City, University of London MSc in Data Science

Project Report

2024

Deep Learning Approaches for Stock Price Forecasting: Implementation of CNN, LSTM, and CNN-LSTM Models on FTSE 100 Index

Alisher Amerkhojayev

Supervised by:

16/10/2024

Table of Contents

Chapter 1 Introduction

1.1 Background

Stock price prediction has been one of the most important challenges that face financial markets, considering the potential benefits such an anticipation would bring to investors, financial institutions, and policymakers. It helps investors, financial organizations, and policymakers make better decisions on portfolio management, evaluation of risk, and optimization of strategy. In essence, therefore, it presents an excellent avenue through which more-informed decisions may be arrived at. However, their nature itself imposes an especially difficult task on financial markets, characterized as they are by properties of complexity, volatility, and nonlinearity. Indeed, traditional models, innumerable as they might be, are found incapable of encapsulating the various aspects of dynamic nonlinear nature in stock markets, since most of them rely on linear assumptions and historical trends. It has only been with the rise of deep learning methods such as CNNs and LSTM network architectures in recent years that new opportunities for modelling such relationships have been opened.

This project was inspired by the necessity to investigate and assess the predictive performance of such state-of-the-art deep learning architectures, CNN, LSTM, and the hybrid CNN-LSTM model on stock price prediction for the FTSE 100. This paper integrates some essential macroeconomic indicators like inflation, interest rates, GDP growth, unemployment rates, and exchange rates to give a wider economic context to the predictive models. These indicators, according to the literature, reflect a high relevance to market performance and investors' behaviour. Therefore, they are necessary variables in stock market forecasting models.

1.2 Problem Definition and Project Objectives

This project deals with the core problem of formulating accurate models for stock price predictions using deep learning techniques. More precisely, this deals with identifying the best model to forecast daily stock prices by integrating macroeconomic indicators to gain a better understanding of the elements leading to the movement in the stock market.

The main objectives of the project are:

1. Model Performance Evaluation: The performance of CNN, LSTM, and CNN-LSTM using daily stock price predictions for the FTSE.
2. Impact of the Macroeconomic Indicators: This will demonstrate the impact of using macroeconomic indicators to improve the accuracy of the forecast by such models.
3. Model Generalization: Each model generalization capability will be studied by considering that the performance of the model when tested with data it had never seen may change considering changes in market situations, which are more often volatile.

4. Approaches Comparison: This work compares the strengths of CNN, LSTM, and CNN-LSTM in capturing the local pattern of stock market data using CNN and its long-term dependency using LSTM.
5. Sliding Window Methodology: The proposed system is designed based on the Sliding Window methodology in the experimentation section for these models to capture short-run and long-run trends in stock price prediction.

The target beneficiaries for this project include financial analysts, portfolio managers, and policy makers to make informed decisions in stock trading, risk management, and economic policy. The project contributes to the increasing studies on forecasting stock prices, especially regarding the use of macroeconomic indicators within deep learning architectures by evaluating the accuracy and generalization of the deep learning models.

1.3 Research Questions

The basis for this work includes the following research questions:

1. Which deep learning model-CNN, LSTM, or CNN-LSTM-better improves the stock price prediction?

2. How would including macroeconomic indicators improve the performances of stock price prediction models?

3. What changes will the sliding window approach bring about on the prediction accuracy for these models?

4. How generalizable are the models on new data with times of high market volatility?

The project outcomes include developing the deep learning models for stock price prediction and their comparative analysis. In the course of development, these models will be considered in regard to predictability, generalization, and response to the changes of the market. The outcome of this will be presented as a profound assessment of advantages and disadvantages of models, which will give further insights into subsequent applications related to financial forecasting.

1.4 Project Outcomes

The major deliverable for this project is a comparative evaluation of the performance for CNN, LSTM, and CNN-LSTM deep learning models in stock price prediction. Concretely, this project offers the following:

1. Optimized set of deep learning models: Techniques for hyperparameter optimization are applied to the models as a guarantee of optimal performance.

2. Evaluation Metrics and Results: The various models are evaluated using main performance indicators such as MAE, RMSE, and Directional Accuracy across the training and test datasets.

3. Insights about Model Generalization: The project demonstrates the ability or inability of the models to generalize on unseen test data. This is important for practical applications in stock price forecasting.

1.5 Methods and Work Plan

The project methodology is divided into three important phases: data pre-processing, model development and training, and evaluation. This included stock market data and macroeconomic indicators collected from 2013 to 2023 for the two indices, FTSE 100. We have changed the topic moving from 2 indices to one, due to time constraints and for provision of in depth analysis. Normalization of data was then done using the Min-Max Scaler, hence consistency at variable level. A sliding window approach was utilized such that the input to each model consisted of a 20-day train sequence to forecast the stock price on the next day.

Development covered the construction, tuning, and training for CNN, LSTM, and CNN-LSTM. All these models have undergone wide hyperparameter tuning with Keras Tuner to have the best configuration. Later, the models were trained on historical data, and sets were put aside to allow the models to avoid overfitting and that their performance be good. The evaluation phase consisted of testing the models on unseen data, namely 2021-2023, to evaluate the generalization capability. Performance metrics have been quantified in terms of MAE, RMSE, R-squared, and directional accuracy.

