KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Technology

First cycle, 15 credits

How do different machine learning models compare in their ability to predict short-term movements in the S&P 500 index?

MAHAD AHMED
ISMAIL MOHAMMED

Stockholm, Sweden, 2024

How do different machine learning models compare in their ability to predict short-term movements in the S&P 500 index?

MAHAD AHMED

ISMAIL MOHAMMED

Degree Programme in Computer Engineering
Date: June 5, 2024

Supervisor: Zhenyu Li
Examiner: Håkan Olsson
School of Electrical Engineering and Computer Science
Swedish title: Hur jämför sig olika maskininlärningsmodeller när det gäller deras förmåga att förutsäga kortsiktiga rörelser i S&P 500-indexet?

Abstract

This thesis explores the application of machine learning techniques within the financial technology sector, specifically targeting the prediction of the S&P 500 index. The S&P 500 is a key indicator of the US economy which reflects the stock performance of 500 major companies. Traditional statistical methods for predicting stock market trends often struggle with the volatility and complexity of financial markets. Our research aims to fill this gap by comparing the effectiveness of various machine learning models which are Polynomial Regression, Decision Trees, and Long Short-Term Memory (LSTM) networks to predict short-term movements of the S&P 500 index.

The significance of this problem lies in the growing reliance on ML models by financial companies to inform investment strategies. The study uses historical S&P 500 data from 1995 to 2024, retrieved from Yahoo Finance, focusing on metrics such as Open, High, Low, Close, and Volume prices. The data was preprocessed to ensure consistency and was divided into training and testing sets to evaluate the models.

Our results show that the Polynomial Regression and Decision Tree models performed well on the training data but they had significant errors on the testing data. While, the LSTM model showed better performance, effectively capturing both short-term and long-term market trends. Based on these findings results, suggest that LSTM networks provide the most reliable predictions.

Keywords

S&P 500, ML-models, LSTM, Polynomial Regression, Decision Trees

Sammanfattning

Denna avhandling utforskar användningen av maskininlärningsmodeller (ML-modeller) inom finansiell teknik, med särskilt fokus på att förutsäga S&P 500 index. S&P 500 är en viktig indikator på USA:s ekonomi som återspeglar aktieprestationen för 500 stora företag. Traditionella statistiska metoder för att förutsäga aktiemarknadstrender kämpar ofta med volatiliteten och komplexiteten på finansmarknaderna. Vår forskning syftar till att fylla denna lucka genom att jämföra effektiviteten hos olika ML-modeller, nämligen polynomregression, beslutsträd och Long Short-Term Memory, för att förutsäga kortsiktiga rörelser i S&P 500-indexet.

Betydelsen av detta problem ligger i den ökande tilliten till ML-modeller av finansiella företag för att informera investeringsstrategier. Studien använder historiska S&P 500-data från 1995 till 2024, hämtade från Yahoo Finance, med fokus på mätvärden som öppning, högsta, lägsta, stängning och volympriser. Data förbehandlades för att säkerställa konsistens och delades in i tränings- och testuppsättningar för att utvärdera modellerna.

Våra resultat visar att polynomregression och beslutsträdsmodeller presterade väl på träningsdatan men hade betydande fel på testdatan. Däremot visade LSTM-modellen bättre prestanda genom att effektivt fånga både kortsiktiga och långsiktiga marknadstrender. Baserat på dessa resultat indikerar studien att LSTM ger de mest tillförlitliga förutsägelserna.

Nyckelord

S&P 500, ML-modeller, LSTM, Polynomregression, Beslutsträd

Acknowledgments

Alhamdulillah, All Praise And Gratitude Belongs To Allah.

We want to thank our supervisor Zhenyu Li for his excellent work of guiding us through this thesis. We also want to thank our examiner Håkan Olsson for his support.

Stockholm, June 2024
Mahad Ahmed Ismail Mohammed

Innehåll

1 Introduction **1**
1.1 Background . . . 1
1.2 Purpose . . . 2
1.3 Goals . . . 2
1.4 Delimitations . . . 2
1.5 Thesis outline . . . 2

2 Technical Background **4**
2.1 S&P 500 index . . . 4
2.2 Data . . . 4
2.3 ML-models . . . 7
2.3.1 Polynomial Regression . . . 7
2.3.2 Decision Tree . . . 7
2.3.3 Neural networks . . . 9
2.3.3.1 Recurrent neural networks . . . 10
2.3.3.2 Long Short-Term Memory . . . 11
2.4 Evaluation Metrics . . . 12
2.4.1 Mean Absolute Error (MAE) . . . 12
2.4.2 Root Mean Squared Error (RMSE) . . . 12
2.4.3 Mean Absolute Percentage Error (MAPE) . . . 12

3 Methods **15**
3.1 Methodology . . . 15
3.1.1 Framework and Libraries . . . 15
3.2 Data Collection . . . 16
3.2.1 Data processing . . . 16
3.2.2 Feature Preparation . . . 17
3.2.3 Normalization and Sequence Creation . . . 17
3.3 Model Implementation & Performance Evaluation . . . 18

3.3.1 Determination of the Optimal Epoch for LSTM 18

4 Results & Analysis **21**
4.1 Polynomial Regression 21
4.1.1 Model Training 21
4.1.2 Prediction Results 22
4.1.3 Analysis Causes Of Prediction Errors 25
4.2 Decision Tree 25
4.2.1 Prediction Results 25
4.2.2 Analysis of limitations of Decision Tree 28
4.3 LSTM 29
4.3.1 Model Training 29
4.3.2 Prediction Results 30
4.3.3 Analysis of LSTM Model Performance 32

5 Discussion **33**
5.1 Model Comparison 33
5.1.1 Visual Analysis of Predictions 33
5.1.2 Detailed Analysis of Model Performance 39
5.1.3 Strengths and Weaknesses 40

6 Conclusions **43**

References **45**

Figurer

2.1 S&P 500 historical data from 1995 to 2024 before processing. 6
2.2 Heart attack risk assessment decision tree [8] 8
2.3 Neural Networks architecture [9]. 9
2.4 (RNN architecture[10]) 10
2.5 An LSTM unit's structure [11] 11

3.1 Optimal Epoch Value 19

4.1 Actual Closing Prices vs. Predicted Prices by the Polynomial Regression Model 22
4.2 Actual Closing Prices vs. Predicted Prices by the Decision Tree Model 26
4.3 Actual Close Prices vs. Predictions by the Decision Tree Model 29
4.4 Actual Closing Prices vs. Predicted Prices by the LSTM Model 30

5.1 Actual Close versus Model Predictions. This plot shows the performance of Polynomial Regression, Decision Tree, and LSTM models in predicting the S&P 500 closing prices over the years. 33
5.2 Close Price Predictions Comparison during the Training Phase (2004-2007). This detailed view highlights how each model aligns with the actual market trends during a stable period, emphasizing the nuances of each model's prediction capabilities. 34
5.3 Test Phase Model Predictions (2018 Onwards). This graph illustrates the performance of each predictive model against the actual closing prices of the S&P 500 during the test phase, covering the most recent data from 2018 onwards. 35
5.4 MAE 37
5.5 RMSE 38
5.6 MAPE 39

Tabeller

2.1 S&P 500 historical data sample from 1995 to 2024 before processing. 5

4.1 Snippet of training results showing actual vs. predicted closing prices 23
4.2 Snippet of testing results showing actual vs. predicted closing prices 24
4.3 Performance Metrics for Polynomial Regression Model 24
4.4 Snippet of training results showing actual vs. predicted closing prices by the Decision Tree model 27
4.5 Snippet of testing results showing actual vs. predicted closing prices by the Decision Tree model 27
4.6 Performance Metrics for Decision Tree Model 28
4.7 Snippet of training results showing actual vs. predicted closing prices by the LSTM model 31
4.8 Snippet of testing results showing actual vs. predicted closing prices by the LSTM model 31
4.9 Performance Metrics for LSTM Model 31

5.1 Comparison of actual and predicted closing prices by Polynomial Regression, Decision Tree, and LSTM models . . 36

Chapter 1

Introduction

1.1 Background

This project lies at the intersection of financial technology (Fin-tech) and machine learning (ML), focusing on the prediction of the Standard and Poor's 500 (S&P 500) index [1]. The S&P is a stock market index that measures the stock performance of 500 large companies listed on stock exchanges in the United States. The index includes companies from all sectors of the economy, making it a significant indicator of the performance and condition of the United States economy.

Fin-tech is increasingly applying ML-models to predict stock market trends instead of traditional forecasting methods. This makes the topic of ML-models highly relevant in today's world[2]. The research done in this space of ML-models with financial market analysis has transformed the investment strategies of the biggest financial companies in the world [3]. The ease of use of the ML-models and their way of giving reliable output from a vast amount of data has led to their widespread adoption.

Traditional methods of statistical analysis forecasting the volatile stock market fall short due to their inability to process large amounts of data or adapt quickly. This is because financial markets are inherently volatile and influenced by different factors ranging from wars to weather phenomena. ML-models offer a way that enables models to learn from current market trends and improve over time, thereby providing reliable and more accurate predictions.

1.2 Purpose

We think that the effectiveness or rather the performance of different ML-models in volatile financial markets has not been compared comprehensively, in particular for short-term predictions of indices like the S&P 500. This thesis aims to give a comprehensive comparison and analysis in evaluating these ML-models. Through this work, we seek to identify the model that most accurately forecasts the movements of a volatile index, thereby providing insights to financial analysts which they can use to be more informed and reinvent their investment strategies. This project aims to evaluate these models' effectiveness and how well they perform in the complexities of the volatile stock market. By enhancing the understanding of these models' predictive power, the study seeks to contribute to the optimization of investment strategies and risk management, benefiting the broader financial community and potentially influencing future research directions in economic forecasting.

1.3 Goals

The primary goal of our thesis, is to learn more about the finance technology domain, focusing on the use of ML techniques for stock market prediction. Our aim is to critically assess and compare the efficacy of various ML models, such as Long Short-Term Memory, decision trees and polynomial regression, in how they accurately forecast the fluctuations within the S&P 500.

1.4 Delimitations

The study is limited to predicting short-term movements of the S&P 500 index using publicly available data. For the purpose of this research, "short-term movement" refers to daily changes in market prices. We will not consider private data or detailed individual stock movements within the index.

Additionally given the time we had, the analysis will limit the index data to the open, close, low, high and volume, excluding other potentially influential data like news sentiment and dividend yields.

