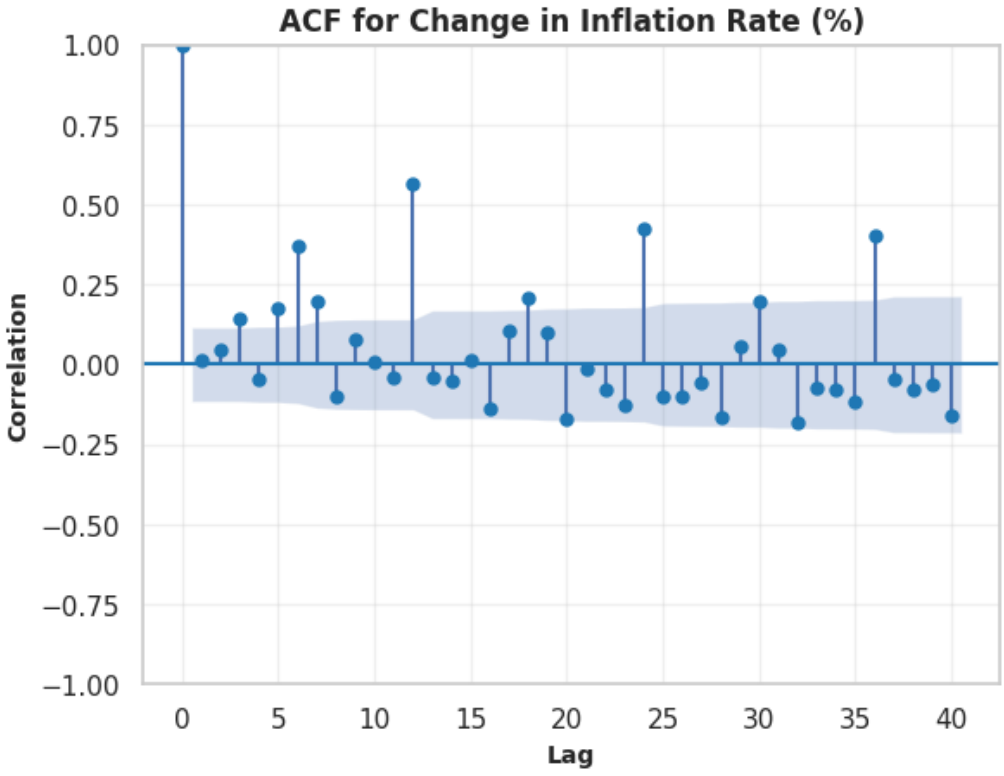


**Figure A.7** ACF for Change in Inflation Rate

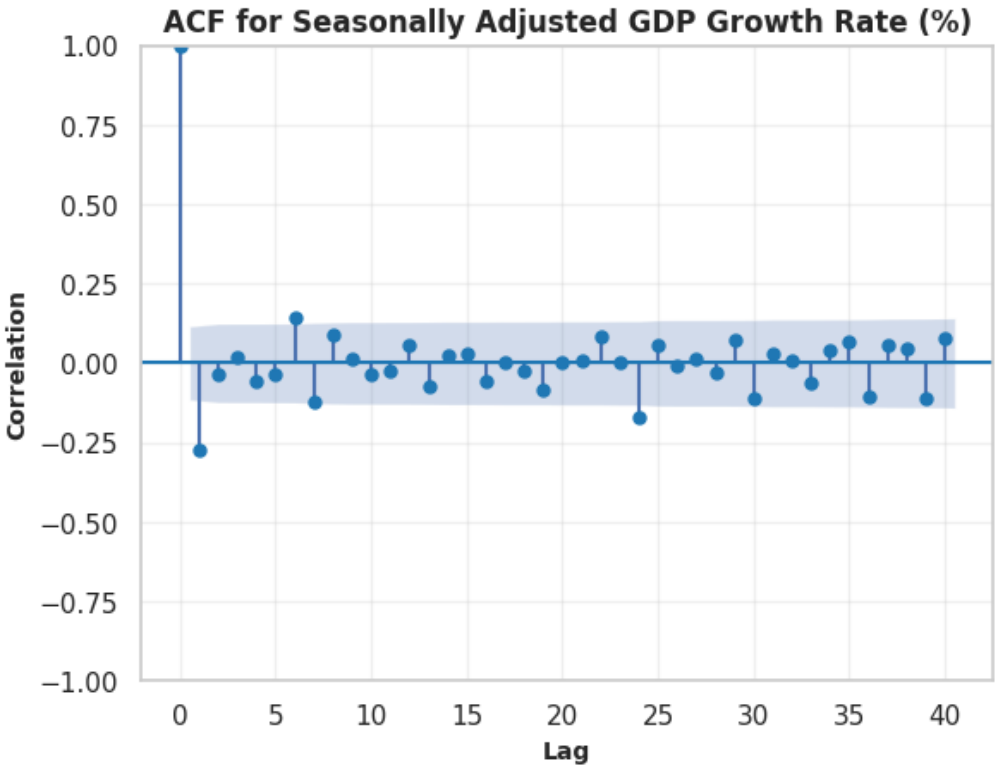


**Figure A.8** ACF for Seasonally Adjusted GDP Growth Rate

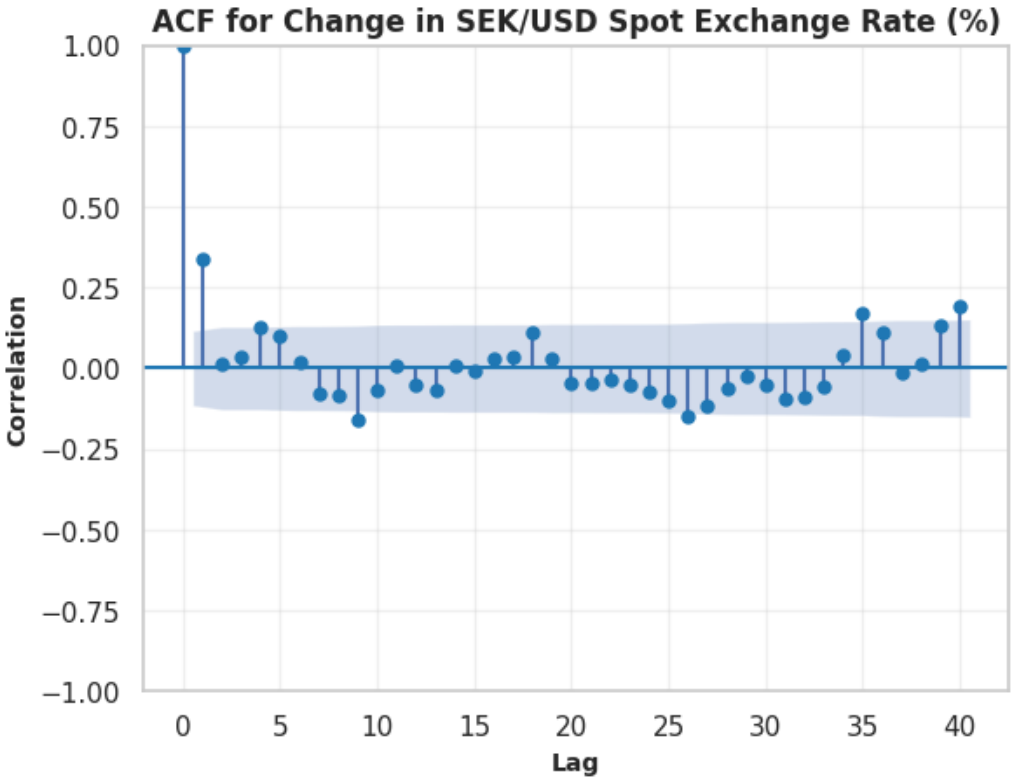


**Figure A.9** ACF for Change in SEK/USD Spot Exchange