1.6 Structure of the report

Chapter 2 provides an overview on the existing literature on economic indicators, stock price prediction, and the CNN, LSTM, CNN-LSTM models and their use in stock prediction.
Chapter 3 Outlines the data collection, pre-processing steps, design, training and evaluation methods
Chapter 4 Outlines and analyses the performance results of the models. Provides an in-depth understanding of the design and problem at hand.

Chapter 5 This chapter discusses the implications of the results, highlighting the strengths and limitations of each model in predicting stock prices, especially in light of changing macroeconomic conditions

Chapter 6 The final chapter summarizes the findings, evaluate the whole project and suggests future work.

Chapter 2 Literature Review

2.1 Introduction

Accurate stock price forecasting is a critical endeavour in the field of finance, offering substantial benefits for investors, financial institutions, and policymakers.. Neural networks and neuro-fuzzy models have been found quite suitable for the stock market. Experimental results reveal that soft computing techniques generally outperform traditional models in their domain of applicability and provide better results with respect to trading systems, yielding higher forecasting accuracy. Traditional methods often cannot capture complex, nonlinear, dynamic financial markets. Now, deep learning has opened new possibilities to model such complexities. Among many such models, two of the most important are the Convolutional Neural Networks and the Long Short-Term Memory networks.

This Literature review will present a critical overview of the application of economic indicators toward stock market prediction and the use of CNN and LSTM models on stock price prediction. To this effect, we consider notable studies, their methodologies, and findings as we seek to find insights and gaps towards our project objectives, which aims at indicating the deep learning approach that yields the best performance both on the S &P 500 and the FTSE 100 when relevant economic indicators are factored in.

2.2 Economic Indicators and Stock Markets

Economic indicators are quantitative measures that reflect the overall health and direction of an economy. They have deep impacts on stock markets, influencing investor confidence, profitability of firms, and expectations about an economy. These indicators improve the capabilities of predictive models by capturing the dynamics of markets for better accuracy. Among many others, Rapach and Zhou (2012) investigate how such well-known economic variables as interest rates, inflation, dividend-price ratios, and earnings-price ratios affect stock market returns. This data is U.S. equity-market-centric; the focus is on the S&P 500 and covers several decades, including variation within the business cycle. Predictive regressions are done with the methodological approach in assessing the predictive power of these predictors on future stock returns. It follows that the results suggest that these indicators possess some predictive power, especially during times when the economy is under turmoil, as apt reminders that at an aggregate level, the economy and the stock market are intertwined. Their findings emphasize that economic indicators are the most important variables in the prediction of stock market behaviour (Rapach & Zhou, 2012). dditionally, in the study conducted by Maghayereh (2003) on the Jordanian Stock Exchange between the period 1987-2000, interest rate and inflation have been found to have a notable relationship with the long-term stock price index. In another comparative study of the effects of the macroeconomic indicators on the stock market between US end Japan conducted by (Humpe & Macmillan, 2009), they found that the interest rates and inflation have affected both the stock markets. However, they note that not all the tested indicators affected the markets to same degree. Moreover, money supply had no effect on the US market and a sizeable effect on the Japanese market. This highlights the importance of using multiple indicators as some indicators might not be as effective as others. This is further highlighted by findings from conducted by Gan et al. (2006) about the relationship between the indicators and the New Zealand Stock Exchange. It was found that on top of the already mentioned interest rate and money supply, real GDP has been found to have a noteworthy effect on the performance of the New Zealand stock market. Moreover, they have concluded that inflation rate and exchange rate do not affect the results as much as it could be from the more sizeable markets like in the US, Korea, and Japan.

2.2.1 Interest Rates

Interest rates, determined by central banks, are pivotal in influencing economic activity. High interest rates increase the cost of borrowing, suppressing consumer spending and business investment, leading to lower corporatche earnings and declining stock prices. Conversely, lower interest rates can stimulate economic activity and boost stock markets. Bernanke and Kuttner (2005) conducted an event-study analysis to examine the impact of unexpected changes in the federal funds rate on U.S. stock prices. Utilizing data from 1989 to 2002, they found that a 25 basis point unexpected decrease in the federal funds rate is associated with approximately a 1% increase in broad stock indexes. This effect was attributed to the influence of monetary policy on expected future excess returns and dividends. The study highlights the sensitivity of developed markets to interest rate changes and underscores the importance of including interest rates in predictive models for indices like the S&P 500 and FTSE 100.

Kyereboah-Coleman & Agyire-Tettey (2008) have conducted a study to find how the macroeconomic indicators affect the Ghana Stock Exchange. They have found that the lending rate is a strong indicator of the direction of the stock market as the policy changes significantly affect the businesses in Ghana. The significance of interest rates extends beyond the U.S. context. In the U.K., Ioannidis and Kontonikas (2008) analyzed the relationship between monetary policy and the stock market using data from 1972 to 2002. They found that contractionary monetary policy shocks (interest rate increases) lead to significant declines in stock prices since the expected returns decline meaning that the interest rate has a lingering effect after the enforcement. Their findings reinforce the notion that interest rates are a critical determinant of stock market performance in developed economies.