1.5 Thesis outline

In the subsequent Chapter 2 *Technical Background*, we will delve into over the technical background necessary to understand the results and conclusions.

Furthermore, this chapter will also discuss the data from the S&P 500 and the different employed ML-models. Chapter 3 *Method*, will detail process of data collection and cleaning. It will go in detail about how the data was then used to train our different models and how it gave us the results.

In Chapter 4 *Results and Analysis*, we will present our results supplemented with figures clarifying these outcomes for the reader with an analysis. In the last chapters *Discussion* 5 and *Conclusions* 6, a discussion of the comparison of the different ML-models and a conclusion will follow.

Chapter 2

Technical Background

2.1 S&P 500 index

The Standard & Poor's 500, or S&P 500, is a market-capitalization-weighted index of 500 of the largest publicly traded companies in the U.S, and is one of the most widely followed equity indices, representing about 80% of available market capitalization[4]. First introduced in 1957, it serves as a critical barometer of the overall U.S. equity market's health and a benchmark for the performance of various investment assets. The index's value is calculated by summing the adjusted market caps of its components, making it sensitive to changes in stock price and the number of shares outstanding. Given its prominence, the S&P 500's movements are influenced by a multitude of factors including economic indicators, geopolitical events, and market sentiment, leading to its known volatility.

2.2 Data

The dataset for this study comprises historical data of the S&P 500 index retrieved from Yahoo Finance[5], covering daily market activities over a period from January, 1995, to March, 2024. This dataset specifically includes the open price, high price, low price, close price and the adjusted close price of the trading day. The Open price indicates the value of the S&P 500 index at the beginning of the trading day, while the Close price reflects the index's value at the market close, serving as a critical benchmark for daily market performance. The High price represents the highest value that the index reached during the day, while the lowest value that the index reached within the same day. The Adjusted Close (AdjClose) price is the closing price adjusted for both

dividends and stock splits. The Volume represents the total number of shares traded during the day and is indicative of the market activity and liquidity. For ease of writing, in the rest content of this thesis we use terms *Open*, *High*, *Low*, *Close*, *AdjClose* to indicate the open price, high price, low price, close price, and the adjusted close price respectively. These elements are pivotal for analyzing market trends and will be used to train various machine learning models to predict short-term movements of the index. The data will undergo preprocessing to ensure consistency and reliability, including handling missing values and anomalies typically associated with non-trading days such as weekends and public holidays. A sample of the S&P 500 index is given in Table 2.1. As an example, Figure 2.1 illustrates the close price from 1995 to 2024. From which we can see that the close price is varying in a small scale in a short sequential period of time, and shows a rising trend in the long run. This brings challenges to accurate predicting, thus it is selected to be the target to evaluate the performance of different predicting methods.

Index	Date	Open	High	Low	Close	Adj Close	Volume
0	1995-01-03	459.2099	459.2699	457.2000	459.1099	459.1099	262450000
1	1995-01-04	459.1300	460.7200	457.5599	460.7099	460.7099	319510000
2	1995-01-05	460.7300	461.2999	459.7500	460.3399	460.3399	309050000
3	1995-01-06	460.3800	462.4899	459.4700	460.6799	460.6799	308070000
4	1995-01-09	460.6700	461.7699	459.7399	460.8299	460.8299	278790000
...	...	...	...	...	...	...	...
7336	2024-02-23	5100.9199	5111.0600	5081.4599	5088.7998	5088.7998	3627290000
7337	2024-02-26	5093.0000	5097.6601	5068.9101	5069.5297	5069.5297	3683930000
7338	2024-02-27	5074.6000	5080.6899	5057.2093	5078.1801	5078.1801	3925950000
7339	2024-02-28	5067.2001	5077.3701	5058.3099	5069.7597	5069.7597	3789370000
7340	2024-02-29	5085.3986	5104.9023	5061.8901	5096.2700	5096.2700	5219740000

Table 2.1: S&P 500 historical data sample from 1995 to 2024 before processing.

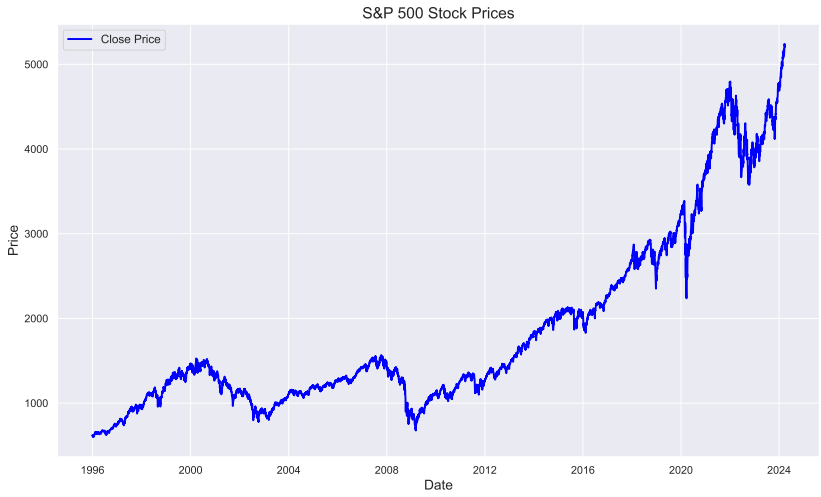

Figure 2.1: S&P 500 historical data from 1995 to 2024 before processing.

2.3 ML-models

2.3.1 Polynomial Regression

Polynomial regression [6] extends linear regression by modeling the relationship between the independent variable x and the dependent variable y as a polynomial of degree n. In regression analysis, we seek to find the functional form that best represents the relationship between variables in a dataset. For polynomial regression, this functional form is a polynomial equation:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n + \epsilon \tag{2.1}$$

where y is the dependent variable we aim to predict, x is the independent variable, and $\beta_0, \beta_1, \ldots, \beta_n$ are the model coefficients that are determined through the regression process. The term ϵ accounts for the error in the predictions.

Identifying the most suitable coefficients β_i for our model involves an estimation technique known as the least squares method. This technique seeks to minimize the sum of the squared residuals, which are the differences between the observed values and those estimated by the model.

Parameter estimation via the least squares method involves solving a system of equations that derive from setting the partial derivatives of the sum of squared residuals with respect to each coefficient to zero. This approach ensures that the fitted model minimizes the total squared error between the predicted and observed values.

2.3.2 Decision Tree

Decision Trees are a robust modeling tool for classification and regression tasks in financial analytics. This makes them particularly useful in handling complex decision-making processes, such as that for predicting stock price movements, based on various economic indicators.

A Decision Tree model creates a binary tree from the data, where decision nodes split the dataset according to specific feature thresholds until it comes to a conclusion at the leaf nodes. This simple structure makes it easier to interpret, since it's possible to trace the decision paths visually, which is very valuable in strategy formulation. The ability of Decision Trees to model non-linear relationships with minimal preprocessing makes them robust for modeling and prediction in financial market trends [7] A Decision Tree works

by creating a binary tree from the dataset. Each recursive partitioning of the tree is based on the attribute value that most separates the subsets from the data. It measures the information gain obtained by such separation.

Decision Trees are constructed by a recursive process that works as follows: it selects the attribute that separates the data most from the set of attributes, then splits the data set according to the values of that attribute. Each new data subset is further divided until every subset at a node has the same value for the target variable or until there are no more remaining attributes to distinguish samples. Finally, there's the important pruning step in refining the Decision Tree. This involves pruning the tree; that is, eliminating parts of the tree that could be based on noise or outliers. This improves the predictive power of the model and can be used to avoid overfitting. Pruning enhances the model's generalization ability by removing less important branches.

Decision Tree Example

The decision tree in Figure 2.2 illustrates the risk assessment model for heart attack prevention based on age and smoking status.

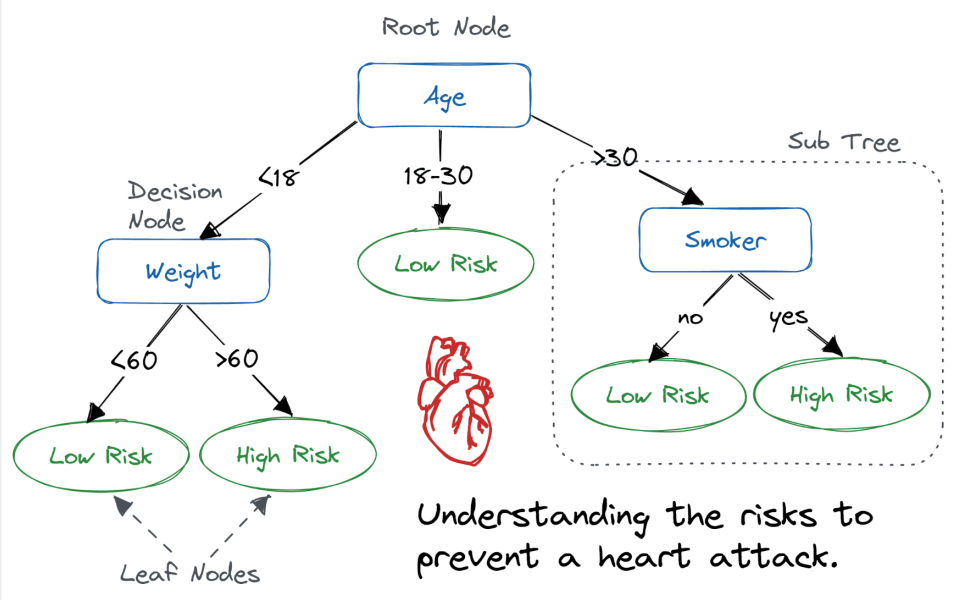

Figure 2.2: Heart attack risk assessment decision tree [8]

2.3.3 Neural networks

Neural networks (NN) are a vital part of machine learning, inspired by our brains' neural function. The core of NN consists of layers of interconnected nodes or another term for it is *neurons*, each capable of performing simple computations. When linked, these neurons go through a series of transformations.

The NN architecture typically includes an input layer, several hidden layers, and an output layer. The data flows through the network from the input layer, which receives the initial data, to the hidden layers, where the processing happens. In these layers, each neuron is a function of a weighted sum of its inputs. *Weights* are adjustable parameters that determine the strength of the connection between two neurons—how much influence one neuron's output will have on another.