Rate

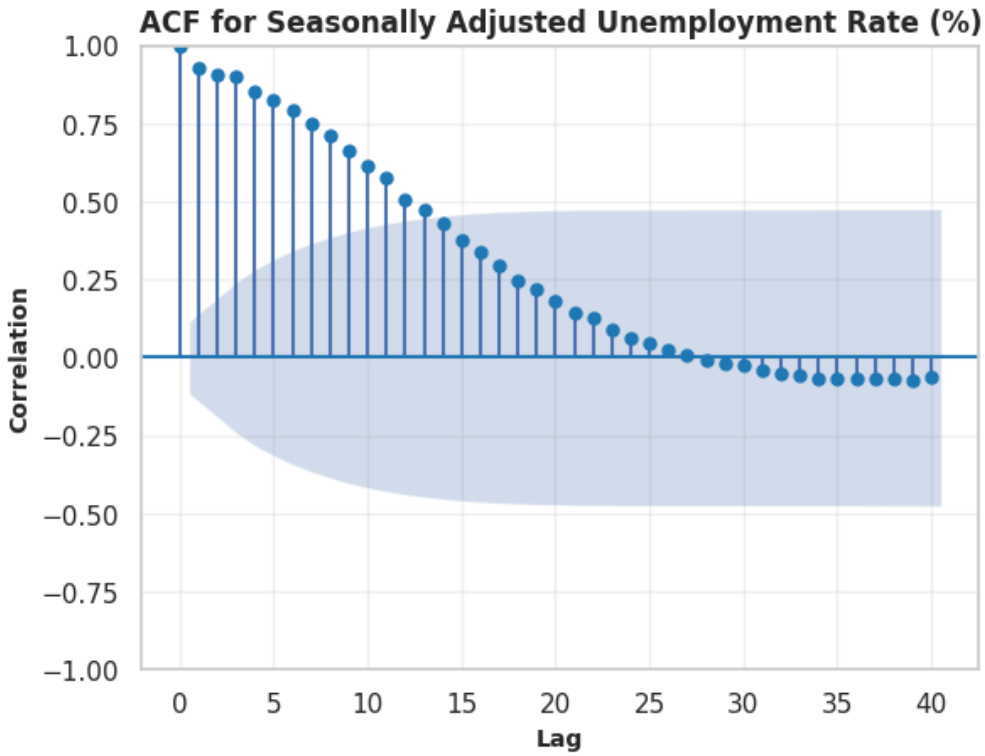


**Figure A.10** ACF for SA Unemployment Rate

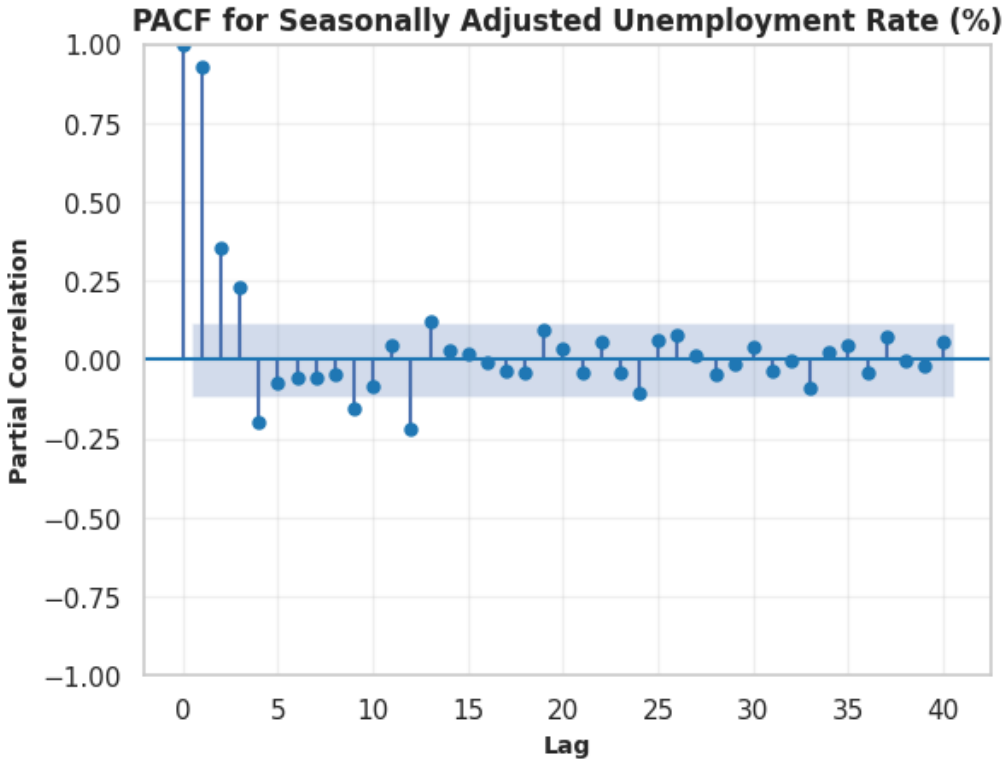


**Figure A.11** PACF for SA Unemployment Rate

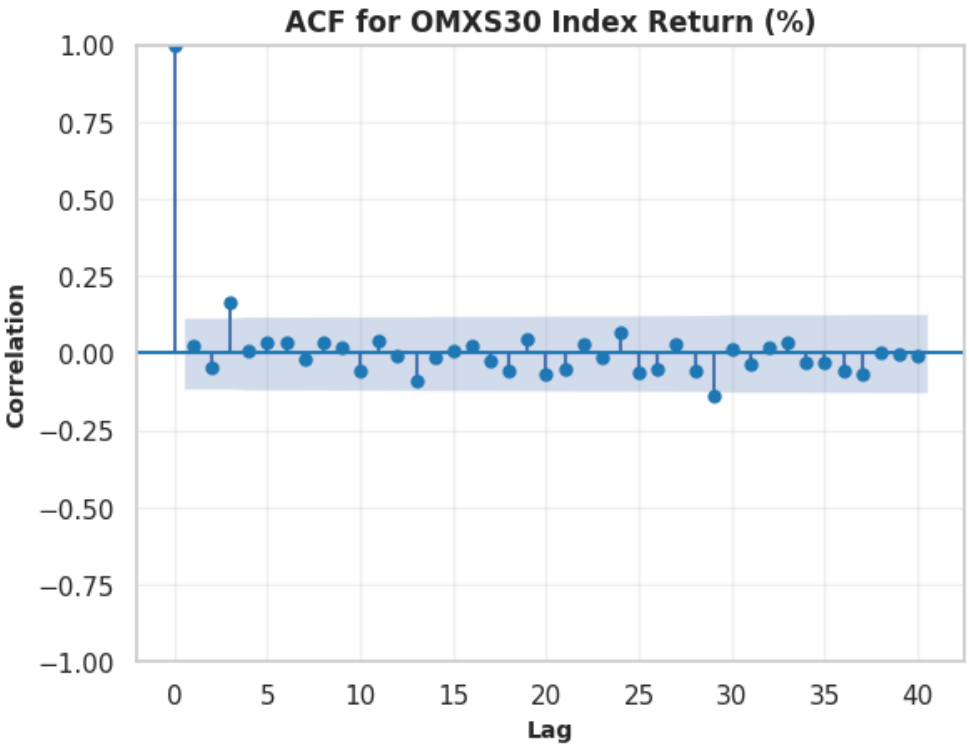


**Figure A.12** ACF for OMXS30 Index Return

## Appendix C: Model