2.2.2 Inflation Rate

Inflation affects the purchasing power of consumers and the input costs of businesses. High inflation can erode real returns on investments and increase uncertainty, negatively impacting stock prices (Fama, 1981). Geske and Roll (1983) argued that inflation adversely affects stock prices due to its impact on future cash flows and discount rates. Inflation sends a signal that there will be changes in the policy that would likely contract the spending, lowering the demand for the products and general spending which would negatively affect the stock market as cash savings will increase. This would lead to investors diverting their investment from the stock market as it would be corroding both their gains and profitability, thus redirecting their money into tangible assets (commodities) such as gold.

Humpe and Macmillan (2009) conducted a comparative study of the U.S. and Japanese stock markets, employing cointegration analysis on data from 1965 to 2005. They found a significant negative relationship between stock prices and inflation in the U.S., indicating that higher inflation leads to lower stock prices. Specifically, a 1% increase in the Consumer Price Index (CPI) was associated with a 1.4% decrease in the U.S. stock market index. In contrast, the Japanese market did not exhibit a statistically significant relationship, suggesting that the impact of inflation on stock prices may vary across countries.

The adverse effect of inflation on stock prices is further supported by Apergis and Eleftheriou (2002), who analyzed the Athens Stock Exchange and found that inflation negatively affects stock returns. Their study utilized vector error correction models and highlighted that investors demand higher returns to compensate for the erosion of purchasing power, leading to decreased stock valuations.

2.2.3 Gross Domestic Product (GDP) Growth

GDP reflects the total value of the overall economic output and is considered as a direct indicator of economic health. Normally, the higher the level of GDP growth, the more vigorous the economic activities are; accordingly, the corporate profit and investor confidence increase (Chen et al., 1986). For instance, Reddy (2012) conducted a research during the period from 1991 to 2010 covers the Indian stock market; SENSEX is applied as a representative of its stock price. It means that the study measures the magnitude of GDP growth rate impact on stock market returns. A regression analysis was conducted in determining the relationship between the variables. The result of such indicates that GDP growth has a strong positive impact on stock market returns. These results indicated that GDP growth accounted for a large portion of the variation in stock prices and, thus, became a major determinant of stock market performance for the study period.

The work of Gan et al. (2006), using quarterly data from 1990 to 2003, considered the relationship between some macroeconomic variables and the New Zealand stock market. Through cointegration tests and vector error correction models, they established that there is a positive effect of the GDP on stock prices in the long run. More precisely, for every 1% increase in GDP, there was an equivalent 0.8% increase in the stock market index. This, therefore, implies that the inclusion of GDP variables in the forecasting models improves the forecast accuracy as it encompasses the growth perspectives of the economy.

2.2.4 Unemployment Rate

The study by Pan (2018) based on panel cointegration and Granger causality tests, examines the relationship between stock market activity and unemployment over the period spanning 2001-2017 across 30 advanced and 11 developing countries. More precisely, sharp contrasts between advanced and developing economies were highlighted; specifically, in the G7 group, one-way causality was running from stock prices to unemployment, which suggests that changes in unemployment could be predicted by how well the stock market was performing. In fact, this relationship is reversed when it comes to developing countries. What that shows is that in emerging economies, labour market conditions have a more direct influence on the stock markets, while in advanced economies, it is mostly financial markets that influence employment.

Additionally, Gonzalo and Taamouti (2017) examine the effect of anticipated versus unanticipated unemployment rates on stock prices. For the study, they found that only anticipated unemployment significantly contributed to changes in stock prices. An increase in the anticipated rate of unemployment witnessed an increase in stock prices mainly in the 0.35 to 0.80 quantile range. This has been attributed to Federal Reserve actions since with higher unemployment, interest rates were usually lower thus driving up the prices of the stock. However, unforeseen changes in unemployment have no great impact on returns. Their analysis stresses nonlinearity and a quantile-specific relationship between unemployment and the performance of the stock market.

2.2.5 Foreign Exchange Rates

Foreign exchange rates affect multinational firms by determining the exports' level of competitiveness and the import prices they would go for. Their changes, which is subject to supply and demand, may, therefore, have ramifications on companies' earnings and, consequently their stock prices also, Dornbusch & Fischer, 1980. Moore and Wang (2014) investigate the relationship between exchange rates and stock prices using monthly stock prices and real exchange rates vis-à-vis the US dollar for six Asian emerging markets and four developed countries from the 1970s/1980s to 2006. By using a DCC GARCH model, they established that there exists a negative significant correlation between exchange rates and the differentials of stock price, where the depreciation of a currency was followed by a decline in stock price. This, of course, was especially evident with the 1997 Asian financial crisis, which underlined the implications of financial instability on the two markets. Composite Indicators and Multivariate Analysis

Since the influences on stock prices are multifaceted, more than one economic indicator joined together can help in improving the predictive models. Chen et al. (1986) developed a multifactor model that incorporates industrial production, inflation, and interest rates. They were able to show, using U.S. monthly data from 1953 through 1983, that such factors together provide a rather substantial explanation of variations in returns. The notion was that stock returns were systematically related to changes in economic variables.