After processing through the hidden layers, the data reaches the output layer, which yields the network's final prediction or classification. The non-linear activation functions applied by neurons in the hidden layers enable NNs to capture complex patterns in data

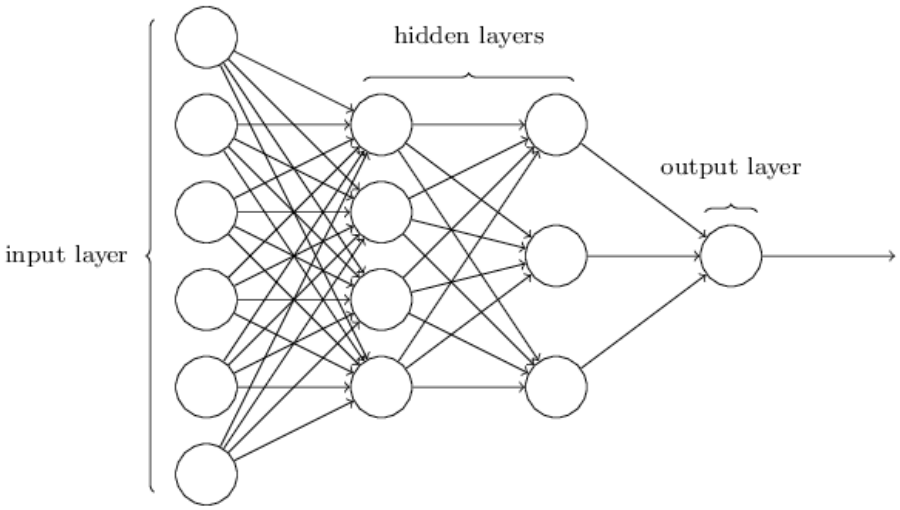

Figure 2.3: Neural Networks architecture [9].

2.3.3.1 Recurrent neural networks

Recurrent neural networks (RNN) are an extension of NNs, where it expands the capabilities of standard NNs by incorporating memory into the architecture, which allows the network to retain information from previous input.

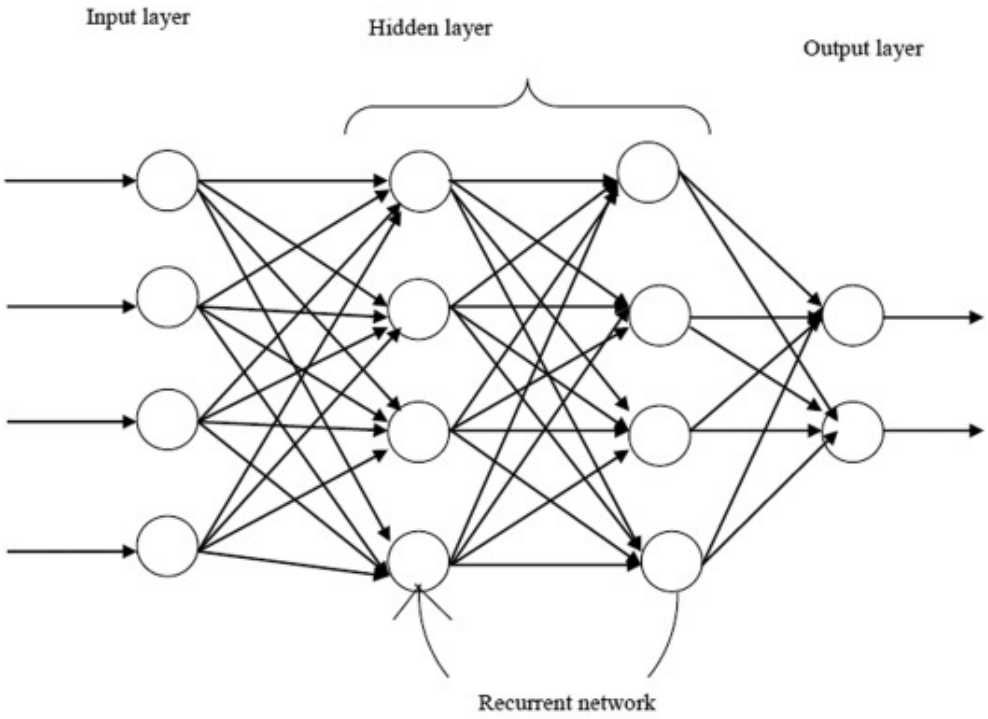

Figure 2.4: (RNN architecture[10])

2.3.3.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) [1] networks are an advanced form of RNNs designed to overcome the challenges associated with traditional RNNs, such as the vanishing and exploding gradient problems. LSTMs are particularly adept at processing sequential data, making them ideal for time-series analysis pertinent to financial markets like the S&P 500 index. LSTMs enhance the basic RNN structure through the integration of complex mechanisms called gates, which control the flow of information by retaining or discarding data based on its relevance, determined through trainable parameters. The primary components of an LSTM cell include:

- *Input gate*: Controls the extent to which a new value flows into the cell.
- *Forget gate*: Determines the parts of the cell state to be thrown away.
- *Output gate*: Decides what next to output from the cell state.

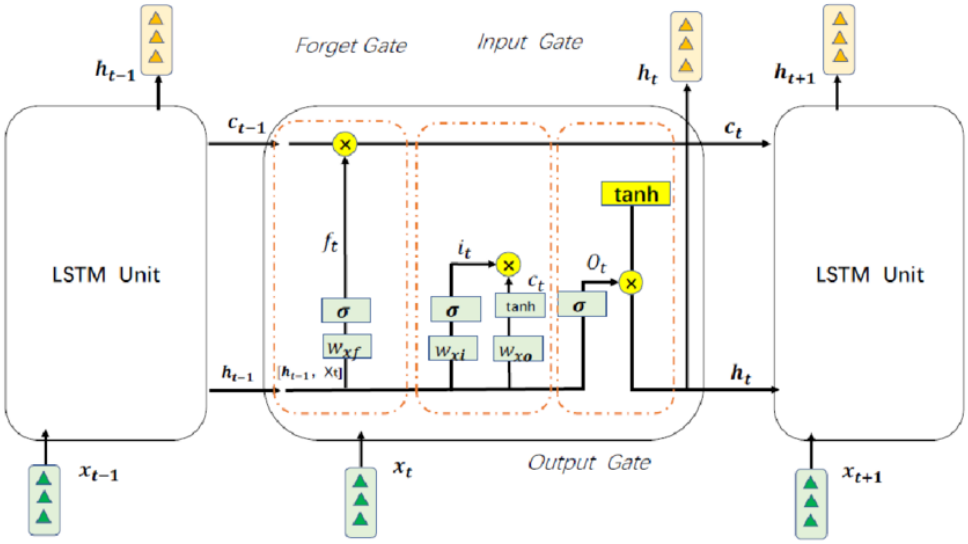

Figure 2.5: An LSTM unit's structure [11]

These capabilities make LSTMs especially valuable in financial modeling for their ability to remember both long and short-term trends and dependencies. This capability allows them to predict future market movements by analyzing patterns over extended periods, critical for the volatility and non-linear nature of financial data like stock prices. LSTMs' ability to effectively model and predict complex patterns in time-series data makes them a top

choice for financial analysts who need to forecast market trends with a high degree of accuracy. Their comprehensive learning and prediction capabilities significantly contribute to enhanced decision-making in financial contexts.

2.4 Evaluation Metrics

2.4.1 Mean Absolute Error (MAE)

MAE is a standard metric that calculates the average magnitude of the errors in a set of predictions, without considering their direction [12]. It is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.2}$$

where y_i represents the true values, $\hat{y}_i$ denotes the predicted values, and n is the number of observations. The MAE is particularly useful because it provides a straightforward indication of predictive accuracy in the same units as the response variable.

2.4.2 Root Mean Squared Error (RMSE)

RMSE is a widely used metric that measures the square root of the average of squared differences between prediction and actual observation [13]. Its formula is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.3}$$

RMSE gives a relatively high weight to large errors, meaning that all errors are not treated equally. This is particularly useful in scenarios where large errors are especially undesirable.

2.4.3 Mean Absolute Percentage Error (MAPE)

MAPE measures the size of the error in percentage terms [14]. It is calculated as:

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{2.4}$$

MAPE is scale-independent and provides a clear representation of error as a percentage of the actual values, which can be intuitive for understanding model performance. MAPE provides a mathematical method to assess the precision of our predictive models.

Chapter 3

Methods

3.1 Methodology

First, we will collect historical data on the S&P 500 index. This involves retrieving data from reliable financial sources such as Yahoo Finance. The next step is to clean and preprocess the data. This includes handling missing values, removing duplicates, and transforming the data into a suitable format for analysis.

After preparing the data, we will implement various machine learning models, including Polynomial Regression, Decision Trees, and LSTM. We will then evaluate the performance of these models using specific metrics such as MAE, RMSE, and MAPE.

Finally, we will analyze the results and draw conclusions on the effectiveness of the different models in predicting the S&P 500 index.

3.1.1 Framework and Libraries

We developed and evaluated our machine learning models using Python, supported by its robust data science libraries. Jupyter Notebook, our primary development environment, enabled interactive coding, data manipulation, and visualization—ideal for exploratory data analysis and model prototyping. For data preprocessing, we used Pandas, which offers powerful data handling capabilities [15]. Model implementation and testing were conducted with Scikit-learn (sklearn) [16], which provides comprehensive tools for data mining and is compatible with Python's numerical libraries like NumPy and SciPy.

The LSTM model was built and trained using Keras [17], which is an open-

source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Keras is user-friendly, modular, and extensible, which facilitates the creation and training of deep learning models.

3.2 Data Collection

Data for this study was sourced from Yahoo Finance [5], as seen in table 2.1 which provides an extensive archive of historical market data. This platform was selected due to its robust dataset offerings and the ability to customize the range of data retrieval for free. For the purpose of obtaining a comprehensive dataset sufficient for training our predictive models, the time span from 1995 to 2024 was chosen as it shown in this figure 2.1. This specific range was selected for two primary reasons:

- **Data Quality and Relevance:** The data before the year of 1995 was observed to be relatively constant, which does not contribute meaningful variations necessary for training efficient predictive models. Including such data could lead to over-fitting and poor generalization in model performance.
- **Time Efficiency**: Training models like LSTM on large datasets requires significant computational power. The chosen time-frame provides enough data to capture market trends while ensuring that training is manageable and not excessively time-consuming. Extending the range further would increase the time and resources needed for training.

This duration ensures a substantial volume of data points, encompassing various market conditions and trends that are critical for the robustness of the study.