Performance Metrics and Comparative Visualizations

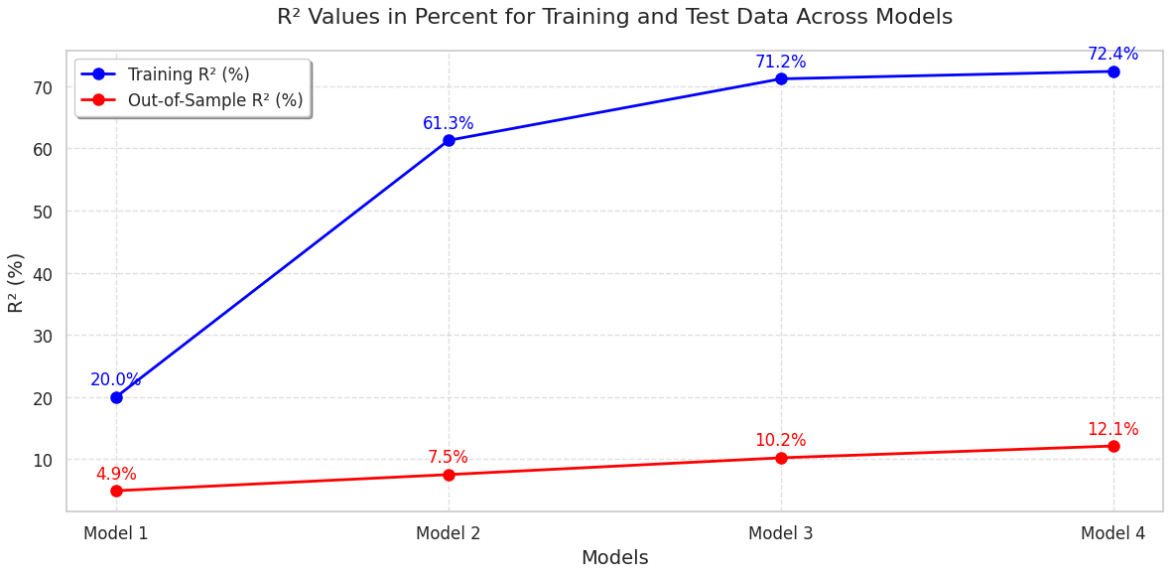


**Figure A.13** R² Values for Training and Out-of-Sample Data Across Four Models

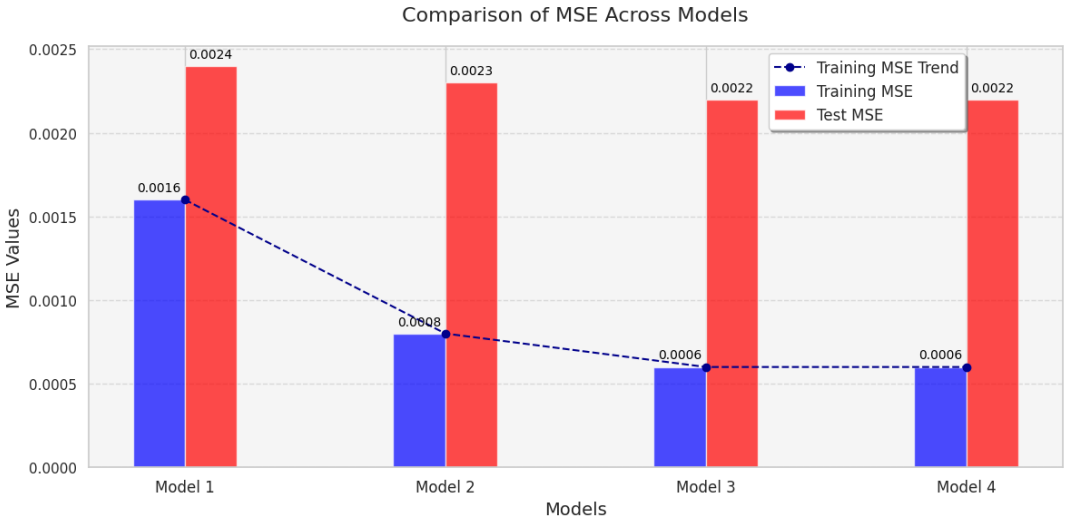


**Figure A.14** Comparison of Training and Test MSE Across Four Models

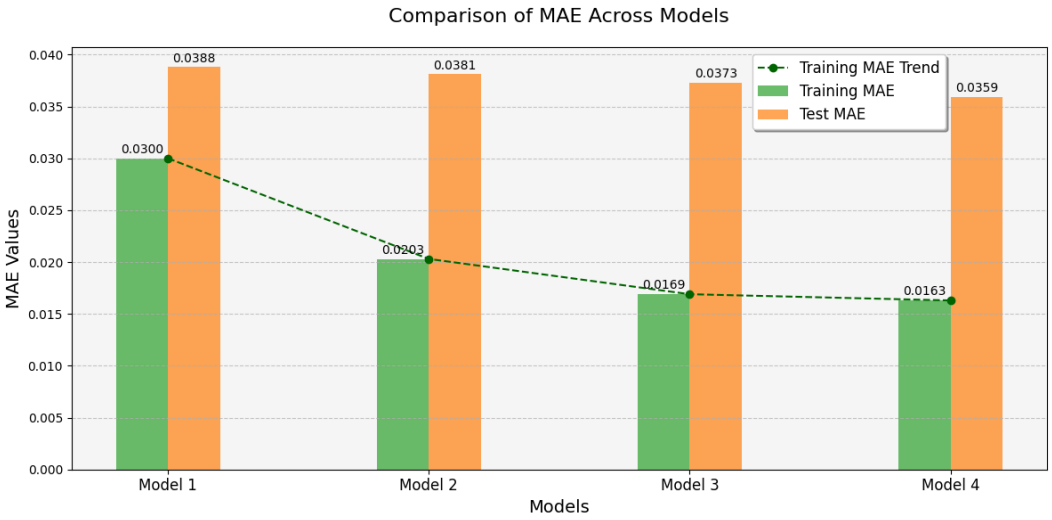


**Figure A.15** Comparison of Training and Test MAE Across Four Models

**Table A.2** Performance Metrics Comparison Across Four Models

| Model | Training $R^2$ (%) | $R^2_{OS}$ (%) | Training MSE | Test MSE | Training MAE | Test MAE |
|---|---|---|---|---|---|---|
| Model 1 | 20.0 | 4.9 | 0.0016 | 0.0024 | 0.0300 | 0.0388 |
| Model 2 | 61.3 | 7.5 | 0.0008 | 0.0023 | 0.0203 | 0.0381 |
| Model 3 | 71.2 | 10.2 | **0.0006** | **0.0022** | 0.0169 | 0.0373 |
| Model 4 | **72.4** | **12.1** | **0.0006** | **0.0022** | **0.0163** | **0.0359** |