2.2.6 Implications for the Current Study

Literature has underlined a big potential factor that affects the economy, which in turn affects stock markets. The incorporation of a comprehensive set of interest rates, inflation, GDP, money supply, and exchange rates that are essential for predictive models will probably capture the finer dynamics of and FTSE 100 index and may have enhanced predictive accuracy.

Thirdly, the dissimilarity in the impact of different countries requires necessary specifications of models concerning the different economic structures of both the U.S. and the U.K., which would involve monetary policies, market structures and international trade relationships. In this respect, by way of example, the vulnerability of the FTSE 100 towards changes in exchange rate due to its international exposure, and fluctuations in currency need to be carefully modelled within the context of the U.K.

2.3 Convolutional Neural Networks in Stock Price Prediction

2.3.1 CNN Architecture for Time-Series Data

Convolutional Neural Networks, a highly trending deep learning architecture since its invention, has seen considerable development and found broad applications in many pattern recognition-related fields such as natural language processing, image analysis, and voice recognition (Albawi et al. 2017). While originally developed for image processing, CNNs have also been used in the analysis of time series by forming one-dimensional arrays of temporal data. More precisely, CNN models can capture spatial hierarchies due to convolutional layers. For time-series applications, the application of convolutional filters over sequential data considers the detection of the local pattern, hence the model extracts features invariant to temporal shifts, which makes them suitable for forecasting (LeCun et al., 2015). The CNN architecture typically includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to extract local features, pooling layers reduce dimensionality by downsampling, and fully connected layers integrate the extracted features for prediction (Cong & Zhou, 2022). This hierarchical structure allows CNNs to capture both low-level and high-level temporal features, making them suitable for modeling complex financial time-series data.

2.3.2 Empirical Studies on CNN in Finance

Long et al. (2019) then proposed a CNN model focusing on extracting features from the stock data for 1-minute intervals comprising prices and volume through convolutional layers to scan for short-term patterns within it. This CNN takes in 120-minute window inputs of financial indicators (like open, close, high, and low prices, and volume); it applies temporal filters to catch temporal dependencies. The best prediction accuracy reached 49.22%, and strong profitability was reflected in market simulations, with an overall return of 20.50% and Sharpe ratio at 3.36. Meanwhile, all other traditional models, including LSTM and SVM, were outperformed in both accuracy and risk-adjusted return by the best model in this work. Selvin et al. (2017) applied CNN for stock price prediction using minute-wise stock data from NSE of India. This was done in a sliding window fashion where each window consisted of a sequence of stock prices over a certain period, and within these windows, it had effectively captured local patterns, hence showing its strength in the tasks of short-term prediction.
This is further reinforced in the study done by Chen et al. (2020), where they utilized the Shanghai and Shenzhen 300 stock index futures minute data between September 30, 2017 and June 30, 2018 coupled with the RSI, BIAS, KDJ, and RSI technical indexes to test CNN. They found that the accuracy of the CNN model was 58% but the cumulative yield was 11.2% meaning that the model was generating profit against the futures. This study advocates that the CNN model can be used for stock prediction to receive earnings by predicting the stock market.

Furthermore, Hoseinzade & Haratizadeh (2019) used a wide range of variables within their CNN model, ranging from index-specific variables to economic indicators. They have found that with the extensive list of variables, the performance of the CNN substantially improves when tested on 5 different indices for the years 2010-2017 (S&P 500, DJI, NASDAQ, NYSE, RUSSELL). They argue that the CNN model should be further studied as their generated results have shown to beat the market. In the study conducted by Cao & Wang (2019), the five indices (HIS, TSEC, DAX, NASDAQ, S&P500) have been chosen due to the fact that they represent different market from Asia, Europe and the Americas. They evaluated the performance of the CNN model using the root MSE, correlation coefficient, and determinant coefficient. After tinkering with the model configurations, the best results have shown that CNN is incredibly efficient in predicting the stock prices. Moreover, they have indicated that if the model is paired with another deep learning algorithm it could generate better results.

2.4 Long Short-Term Memory Networks (LSTM) in Stock Price Prediction

2.4.1 LSTM Architecture and Time-Series Forecasting

Long Short-Term Memory networks (LSTMs) are a specialized form of Recurrent Neural Networks (RNNs) designed to capture long-term dependencies in sequential data (Hochreiter & Schmidhuber, 1997). Traditional RNNs suffer from the vanishing gradient problem, making them ineffective at learning long-term dependencies. LSTMs address this issue through memory cells and gating mechanisms (input, output, and forget gates) that regulate the flow of information. Just like the CNN, LSTM has also been improved over the course of 25 years prominently known for its exceptional results in the fields where the data is sequential such as speech-to-text transcription, language modelling, and machine translations (Sherstinsky, 2020).

In financial time-series forecasting, LSTMs are able to model complex temporal relationships, such as trends and seasonality by keeping relevant historical information across very long periods. Such an ability of LSTMs in remembering patterns in very long sequences provides suitability for the stock price prediction problem, where past market behaviors could have influences on the future movements.