3.2.1 Data processing

After collecting the historical S&P 500 index data, a thorough data preparation phase was initiated to ensure its suitability for predictive modeling. Initially, the raw data was imported into a Python environment using the Pandas library, which offers robust functionalities for handling datasets. This phase was structured as follows:

- **Data Cleaning:** The integrity of the dataset was handled by identifying and addressing if there are any missing values or duplicates, and any irregular entries. Accounting for non-trading days such as weekends and public holidays was essential to prevent distortions in the time series analysis.

- **Column Removal:** Certain columns, like 'Adj Close', were removed from the dataset as they were not needed for the analysis. This step minimizes the dataset, focusing on the most relevant features that would potentially influence the predictions of market trends.

3.2.2 Feature Preparation

Feature Preparation is an important step in the data processing stage that directly influences the performance of machine learning models. The approach to feature preparation varied depending on how the specific models use the data to train.

For models such as Polynomial Regression and Decision Tree, the features 'Open', 'High', 'Low', and 'Volume' were selected based on their potential impact on the 'Close' price, which was used as the target variable. This selection feature was helpful for Polynomial Regression and Decision Tree to predict the stock prices effectively.

3.2.3 Normalization and Sequence Creation

On the other hand, the LSTM model does not necessarily need features such as Polynomial Regression and Decision Tree. Instead, the LSTM model requires the input data to be transformed into a sequence of values that it can process over time. To prepare the data for LSTM:

- The 'Close' prices were first normalized to scale the value between '0' and '1'. This normalization helps the LSTM model to converge faster during training.

- A function was created to convert the series of prices into a supervised learning problem. For each time step, the function looks back over a set number of previous time steps (designated as 'look_back') to compile a sequence used to predict the next value.

- The resulting sequences consist of 30 previous 'Close' prices to predict the subsequent 'Close' price, providing the LSTM model with the necessary temporal context.
- The data is then reshaped into a three-dimensional array suitable for training the LSTM.

3.3 Model Implementation & Performance Evaluation

We implemented three distinct machine learning models, utilizing the built-in functionalities from the Python library, *Sklearn*, for development. Each model underwent training with the same dataset, ensuring consistency in the evaluation process.

The data was partitioned in an 80/20 split, with 80% utilized for training purposes and the remaining 20% allocated for testing the models' predictive performance. The decision to use an 80/20 split was driven by the need to maximize the amount of data available for training while still retaining a substantial, separate dataset for an unbiased evaluation of the model's effectiveness. This balance is crucial in achieving a model that not only performs well on the data on which it was trained but also maintains that performance in practical applications, thereby ensuring the model's utility in real-world tasks.

The effectiveness of the models was gauged using established metrics such as MAE, RMSE, and MAPE, allowing for a comprehensive assessment of their forecasting capabilities on the S&P 500 index.

3.3.1 Determination of the Optimal Epoch for LSTM

In neural network training, an epoch is defined as one complete pass of the entire training dataset through the model. The concept of epochs is central to adjusting the model's weights effectively through repeated exposure to the data, thereby minimizing the loss function iteratively.

The number of epochs plays a crucial role in the model's ability to learn and generalize. Too few epochs can lead to underfitting, where the model does not learn the data sufficiently, while too many epochs can lead to overfitting, where the model learns the noise and specific details of the training data at the expense of its ability to generalize.

Figure 3.1: Optimal Epoch Value

During our model training, we monitored the Mean Squared Error (MSE) as a function of both training loss and validation loss. Training loss measures the model's error on the training data, showing how well the model fits this data. Validation loss, on the other hand, evaluates the model's performance on a separate validation set, reflecting its generalization capabilities.

The analysis of these losses over various epochs revealed that while the training loss consistently decreased, indicating an improving fit on the training data, the validation loss presented a minimum value at epoch 87. This particular epoch marks the point where the model achieved the best balance between learning from the training data and generalizing to new data. Post epoch 87, the validation loss showed signs of instability and a slight increase, which typically suggests the beginning of overfitting.

Given these observations seen in 3.1, epoch 87 was selected as the optimal stopping point for training. This decision ensures that the model is sufficiently trained to recognize underlying patterns without overly fitting to the noise within the training data, thus enhancing its predictive performance on unseen data.

Choosing this specific epoch is aligned with the strategy of using early stopping in training neural networks, a technique designed to prevent overfitting by terminating the training process when the validation loss begins to deteriorate or fails to improve further. This approach not only conserves computational resources but also secures a model that is robust and reliable in making predictions in real-world settings.

Chapter 4

Results & Analysis

4.1 Polynomial Regression

The Polynomial Regression model was trained on the historical closing prices of the S&P 500 index using a set of features that included ‘Open‘, ‘High‘, ‘Low‘, and ‘Volume‘ which explains more in Section 2.2. The features were transformed into polynomial features of degree 2 to better capture the non-linear relationships inherent in financial data. This transformation includes not only the square of each individual feature (e.g., Open^2, High^2, Low^2, Volume^2) but also all the interaction terms between them (e.g., Open $\times$ High, Open $\times$ Low, etc.). This polynomial expansion helps in modeling suitable graph which a linear model would miss. For example, the interaction term Open $\times$ High can capture how periods of high ”opening prices” combined with ”high highest prices” might affect the ”closing price” differently compared to periods when both are low. By using a second-degree polynomial, the model can fit not only upward or downward trends but also any parabolic trends.

In figure 4.1, it was noted that the resulting curve displayed a pattern differing from what might be expected with a quadratic function. This observation suggests that the relationships between the input features and the target variable are more complex than can be fully captured by a simple quadratic equation.

4.1.1 Model Training

The data was divided into training and testing sets with an 80% to 20% split as explained in 3.3, respectively, without shuffling to preserve the sequential order of the time series. This partitioning resulted in a training set consisting

of the first 5872 data points (index range 0 to 5871) and a testing set comprising the subsequent 1469 data points (index range 5872 to 7340), which was used to evaluate the model's predictive performance.

After data segmentation, the model was developed using a linear regression framework adapted to handle polynomial features. This adaptation was necessary to capture the non-linear patterns observed in the financial market data. This model was chosen for its ability to efficiently handle multiple regression scenarios where relationships between predictor variables and the target variable are not strictly linear. By fitting the polynomial-transformed features to the historical closing prices, the model could leverage enhanced pattern recognition to forecast future market behaviors.

4.1.2 Prediction Results

Visual Representation of Predictions: Figure 4.1 shows the actual versus predicted closing prices by the Polynomial Regression model, providing a visual assessment of its performance across different market conditions. This graph illustrates the alignment of training predictions (represented by the orange line) closely following the actual closing prices during the training phase. While, the predictions for the testing phase (represented by the green line) show significant divergence from the actual closing prices as the model attempts to project future market behaviors.

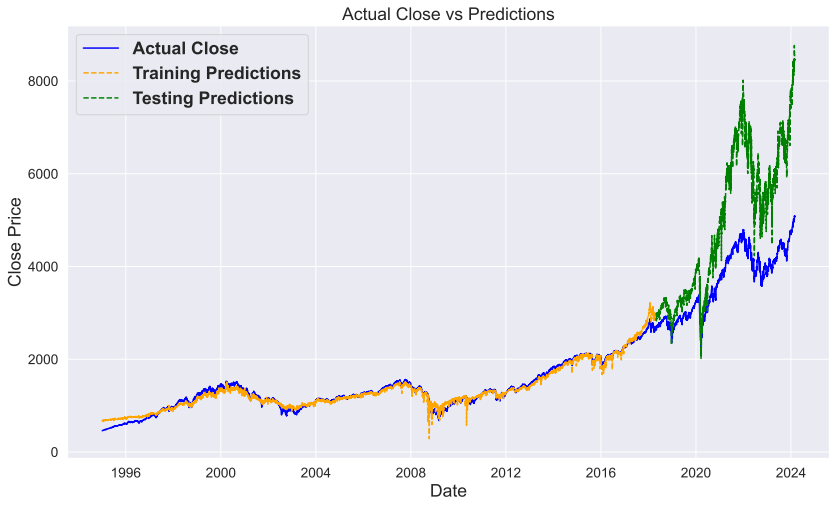

Figure 4.1: Actual Closing Prices vs. Predicted Prices by the Polynomial Regression Model

Analysis of Training Phase Results: The results in Table 4.1 illustrate the model's performance during the training phase. As shown, the predicted closing prices are reasonably close to the actual closing prices, indicating that the Polynomial Regression model has effectively learned the underlying patterns in the historical data.

Date	Actual Close	Predicted_Close
1995-01-03	459.109	667.145
1995-01-04	460.709	675.345
1995-01-05	460.339	673.592
1995-01-06	460.679	670.284
1995-01-09	460.829	670.228
...	...	...
2018-04-23	2670.290	2913.400
2018-04-24	2634.560	2925.766
2018-04-25	2639.399	2846.865
2018-04-26	2666.939	2812.266
2018-04-27	2669.909	2878.095

Table 4.1: Snippet of training results showing actual vs. predicted closing prices

Analysis of Testing Phase Results: While, the testing phase results in Table 4.2 display a significant deviation between the actual and predicted closing prices. For instance, on January 13, 2022, the model predicted a closing price of '7586.242', while the actual closing was only '4659.029', indicating a substantial error. The detailed reasons for these differences are discussed in Section 4.1.3.

Date	Actual_Close	Predicted_Close
2022-01-13	4659.029	7586.242
2022-01-14	4662.850	7310.554
2022-01-18	4577.109	7088.460
2022-01-19	4532.759	7195.341
2022-01-20	4482.729	7344.295
2022-01-21	4397.939	6727.009
2022-01-24	4410.129	6550.577
2022-01-25	4356.450	6693.504

Table 4.2: Snippet of testing results showing actual vs. predicted closing prices

Analysis of Performance Metrics: The performance metrics presented in Table 4.3 offer quantitative insight into the Polynomial Regression model's efficacy across both the training and testing phases. During the training phase, the model exhibits relatively low errors, with a MAE of 61.77, a RMSE of 81.99, and a MAPE of 5.77%, which are explained more in how the errors are calculated in this section 2.4. These metrics indicate that the model is highly effective at approximating the training data's historical price movements, suggesting a strong fit within the observed range.

However, the testing phase tells a different story, highlighted by significantly higher error rates: an MAE of 1301.87, an RMSE of 1576.16, and an MAPE of 32.09%. This huge increase in error metrics during the testing phase underscores the model's challenges when exposed to new, unseen market data.