2.4.2 Empirical Research on LSTM in Finance

Fischer and Krauss (2018) applied LSTM networks to predict stock returns for constituents of the S&P 500 index. Using daily data from 1992 to 2015, the authors have trained LSTM models to predict the next day's return given the previous 10 days of returns. In turn, this allows the LSTM models to outperform other traditional models such as logistic regression and random forests by an average excess return of 0.46% per day, or an overall Sharpe ratio of 5.8 after transaction costs. Their findings proved that LSTM can grasp the most rewarding temporal dependencies to produce trading strategies. In Nelson et al. (2017), LSTMs had been used to make predictions on the movement of the stock market based on historical price data like open, close, high, low prices, and trading volume instead of technical indicators like moving averages or RSI. This dataset consisted of a 15-minute interval of the Brazilian stock market represented by the Bovespa index from 2008 to 2015. The LSTM model reached an accuracy of up to 55.9%, outperforming the other traditional machine learning models, such as multilayer perceptron and random forests. Again, interestingly, this gain confirmed that LSTMs have a lot of potential for short-term financial prediction with especial temporal dependencies.

Pang et al. (2018) conducted a study where the performance of LSTM with an embedded layer and LSTM with an automated encoder were evaluated using the daily prices of Shanghai A-share composite index across 10 years. The findings showed that the accuracy was 57% which is considerably better than the stochastic method. Building on that, Rundo et al. (2019) used the historical data of banking stocks and corporates shares listed in the Milano Stock Exchange Market. The used framework consisting of two LSTM networks has been constructed: one for predicting trends and another for regressing the time-series of stock close prices. The initial implementation of the LSTM network is employed for predicting stock trends. It consists of an input layer that receives one stock close price value at a time, a hidden layer with 400 neurons, and a fully connected output layer with a single neuron. The second layer of LSTMs exhibits a similar structure, although with distinct configuration parameters and training updates. The reported accuracy ranged from 50%-90% and the model consistently generated profits. Additionally, Ghosh et al. (2019) had a different approach to testing the LSTM model. They have conducted a study on the Bombay Stock Exchange choosing different banking stocks and sectors comprising the biggest companies and testing them using daily prices in the span of 1,3,6 month and 1,3 years. The results have shown that with the increased time frame the error values have significantly decreased and provide exceptional results at range of less than 1 which was equivalent to 99% accuracy. Furthermore, (Ding & Qin, 2019) conducted a study where the LSTM models with different parameter settings were tested on the Shanghai Composite Index, PetroChina, and ZTE. The study shows that the best-performing LSTM model reached outstanding results, where it had an accuracy of the predicted value at 95%. This reinforces that the LSTM model when tested with different settings and on different stocks could have profound results.

2.5 Synthesis and Research Gaps

2.5.1 Integration of Economic Indicators

While several works incorporate technical indicators into deep learning models, fewer works incorporate macroeconomic indicators. Therefore, the incorporation of interest rates, inflation, GDP, unemployment rate, and exchange rates into deep learning models remains relatively unexplored, especially for hybrid models. This can also be an opportunity that may bring performance up by capturing the influence of fundamental economic factors in stock prices. Also, we have incorporated macroeconomic data in the paper because most stock price fluctuations are determined or caused by the economic conditions and by that way, we can check if models perform better in predicting the same.

2.5.2 Market Conditions

Few studies have tried to check or see the performance of the models in variable market conditions, like bull or bear markets or when the market volatility was high. One of the major points concerning real-world applicability is model robustness in conditions of variability. The models developed on data from stable market periods would not generalize during a crisis. Our aim is to test these models under unstable market conditions.

2.5.3 Data Limitations and Overfitting

Overfitting is an essential problem that arises in financial modeling, especially when limited data is involved. Deep learning models, particularly those with complex complexity-increasing area, are always prone to overfitting. This gets worse when several variables are included but with limited observations. The problem of overfitting may be soled by a number of techniques, including cross-validation, regularization, and model simplification when necessary.

2.6 Summary

The review has underlined that the most relevant economic indicators have the capability to influence the stock market and the potential of deep learning models, especially CNN and LSTM networks and their hybrid forms, in predicting stock prices. Incorporation of comprehensive sets of economic indicators into predictive models may strengthen the capability of the models in capturing the multifaceted influences on stock prices, with a view to improving forecasting accuracy.

Both models have strengths regarding the modeling of financial data: CNN is effective in capturing local temporal patterns and complex nonlinear relationships, while LSTM has huge potential in modeling long-term dependencies and temporal dynamics. The hybrid models that combine both architectures have promising potential to leverage these strengths further in improving predictive performance. This can help investors and analysts decide on how to use deep learning for stock price prediction. It further demonstrates the existing literature gap by embedding the macroeconomic variables into deep learning models, analyzing them in terms of performance under different market regimes.

Chapter 3 Methodology

3.1 Overview

This work adopts two potent deep learning architectures: CNN and LSTM for stock price prediction. While both models are fine-tuned for the task of capturing temporal relationships inherent in time series data, they address this problem using different approaches. CNN is particularly excellent at recognizing localized patterns within the data over multiple scales, while LSTM is exceptional at modelling long-term dependencies by keeping internal memory states. Despite these important differences, both models have a common structure through the use of dropout regularization, dense layers, hyperparameter optimization, and time series cross-validation. This section is going to discuss how each of the models is developed, the architectural differences setting them apart, the methods to prevent overfitting and improve generalization, and the metrics used for evaluation.

3.2 Data Collection and Pre-processing

In view of the deep learning techniques under consideration, full data pertaining to historical performance of stock markets and macroeconomic indicators will be needed. This involves collecting daily data for the FTSE 100 index ranging from 2013 to 2023. After gathering the data, there were 2778 days in the FTSE 100 index. The discrepancy between the days of the year and the market days can be explained by the fact that the market is closed during weekends and public holidays in different geographical locations.

This will provide insight into the recent historical context of the markets, ensure a considerable dataset is available, and test how well the models perform during times of crisis, as was the case in the year 2020. It contains all the significant stock market variables that are really critical for understanding the moves in the market, namely, 'Open', 'Close', 'High', 'Low', and 'Volume'. These five basic variables provide a complete picture of daily oscillations in the stock markets. Moreover, we add five hand-picked macroeconomic indicators simply due to the fact that 'Inflation (CPI)', 'Interest Rate', 'GDP growth', 'Exchange Rate to USD', 'Exchange Rate to EUR’ and 'Unemployment Rate' show up very frequently in financial literature. Again, the only one updated on a daily basis is the Exchange Rate. Whereas the inflation rate, interest rate, and unemployment rate are revised on a monthly basis, the GDP growth gets updated quarterly. These features have been added so that a broader economic context in which changes in stock prices take place can be captured and the predictive power of the model over future behavior improves. Given the diversity of the data sources and the varying scales of the variables, preprocessing was a crucial step.

3.3 Normalization

To ensure consistency and improve the performance of the deep learning models, the data was normalized using the Min-Max Scaler. The Min-Max Scaler scales values in the range of 0 to 1, which helps to avoid domination by variables with larger magnitudes and accelerates model convergence during training. Min-Max scaling is extensively employed in tasks involving time series forecasting because it effectively maintains the relative distribution of values while normalizing the range, a factor that is especially significant for neural networks that exhibit sensitivity to the scale of input data (Patro & Sahu, 2015). Min-Max scaling normalized the input to the models, preserving the proportional relationship amongst data points so that no one feature predominantly influences model learning dynamics. First, the Min-Max scaler was fitted on the training data for both the feature set, X_train, and the target variable, y_train. This is a serious methodological decision because, if the whole dataset was scaled-including the test set-in that case, data leakage would occur; the information from the test set would bleed into the training process. To avoid that from happening, the scaler has been fitted only to the training data, and it also made sure that the test data were blind while scaling them. The same transformations were performed on the unseen test data after scaling on the training set. In this way, the models see training examples of the same type as those on which assessment will be performed.

3.4 Sliding Window

Coupled with normalization, the sliding window approach has been considered in preparing the data for time series analysis. Sliding windows are often used in financial time series prediction because they allow the model to incorporate historical data over a fixed number of time intervals, providing both contemporary and historical trends during model training (Selvin et al, 2017). A custom function was built to generate these sliding windows. This function processes the time series data and divides it into overlapping segments of defined length. In this example, a window size of 20 was chosen; therefore, the samples fed into the models will be 20 consecutive time steps in the time series.

Approaches with sliding windows, such as this one, work very well for models like CNN and LSTM, whose predictions rely on sequential patterns. In CNNs, being good at catching localized patterns in the data, these convolutional filters scan over these 20-step windows and detect short-term, or localized, temporal dependencies. Because of this, the CNN can consider the time series in digestible chunks, this permits the CNN to target the short-term fluctuations while still allowing the model to pick up larger trends that only materialize after some time has elapsed. The sliding window architecture complements the primary strength of CNNs in picking out localized features while ensuring that the model identifies the informative content from small-scale patterns as well as preserves the temporal ordering present in the data. Although CNNs are specifically designed to capture local features, the sliding window method would enable them to capture sufficient history to allow meaningful forecasting in stock prices.

For LSTMs, whose specific architecture lets them capture long-range dependencies, this approach of using the sliding window serves a different but equally important purpose. While LSTMs thrive on sequential data where long-range correlations exist, feeding the whole time series into an LSTM model may lead to computational inefficiencies and reduce generalization due to the volume of data. Sliding windows are used to feed the LSTM with smaller sequences of fixed length, which it can process more efficiently, while still being able to maintain its strengths regarding modeling long-term dependencies. In each window set, the LSTM is able to observe the relationships and patterns for 20-time steps, and its memory cells allow for the storage of information across successive windows. It strikes a balance between the LSTM model's capability of modeling long-term relationships while making practical demands on handling large datasets so that it trains with efficiency without losing the capability of learning from the distant past.

3.5 CNN Architecture

The architecture of the CNN model in this project consisted of three 1D convolutional layers, each followed by MaxPooling. Although they are traditionally designed to process images, the CNNs are practical in time series forecasting because of their convolutional layers, which may detect temporal patterns while applying filters, scanning through the input data. Each convolutional layer of this model carries out a filtering process on the input data to capture patterns across diverse scales. This multi-scale nature of pattern recognition forms the most relevant part for the problem of stock price forecasting, where both short-run fluctuations, like daily volatility, and long-run trends, like economic cycles, remain key drivers.