Metric	Training Set	Testing Set
MAE	61.77	1301.87
RMSE	81.99	1576.16
MAPE	5.77%	32.09%

Table 4.3: Performance Metrics for Polynomial Regression Model

4.1.3 Analysis Causes Of Prediction Errors

The Polynomial Regression model has demonstrated notable differences between actual and predicted values, particularly in the testing phase. This section delves into the underlying causes contributing to these substantial prediction errors, which become evident when comparing training and testing performance metrics.

The first reason for this result happens could be because of the model may have over-fitted the training data, capturing noise and specific patterns that do not generalize well to unseen data.

The other reason for the significant difference could be the use of polynomial features of degree 2. Polynomial Regression tends to exaggerate predictions because it squares the values of features like Open, High, Low, and Volume. This squaring can overly increase changes in output, leading to predictions that are much higher or lower than they should be, especially in unpredictable market conditions.

For instance, when the market shows even small changes, the model squares these changes, making the outcome seem more drastic than reality. This is particularly evident when the model faces new market conditions that weren't part of its training data.

4.2 Decision Tree

The Decision Tree model was also trained as the Polynomial Regression, employing historical closing prices of the S&P 500 index, utilizing the same features: 'Open', 'High', 'Low', and 'Volume'. Unlike the polynomial regression, the decision tree model makes no assumptions about the linearity of the input-output relationship, allowing it to potentially capture more complex patterns in the data.

This model's results are presented through visualizations, predictions, and performance metrics, providing a comprehensive view of its efficacy and limitations.

4.2.1 Prediction Results

Visual Representation of Predictions: Figure 4.2 shows the actual versus predicted closing prices by the Decision Tree model, offering a visual assessment of its performance across different market conditions.

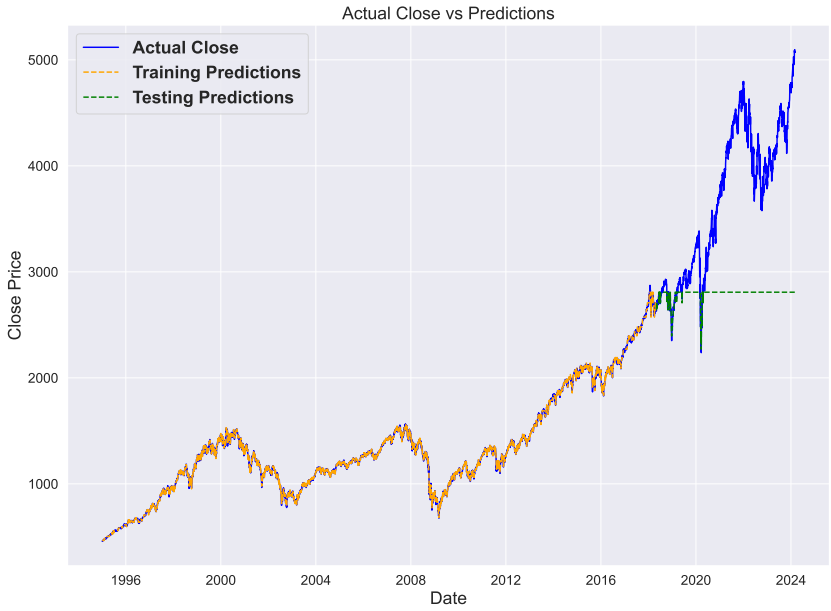

Figure 4.2: Actual Closing Prices vs. Predicted Prices by the Decision Tree Model

Prediction Table Result: This analysis includes detailed comparisons of actual versus predicted closing prices. The results highlight the model's capacity to learn from historical data and its limitations in generalizing to new data. Tables 4.4 and 4.5 showcase these comparisons.

Date	Actual Close	Predicted Close
1995-01-03	459.109985	465.801739
1995-01-04	460.709991	465.911269
1995-01-05	460.339996	466.801739
1995-01-06	460.679993	465.743199
1995-01-09	460.829987	465.801739
...	...	...
2018-04-23	2670.290039	2707.880647
2018-04-24	2634.560059	2641.943436
2018-04-25	2639.399902	2642.123256
2018-04-26	2666.939941	2640.973436
2018-04-27	2669.909912	2641.943436

Table 4.4: Snippet of training results showing actual vs. predicted closing prices by the Decision Tree model

Date	Actual Close	Predicted Close
2018-04-30	2648.050049	2641.943436
2018-05-01	2654.800049	2641.943436
2018-05-02	2635.669922	2641.943436
2018-05-03	2629.729980	2641.943436
2018-05-04	2663.419922	2641.943436
...	...	...
2024-02-23	5088.799805	2808.062512
2024-02-26	5069.529785	2808.062512
2024-02-27	5078.180176	2808.062512
2024-02-28	5069.759766	2808.062512
2024-02-29	5096.270020	2808.062512

Table 4.5: Snippet of testing results showing actual vs. predicted closing prices by the Decision Tree model

Performance Metrics: Table 4.6 summarizes the model's performance metrics, including MAE, RMSE, and MAPE, which quantify the accuracy and reliability of the Decision Tree in training and testing phases.

Metric	Training Set	Testing Set
MAE	5.90	884.18
RMSE	8.39	1111.20
MAPE	0.46%	21.31%

Table 4.6: Performance Metrics for Decision Tree Model

These metrics 4.6 indicate that the Decision Tree model performs exceptionally well on the training set as shown in Table 4.3, but similarly as polynomial regression, it shows increased error rates on the testing set. This increase is largely because the Decision Tree is limited to predicting within the range of values it has previously seen during training.

Figure 4.2 illustrates the Decision Tree model's predictions against the actual closing prices of the S&P 500 index. The graph reveals several key insights into the model's behavior over the training and testing periods.

During the training phase, represented by the orange dashed line, the model predictions align closely with the actual closing prices, indicating that the model has effectively learned the patterns within the historical data.

However, the testing phase, marked by the green dashed line, shows a significant deviation from the actual closing prices, especially as the market values rise well beyond the highest point observed in the training data. This limitation is particularly visible where the predictions flatline at the maximum value encountered during training, which is a characteristic limitation of Decision Trees when dealing with values outside their trained range.

4.2.2 Analysis of limitations of Decision Tree

The decision tree was applied to a different dataset. This scenario was chosen to test how well the model can generalize its predictions across a more volatile part of our data from 2022 to 2024, instead of the original dataset, to highlight some issues with the Decision Tree model (see 4.3).

In our analysis, the decision tree model was constrained by the highest and lowest values observed during its training phase. This limitation was evident as the model did not predict any values beyond these bounds during the testing phase. This behavior underscores a fundamental challenge with decision trees when applied to forecasting financial markets, which are subject to rapid changes and can reach new highs or lows that were not present in the training data. The model's inability to extrapolate beyond known data points from its training phase limits its utility in predicting future market movements accurately, particularly in a volatile market. To address these challenges,

incorporating models with extrapolation capabilities or regularly updating the training dataset to include the most recent data could be beneficial. This approach would allow the model to adjust to new market conditions and predict a wider range of possible outcomes.

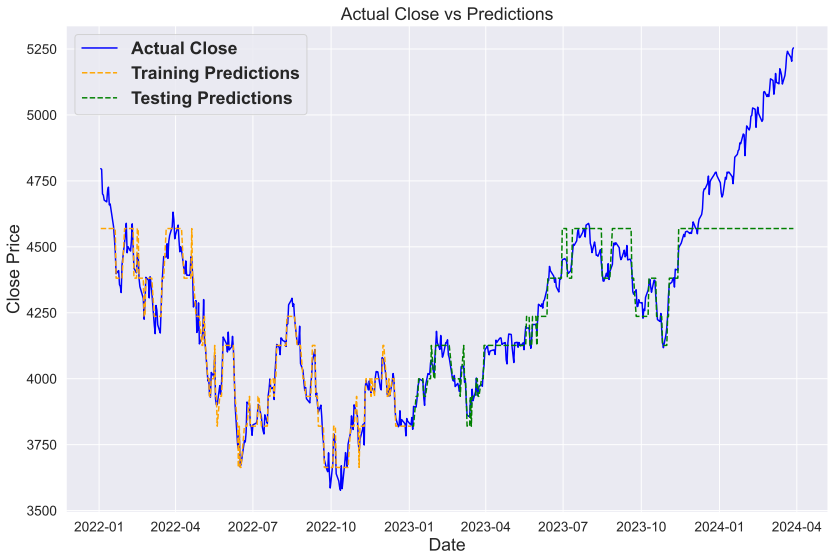

Figure 4.3: Actual Close Prices vs. Predictions by the Decision Tree Model

4.3 LSTM

The LSTM model applied a different processing approach compared to the Decision Tree and Polynomial Regression models. This approach is particularly effective for time series data like stock prices due to LSTM's ability to maintain state over sequences. For effective learning, the closing price data was first normalized 3.2.3, a critical step for neural networks, which helps in stabilizing the training process.

4.3.1 Model Training

Training the LSTM model involved different mechanisms, which first a sequence length of 30 days was chosen to predict the next day's closing price, ensuring the model had ample historical context for its predictions. This setup allows the LSTM to capture underlying patterns in the past month's

data to forecast future values. This duration, approximately one month of trading days, was selected to provide a balanced view of short-term trends and volatilities in the stock market, enhancing the model's ability to identify relevant patterns.

Then the LSTM model involved determining the optimal number of training epochs to avoid over-fitting while ensuring sufficient learning. The number of epochs was selected based on monitoring the decrease in validation loss during training, a method detailed in Section 3.3.1. An early stopping mechanism was employed to stop training when the validation loss began to plateau, indicating the model had learned as much as possible without starting to memorize the training data excessively. The optimal number of epochs was chosen to balance between under-fitting and over-fitting, ensuring the model generalized well to new, unseen data.

4.3.2 Prediction Results

The figure 4.4 illustrates the comparison between the actual S&P 500 index closing prices and the predictions made by the LSTM model. This visual analysis is crucial for evaluating the model's performance throughout both the training and testing periods.

As shown in the result figure 4.4, the LSTM model tracks closely with the actual market trends during the training phase, represented by the orange line for training predictions. The blue line indicates actual close prices and the green line indicates for testing prediction.