These filters within 1-D convolutional layers scan over the time series, finding critical patterns in moving averages or other sudden price movements. Such filters perform elementwise multiplications with the input data. Hence, the feature map is provided, highlighting some of the essential features in the timeseries. It is then fed through ReLU-an activation function that gives some degree of non-linearity to the model. This non-linearity is necessary for the identification of complex, non-linear relationships between input features.

Application to financial forecasting: Stock prices are supposed to be influenced by several factors, such as market sentiments, macroeconomic events, and unexpected shocks.

The CNN model has filter size ranging from 32 to 256, while the kernel size either takes the value of 2 or 3. This set of hyperparameters enables the model to capture small but important changes in the time series and preserve the temporal dimension for longer periods. This also makes use of 'same' padding to ensure that the dimensions of the output are equivalent to the inputs, ensuring that no loss in temporal resolution occurs. That could mask a very significant historical trend in the stock price. This is further supported in the literature by Borovykh et al. (2017), showing that CNNs with small kernel sizes and appropriate padding are particularly effective for time-series analysis, such as financial forecasting.

For each convolutional layer, there is an accompanying MaxPooling to downsample the data. MaxPooling works to decrease the dimensionality of feature maps by taking the maximum value across sub-regions within the map. By doing this, the model's computational complexity decreases and adds a touch of translation invariance such that the CNN focuses on the strong patterns of large price movements or a sudden increase in volatility while ignoring irrelevant fluctuations or noise. Liu and Zhao (2022) that CNNs with moderate downsampling have superior generalization performance while still preserving substantial capacity to capture significant temporal features. Therefore using a relatively small pooling size, for instance, 2, the model does not go to the extreme in down-sampling, it retains enough information to make its predictions successfully.

After the convolution and pooling steps, the data is flattened out into a 1D vector and passed through a fully connected dense layer, which synthesizes lower-level patterns, as detected by the convolutional layers with higher-level patterns to arrive at a comprehensive understanding of the time series. The dense layer forms the decision-making layer for the model because it is at this layer that the aggregation is used to make the future prediction of stock prices.

3.6 LSTM Architecture

The LSTM model is fundamentally different from CNNs, as it is designed to model long-range dependencies in sequential data. The LSTM Network or Long Short Term Memory Network is a special type of RNN which can memorize information for very long periods using memory cells and gate mechanisms. Every LSTM cell contains an input gate, forget gate, and output gate that would help in regulating the flow of information. While the input gate controls what new information gets added to the memory, the forget gate decides what old information is to be discarded, and the output gate determines what part of the memory gets carried over to the next time step. This makes LSTMs particularly suitable for time series forecasting tasks, where often past events can have a long-lasting influence on the future outcome, as is the case with stock prices (Fischer & Krauss, 2018).

In this model, the first LSTM layer includes a tunable number of units ranging from 32 to 256, optimized using Keras Tuner. That permits the model to learn more complex forms of dependency in the time series but at greater computational cost and higher risk of overfitting. Finally, the LSTM cell uses a tanh activation function when learning nonlinear relationships in the input data. Hochreiter and Schmidhuber (1997) first proposed the architecture of the LSTM network using tanh as its internal activation, and indeed for the most part, LSTM has performed really well on various tasks on time series because it is able to keep gradients steady over long sequences.

Perhaps the most interesting feature of the LSTM model is returning the sequence hyperparameter, depending on whether one wants the layer to return the full sequence or just the last time step. If return_sequences=True, the entirety of the sequence is propagated to the next layer so that the model can capture detailed temporal relationships. This could prove very useful when stacking several LSTM layers, as this technique would allow the model to catch high- and low-level features across the time series. Gers et al. (2000) demonstrated that in complex time series tasks, the performance can be enhanced through stacking LSTM layers, which lets the network learn the hierarchical representation of data.

3.7 CNN-LSTM Architecture

After finding the best models in both CNN and LSTM, we will build out CNN-LSTM. The hybrid CNN-LSTM model combines the strengths of both CNNs and LSTMs to capture spatial patterns and temporal dependencies in the stock price data. This architecture first applies convolutional layers to extract local features from the input sequences and then uses LSTM layers to model the temporal relationships among these features. By integrating the convolutional and recurrent layers, the hybrid model can extract meaningful features from the input data and use them to make more accurate predictions. The use of ReLU and tanh activation functions provide the necessary non-linearity to model complex relationships within the data.

3.8 Dropout Regularization and Dense Layers

Both CNN and LSTM models use Dropout layers to regularize them from overfitting. In neural networks, regularization via dropout is a technique in which, during training, random fractions of neurons are dropped out that are not used. This has the effect of forcing the model to rely on a more distributed representation rather than relying overly on any single neuron. Dropout rates between 0.3 and 0.5 are tuned for the two models-a proper balance between regularization and model capacity. Dropout is an important element in financial forecasting; noisy data and market conditions that are hard to anticipate would eventually make the model too specialized to the training set and produce overfitting. Srivastava et al. (2014) demonstrated that dropout improves the generalization of deep learning models for tasks dealing with noisy datasets, such as those used for stock price prediction.