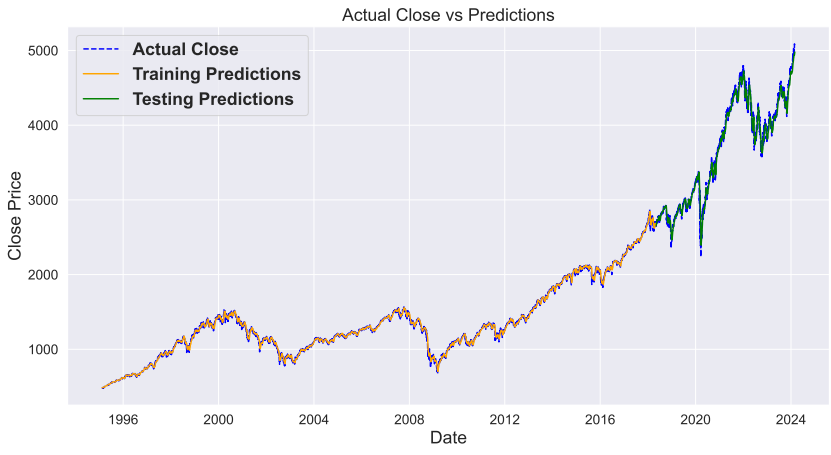

Figure 4.4: Actual Closing Prices vs. Predicted Prices by the LSTM Model

Date	Actual Close	Predicted Close
1995-02-14	482.549988	464.742290
1995-02-15	484.540009	465.726227
1995-02-16	485.720001	466.765259
1995-02-17	481.290001	467.260926
1995-02-20	482.700001	468.491936
...	...	...
2018-05-01	2654.800049	2642.731164
2018-05-02	2635.669922	2642.801462
2018-05-03	2629.729980	2642.549316
2018-05-04	2663.419922	2641.942260
2018-05-07	2672.629883	2643.764826

Table 4.7: Snippet of training results showing actual vs. predicted closing prices by the LSTM model

Date	Actual Close	Predicted Close
2018-05-08	2671.919922	2660.590088
2018-05-09	2697.790039	2600.786133
2018-05-10	2721.070068	2614.673340
2018-05-11	2727.719971	2620.781494
2018-05-14	2730.129883	2630.752441
...	...	...
2024-02-23	5088.799805	4622.431646
2024-02-26	5069.529785	4856.552933
2024-02-27	5078.180176	4856.242188
2024-02-28	5069.759766	4875.675293
2024-02-29	5096.270020	4885.317383

Table 4.8: Snippet of testing results showing actual vs. predicted closing prices by the LSTM model

Metric	Training Set	Testing Set
MAE	33.40	100.12
RMSE	39.24	117.65
MAPE	1.82%	2.46%

Table 4.9: Performance Metrics for LSTM Model

The performance metrics for the LSTM model demonstrate its effective training outcomes, with a significantly high accuracy reflected in the training

set results. Moreover, unlike the Decision Tree and Polynomial Regression models, the LSTM shows even better results in the testing set, indicating that LSTM has ability to handle both historical data and new unseen scenarios better way than the other models.

4.3.3 Analysis of LSTM Model Performance

- **Training Time:** LSTM models are computationally intensive due to their complex architecture, which includes multiple layers of neurons that maintain a state over time. Training an LSTM model on extensive datasets, such as the historical stock prices from 1995 to 2024, can be time-consuming.
- **Resource Requirements:** The LSTM model requires significant computational resources, especially when dealing with large datasets. This includes the need for powerful GPUs to execute the training process, which might not be as critical for simpler input calculation like polynomial regression and decision trees use.

Chapter 5

Discussion

5.1 Model Comparison

This section compares the performance of Polynomial Regression, Decision Tree, and LSTM models in predicting the closing prices of the S&P 500. The comparison is based on visual representations from figures 5.1 to 5.3 and quantified through performance metrics such as MAE, RMSE, and MAPE.

5.1.1 Visual Analysis of Predictions

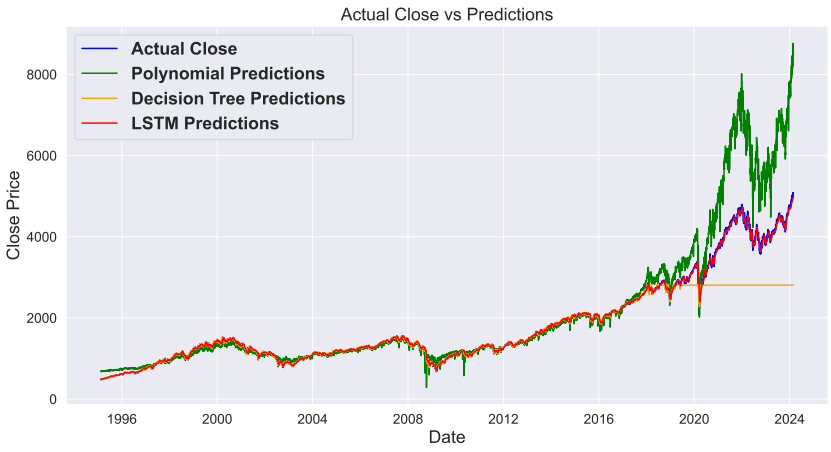

Figure 5.1: Actual Close versus Model Predictions. This plot shows the performance of Polynomial Regression, Decision Tree, and LSTM models in predicting the S&P 500 closing prices over the years.

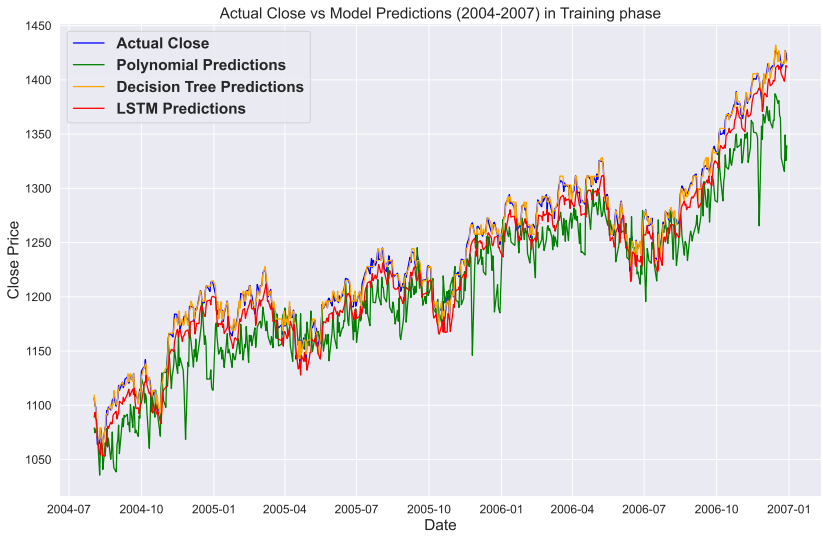

Figure 5.2: Close Price Predictions Comparison during the Training Phase (2004-2007). This detailed view highlights how each model aligns with the actual market trends during a stable period, emphasizing the nuances of each model's prediction capabilities.

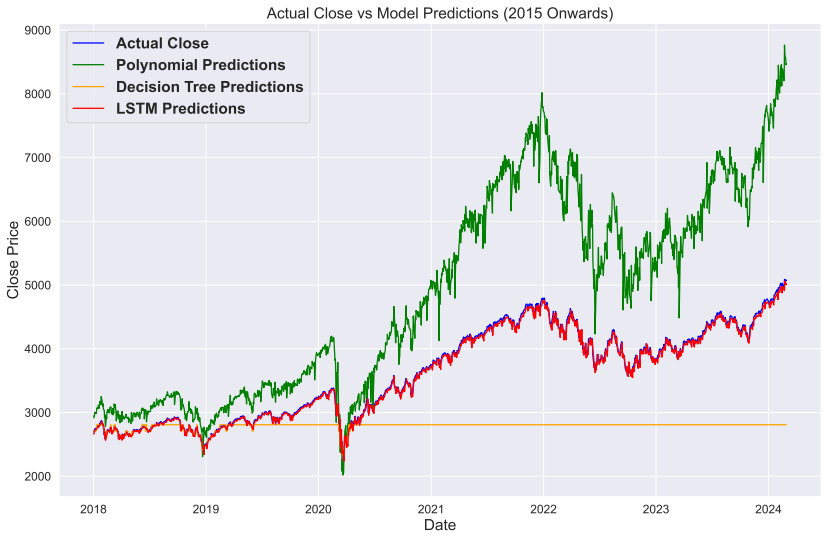

Figure 5.3: Test Phase Model Predictions (2018 Onwards). This graph illustrates the performance of each predictive model against the actual closing prices of the S&P 500 during the test phase, covering the most recent data from 2018 onwards.

Figure 5.1 provides a comprehensive view of each model's predictions over an extended period. It shows that while all models track the general trend of the actual closing prices, there are significant disparities in accuracy and response to market volatility.

Figure 5.2 focuses on the stable market period between 2004 and 2007. During this phase, the Decision Tree and LSTM models demonstrate a close alignment with the actual market trends, showcasing their robustness in tracking consistent patterns. On the other hand, the Polynomial Regression model exhibits notable deviations from the actual trends, indicating a potential overfitting issue or a lack of adaptability to less volatile conditions within the training data.

In contrast, Figure 5.3 from the testing phase shows the Decision Tree model is limited by its inability to predict beyond the highest and lowest values seen during training. This results in flat predictions when new market highs or lows occur. The Polynomial Regression model also struggles, as its tendency to overfit and exaggerate based on squared features leads to significant prediction errors under volatile conditions. In comparison, the LSTM model adapts better, maintaining closer alignment with actual prices

and demonstrating its effectiveness in navigating unpredictable market trends.

Prediction Result in Table Form: The table 5.1 shows the result of the actual closing prices with the predictions made by each of the three models: Polynomial Regression, Decision Tree, and LSTM. This result explains a side-by-side comparison to show which model predicts the closest to the actual prices over different intervals.

Date	Actual_Close	Polynomial	Decision_Tree	LSTM
1995-02-14	482.549	679.072	485.322	483.321
1995-02-15	484.540	690.029	484.572	484.261
1995-02-16	485.220	686.626	484.572	485.211
1995-02-17	481.970	685.117	486.890	486.144
1995-02-21	482.720	679.872	485.555	486.735
...	...	...	...	...
2024-02-22	5087.029	8765.893	2808.062	4936.713
2024-02-23	5088.799	8617.675	2808.062	4945.065
2024-02-26	5069.529	8548.768	2808.062	4957.068
2024-02-27	5078.180	8455.260	2808.062	4967.986
2024-02-28	5069.759	8470.372	2808.062	4978.504

Table 5.1: Comparison of actual and predicted closing prices by Polynomial Regression, Decision Tree, and LSTM models

As it is shown in 5.1, the LSTM model demonstrates the most reliable performance, closely aligning with actual closing prices.

Quantitative Metrics Comparison: The comparative analysis of MAE, RMSE, and MAPE (Figures 5.4, 5.5, 5.6) reveals that the LSTM model consistently outperforms the other two across all metrics. This indicates its higher accuracy and lower error rates, making it a more reliable model for financial forecasting.

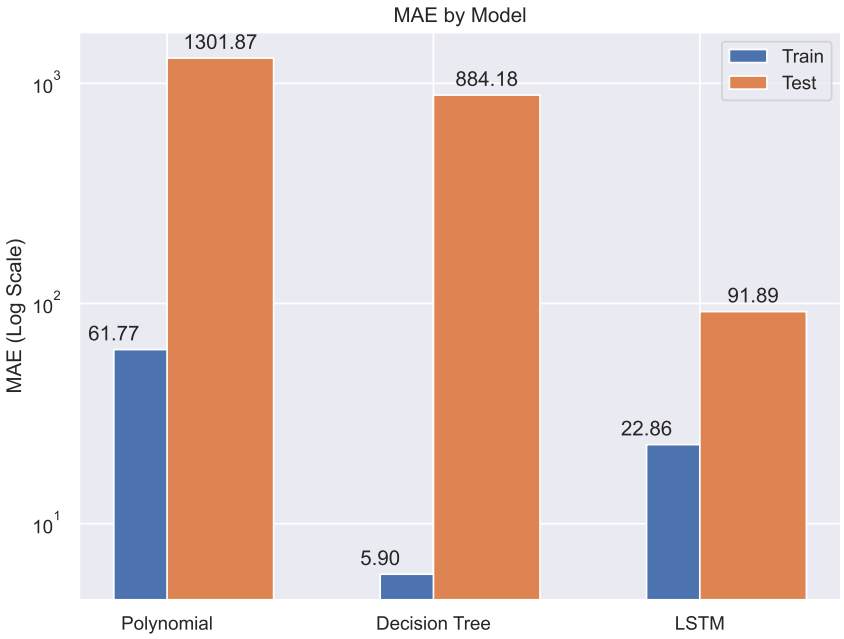

Figure 5.4: MAE

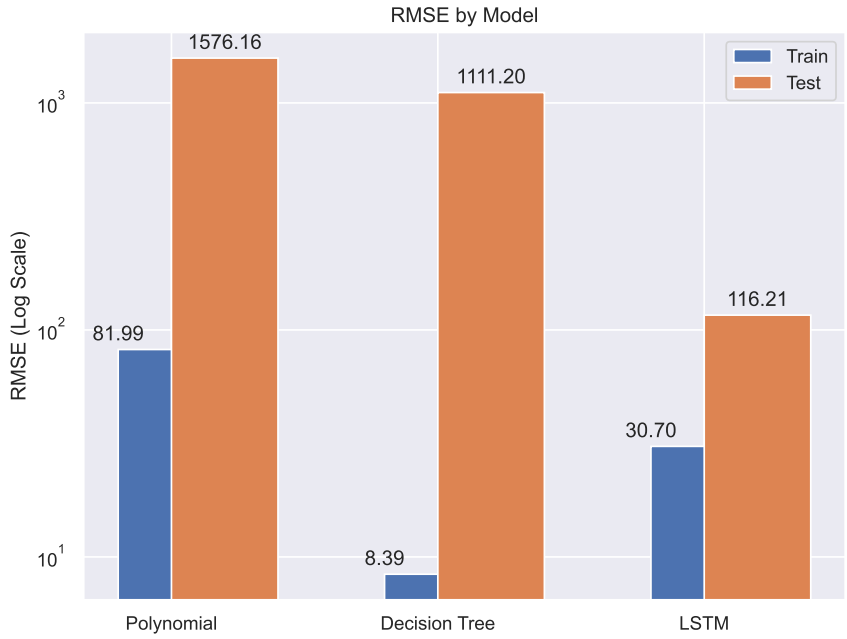

Figure 5.5: RMSE

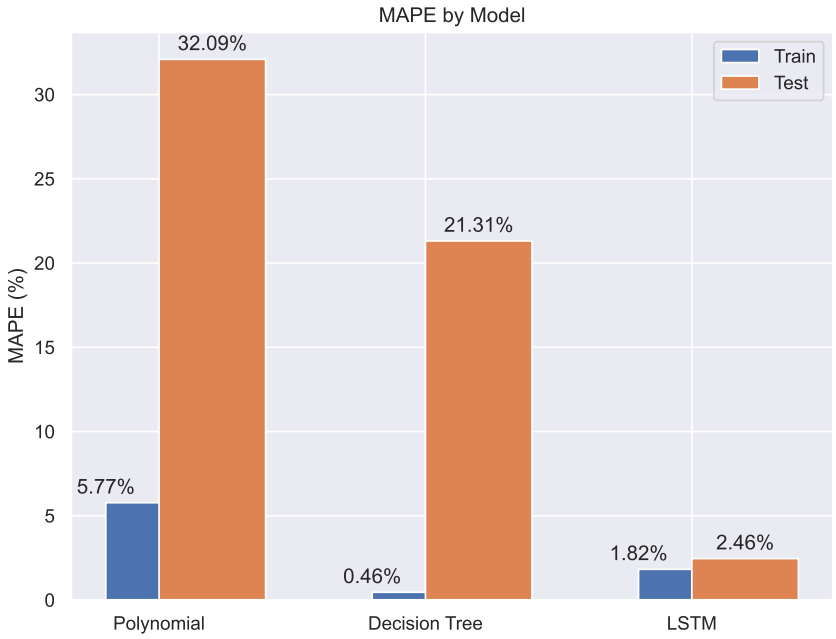

Figure 5.6: MAPE

5.1.2 Detailed Analysis of Model Performance

In Figure 5.2, both the Polynomial Regression and LSTM models closely track the actual market prices during the stable period of 2004-2007. However, the LSTM model displays slightly better adaptability to minor fluctuations. This can be attributed to its ability to remember long-term dependencies, a feature that Polynomial Regression lacks due to its reliance on fixed-degree polynomial transformations, which may not capture all dynamic trends especially in a volatile market.

During the more volatile period post-2018, as shown in Figure 5.3, the LSTM model clearly outperforms the Decision Tree and Polynomial Regression models. The Decision Tree model, while adept at capturing trends within the range it has been trained on, fails to extrapolate beyond these bounds, demonstrating a key limitation in handling new market highs and lows. This limitation is further evidenced in Table 5.1, where the Decision Tree predictions do not capture the peak values observed in the actual market, reflecting its inability to adjust to new and higher price levels.

The Polynomial Regression model, while generally reliable in more stable

conditions, shows significant divergence from actual values during periods of high volatility. This divergence is exacerbated by overfitting, where the model becomes too tailored to the historical data it was trained on, making it less adaptable to new, unforeseen market trends. The effects of this overfitting are evident in Table 5.1, where the Polynomial Regression predictions are consistently higher or lower than the actual values, especially in the later years as market conditions change.

The LSTM model not only closely follows the actual price trends as shown in both figures but also maintains consistent accuracy across various market conditions, as detailed in Table 5.1. Its predictions remain close to the actual closing prices, underscoring its effectiveness in handling both stable and volatile market phases.

Figure 5.4 shows the MAE comparison among the models, with the LSTM model demonstrating significantly lower error rates in both training and testing phases compared to the other models. This aligns with its superior performance in capturing market trends accurately and adapting to new conditions. Similarly, Figures 5.5 and 5.6 for RMSE and MAPE respectively, further validate the LSTM's robustness. Both figures exhibit lower error values for LSTM across the training and testing phases, indicating its consistency and reliability in forecasting. These metrics collectively reinforce the qualitative assessments made earlier, confirming LSTM's effectiveness in handling both stable and volatile market phases.

This comprehensive analysis highlights the superiority of LSTM in this application, largely due to its dynamic learning architecture that is well-suited for financial datasets characterized by their non-linear and time-dependent nature. The consistent performance of LSTM across different metrics and scenarios, as shown in the tables and figures, strongly supports its selection for forecasting tasks in complex and unpredictable financial markets.

5.1.3 Strengths and Weaknesses

The evaluation of model performance across various conditions highlights distinct strengths and weaknesses for each approach. The LSTM model demonstrates exceptional adaptability and precision in predicting market trends, crucial in financial markets influenced by past events. Its robust performance in both stable and volatile periods, as evidenced by lower error metrics (MAE, RMSE, MAPE) and closer alignment with actual price trends, underscores its utility for complex time-series data. However, the complexity of LSTM models may lead to longer training times and greater computational

demands, and they are prone to overfitting if not appropriately regularized.

In contrast, the Decision Tree model excels when market conditions closely match the scenarios seen during its training phase. Its interpretability and ease of implementation make it suitable for applications requiring clear decision-making processes. Nevertheless, its incapacity to extrapolate beyond the highest and lowest data points encountered during training limits its effectiveness in markets experiencing new highs and lows, as highlighted during the test phases.

Polynomial Regression offers effective modeling of complex patterns within stable market conditions, utilizing polynomial transformations to approximate nonlinear relationships. This model's strength lies in its ability to capture and predict trends within the range of its calibration. However, it tends to overfit training data, capturing noise and specific patterns that do not generalize well to unseen data, leading to significant prediction errors in periods of high volatility.

Overall, the strengths and weaknesses of each model reflect their suitability for different forecasting scenarios in financial markets. The selection of a model should thus consider the specific attributes of the data set, the expected market conditions, and the forecasting objectives.

Chapter 6

Conclusions

This thesis evaluated three machine learning models—Polynomial Regression, Decision Tree, and LSTM—on their effectiveness in forecasting short-term movements in the S&P 500 index. Each model displayed unique strengths and limitations impacting their suitability for financial forecasting.

The LSTM model stood out for its robustness across varying market conditions, demonstrating superior adaptability and accuracy through lower performance metrics (MAE, RMSE, MAPE). Its ability to leverage long-term dependencies makes it particularly effective for the volatile nature of financial markets.

In contrast, Polynomial Regression struggled with overfitting, leading to substantial prediction errors during periods of market volatility. The Decision Tree model was hindered by its inability to predict beyond the historical extremes seen in the training data, limiting its effectiveness when the market reached new highs or lows.

These insights highlight the importance of selecting the right model based on the dataset's characteristics and the forecasting requirements. Future efforts could focus on refining LSTM models to prevent overfitting and exploring hybrid models that might offer improved accuracy in forecasting financial time series. Expanding the dataset and incorporating broader economic indicators could also enhance the models' ability to capture complex market dynamics, supporting better decision-making in financial analysis and policy development.

References

[1] A. Fauzan, M. SusanAnggreainy, N. Nathaniel, and A. Kurniawan, "Predicting Stock Market Movements Using Long Short-Term Memory (LSTM)," in *2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, Sep. 2023. doi: 10.1109/AiDAS60501.2023.10284713 pp. 144–147. [Online]. Available: https://ieeexplore.ieee.org/document/10284713 [Pages 1 and 11.]

[2] I. Medarhri, M. Hosni, N. Nouisser, F. Chakroun, and K. Najib, "Predicting Stock Market Price Movement using Machine Learning Techniques," in *2022 8th International Conference on Optimization and Applications (ICOA)*, Oct. 2022. doi: 10.1109/ICOA55659.2022.9934252 pp. 1–5, iSSN: 2768-6388. [Online]. Available: https://ieeexplore.ieee.org/document/9934252 [Page 1.]

[3] M. El Hajj and J. Hammoud, "Unveiling the Influence of Artificial Intelligence and Machine Learning on Financial Markets: A Comprehensive Analysis of AI Applications in Trading, Risk Management, and Financial Operations," *Journal of Risk and Financial Management*, vol. 16, no. 10, p. 434, Oct. 2023. doi: 10.3390/jrfm16100434 Number: 10 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1911-8074/16/10/434 [Page 1.]

[4] "S&P 500®." [Online]. Available: https://www.spglobal.com/spdji/en/indices/equity/sp-500/#overview [Page 4.]

[5] "Yahoo Finance - Stock Market Live, Quotes, Business & Finance News." [Online]. Available: https://finance.yahoo.com/ [Pages 4 and 16.]

[6] E. Ostertagová, "Modelling using Polynomial Regression," *Procedia Engineering*, vol. 48, pp. 500–506, Jan. 2012. doi: 10.1016/j.proeng.2012.09.545. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705812046085 [Page 7.]

[7] Y.-y. SONG and Y. LU, "Decision tree methods: applications for classification and prediction," *Shanghai Archives of Psychiatry*, vol. 27, no. 2, pp. 130–135, Apr. 2015. doi: 10.11919/j.issn.1002-0829.215044. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/ [Page 7.]

[8] "Python Decision Tree Classification Tutorial: Scikit-Learn DecisionTreeClassifier." [Online]. Available: https://www.datacamp.com/tutorial/decision-tree-classification-python [Pages vii and 8.]

[9] M. A. Nielsen, "Neural Networks and Deep Learning," 2015, publisher: Determination Press. [Online]. Available: http://neuralnetworksanddeeplearning.com [Pages vii and 9.]

[10] E. Singh, N. Kuzhagaliyeva, and S. M. Sarathy, "Chapter 9 - Using deep learning to diagnose preignition in turbocharged spark-ignited engines," in *Artificial Intelligence and Data Driven Optimization of Internal Combustion Engines*, J. Badra, P. Pal, Y. Pei, and S. Som, Eds. Elsevier, Jan. 2022, pp. 213–237. ISBN 978-0-323-88457-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323884570000059 [Pages vii and 10.]

[11] "Figure 1. Structure diagram of LSTM." [Online]. Available: https://www.researchgate.net/figure/Structure-diagram-of-LSTM_fig1_373928070 [Pages vii and 11.]

[12] How to calculate mean absolute error - shiksha online. [Online]. Available: https://www.shiksha.com/online-courses/articles/mean-absolute-error/ [Page 12.]

[13] J. Moody. What does RMSE really mean? [Online]. Available: https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e [Page 12.]

[14] Mean absolute percentage error (MAPE): What you need to know. [Online]. Available: https://arize.com/blog-course/mean-absolute-percentage-error-mape-what-you-need-to-know/ [Page 12.]

[15] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. "O'Reilly Media, Inc.", Oct. 2012. ISBN 978-1-4493-2361-5 Google-Books-ID: v3n4_AK8vu0C. [Page 15.]

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and . Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html [Page 15.]

[17] Keras: Deep learning for humans. [Online]. Available: https://keras.io/ [Page 15.]

TRITA-EECS-EX- 2023:0000

€€€€ For DIVA €€€€

{
”Author1”: { ”Last name”: ”Ahmed”,
”First name”: ”Mahad”,
”Local User Id”: ”u100001”,
”E-mail”: ”mahadah@kth.se”,
”organisation”: {”L1”: ”School of Electrical Engineering and Computer Science”,
}
},
”Author2”: { ”Last name”: ”Mohammed”,
”First name”: ”Ismail”,
”Local User Id”: ”u100002”,
”E-mail”: ”ismmoh@kth.se”,
”organisation”: {”L1”: ”School of Electrical Engineering and Computer Science”,
}
},
”Cycle”: ”1”,
”Course code”: ”II42X”,
”Credits”: ”15.0”,
”Degree1”: {”Educational program”: ”Degree Programme in Computer Engineering”
,”programcode”: ”TIDAB”
,”Degree”: ”Bachelors degree”
,”subjectArea”: ”Technology”
},
”Title”: {
”Main title”: ”How do different machine learning models compare in their ability to predict short-term movements in the S&P 500 index?”,
”Language”: ”eng” },
”Alternative title”: {
”Main title”: ”Hur jämför sig olika maskininlärningsmodeller när det gäller deras förmåga att förutsäga kortsiktiga rörelser i S&P 500-indexet?”,
”Language”: ”swe”
},
”Supervisor1”: { ”Last name”: ”Li”,
”First name”: ”Zhenyu”,
”Local User Id”: ”u100003”,
”E-mail”: ”zhenyuli@kth.se”,
”organisation”: {”L1”: ”School of Electrical Engineering and Computer Science”,
”L2”: ”Computer Science” }
},
”Examiner1”: { ”Last name”: ”Olsson”,
”First name”: ”Håkan”,
”Local User Id”: ”u1d13i2c”,
”E-mail”: ”hakano@kth.se”,
”organisation”: {”L1”: ”School of Electrical Engineering and Computer Science”,
”L2”: ”Computer Science” }
},
”National Subject Categories”: ”10201, 10206”,
”Other information”: {”Year”: ”2024”, ”Number of pages”: ”viii,47”},
”Copyrightleft”: ”copyright”,
”Series”: { ”Title of series”: ”TRITA-EECS-EX” , ”No. in series”: ”2023:0000” },
”Opponents”: { ”Name”: ”Ahmedhadi Bashir Ibrahim & Besher Alshikha ”},
”Presentation”: { ”Date”: ”2024-06-03 14:00”
,”Language”:”eng”
,”Room”: ”Ka-304”
,”Address”: ”Isafjordsgatan 22 (Kistagången 16)”
,”City”: ”Stockholm” },
”Number of lang instances”: ”2”,
”Abstract[eng]”: €€€€


```
% \generalExpl{Enter your abstract here!}

This thesis explores the application of machine learning techniques within the financial technology
sector, specifically targeting the prediction of the S\&P 500 index. The S\&P 500 is a key indicator
of the US economy which reflects the stock performance of 500 major companies. Traditional
statistical methods for predicting stock market trends often struggle with the volatility and
complexity of financial markets. Our research aims to fill this gap by comparing the effectiveness of
various machine learning models which are Polynomial Regression, Decision Trees, and Long Short-Term
Memory (LSTM) networks to predict short-term movements of the S\&P 500 index.

The significance of this problem lies in the growing reliance on ML models by financial companies to
inform investment strategies. The study uses historical S\&P 500 data from 1995 to 2024, retrieved
from Yahoo Finance, focusing on metrics such as Open, High, Low, Close, and Volume prices. The data
was preprocessed to ensure consistency and was divided into training and testing sets to evaluate the
models.

Our results show that the Polynomial Regression and Decision Tree models performed well on the
training data but they had significant errors on the testing data. While, the LSTM model showed
better performance, effectively capturing both short-term and long-term market trends. Based on these
findings results, suggest that LSTM networks provide the most reliable predictions.
```

```
\subsection*{Keywords}
\begin{scontents}[store-env=keywords,print-env=true]
% SwedishKeywords were set earlier, hence we can use alternative 2
\InsertKeywords{english}
\end{scontents}
```

€€€€,
"Keywords[eng]": €€€€
S&P 500, ML-models, LSTM, Polynomial Regression, Decision Trees €€€€,
"Abstract[swe]": €€€€

Denna avhandling utforskar användningen av maskininlärningsmodeller (ML-modeller) inom finansiell teknik, med särskilt fokus på att förutsäga S\&P 500 index. S\&P 500 är en viktig indikator på USA:s ekonomi som återspeglar aktieprestationen för 500 stora företag. Traditionella statistiska metoder för att förutsäga aktiemarknadstrender kämpar ofta med volatiliteten och komplexiteten på finansmarknaderna. Vår forskning syftar till att fylla denna lucka genom att jämföra effektiviteten hos olika ML-modeller, nämligen polynomregression, beslutsträd och Long Short-Term Memory, för att förutsäga kortsiktiga rörelser i S\&P 500-indexet.

Betydelsen av detta problem ligger i den ökande tilliten till ML-modeller av finansiella företag för att informera investeringsstrategier. Studien använder historiska S\&P 500-data från 1995 till 2024, hämtade från Yahoo Finance, med fokus på mätvärden som öppning, högsta, lägsta, stängning och volympriser. Data förbehandlades för att säkerställa konsistens och delades in i tränings- och testuppsättningar för att utvärdera modellerna.

Våra resultat visar att polynomregression och beslutsträdsmodeller presterade väl på träningsdatan men hade betydande fel på testdatan. Däremot visade LSTM-modellen bättre prestanda genom att effektivt fånga både kortsiktiga och långsiktiga marknadstrender. Baserat på dessa resultat indikerar studien att LSTM ger de mest tillförlitliga förutsägelserna.

€€€€,
"Keywords[swe]": €€€€
S&P 500, ML-modeller, LSTM, Polynomregression, Beslutsträd €€€€,
}