After the feature extraction layers, be they convolutional or LSTM, a Dense layer in both models serves to synthesize the learned features into a form the model can use to make a prediction. These dense layers utilize the ReLU activation function, allowing the model to learn non-linear relationships between the various features. In both models, the final output layer is a single neuron, which outputs a continuous value representing the predicted stock price for the next day. Using one neuron is in accordance with the common approach in regression tasks, where the task is to predict one real-valued output.

3.9 Hyperparameter Tuning using Keras Tuner

Both the CNN and LSTM models have been subjected to extensive hyperparameter tuning using Keras Tuner. Hyperparameter tuning is an indispensable exercise for any time series forecasting because the performance of deep learning models is grossly sensitive to the choice of hyperparameters. Key hyperparameters like numbers of filters, kernel sizes in CNN, and numbers of units in LSTM were optimized so that the models capture the right level of temporal detail without overfitting. In this respect, it has been demonstrated by Brownlee (2019) that the best performance for time series tasks can be achieved only when the convolutional filters and LSTM units are tuned to the optimum amount of model complexity, as overfitting can easily occur if the models become overly complex relative to the data. Other than optimizing the number of filters and units, the dropout rates and learning rates have also been tuned. Among these, one of the most essential factors that determines how fast the model converges during training is the learning rate. A very high learning rate will surely cause overshooting of the model beyond the optimum solution, while a low rate will make convergence slow and may not give the best performance. Learning rates were tuned for both models in the range of 0.0001 and 0.01 to ensure the convergence of the models efficiently without sacrificing accuracy. In both models, the Adam optimizer is used since it can dynamically adjust the learning rate during training and thus speed up convergence. Kingma and Ba (2014) have shown that the adaptive learning rate mechanism in Adam yields faster and more stable convergence compared to traditional gradient descent. Therefore, Adam finds more applications in deep learning.

3.10 Handling Temporal Data Cross-Validation

Since stock prices are inherently sequential, much emphasis was put on the conservation of temporal structure in this project. With the utilization of TimeSeriesSplit in both models for cross-validation, both models were trained and evaluated respecting the temporal order of the data. TimeSeriesSplit The function makes sure that during every fold of the cross-validation, only previous data is used for training while future data serves for validation; hence, it emulates a real-world task of stock price forecasting based on historical trends. It avoids one common problem in time series forecasting-data leakage-where future data inadvertently creeps into the training dataset. We set the number of splits to 5 meaning that the data will be split into 5 sequential training and validation sets. Since we are diving into windows of 20 days, that means it processes the an amount of windows set in the folds, in the first 2013-2015 to predict on 2016, second 2013-2016 to predict on 2017 and so on. This makes the validation more accurate as more data is processed everytime, while avoiding the data leakage.

Both models utilized early stopping to prevent overfitting during training. Early stopping keeps a tab on the validation loss and stops the training when the model has stopped improving. This would ensure that the models are not overfit to the training data. It is particularly a problem in financial forecasting, because market conditions may change so fast that models overfitting to historical data will fail to generalize into new market conditions. As Prechelt (1998) adds, the same applies to preventing overtraining in neural networks when working with noisy datasets, such as the stock prices.

3.10.1 Iterations

Iterations are a crucial part of training in both the CNN and LSTM models, enhancing generalization, stability, and optimization. Training the models with different iterations, with a split in the data, makes the models iteratively reduce prediction errors and enhance their capability of capturing complex patterns in the data. The advantage of that can be thought of this way: this will ensure that the model's performance in this repeated process is not dependent on a single run, hence mitigating the bias that can be caused by random initialization and specific subsets of the data. We set the number of iterations to 100.

3.11 Evaluation

Next, both models were evaluated in performance for key metrics with regard to accuracy, robustness, and generalization to unseen data. At the end of each iteration of training, summary statistics in the form of means, medians, and standard deviations are computed for each of the model evaluation metrics. These include: MAE, MSE, RMSE, R^2, MAPE, SMAPE, Explained Variance Score, MBD, Directional Accuracy, Accuracy within 1% tolerance, and Exact Accuracy (within 50 units). This metric set will give a full view of the quality of the predictions for the models on both training/validation and unseen test data.

The following are statistics calculated from both training and validation data. These give insight into how well the model learns and fits. For example, MAE, MSE, and RMSE give the estimation of error in absolute terms of prediction, whereas R^2 gives the proportion of explained variance from the model. More importantly, the most crucial measure Directional Accuracy is defined as the ability of the model to predict correctly the upside or downside of stock price in financial forecasting. Due to its high sensitivity to market shifts, Directional Accuracy was one key measure of performance showing how well the model captured the market trend.

Complementary to the evaluation of the training data, a similar analysis was performed for the unseen test dataset held out for testing model generalization. This allows for the comparison of metrics on training/validation data versus unseen data as one way to assess overfitting. More precisely, any increase in the RMSE or drop in R^2 on test data as compared to training data would have indicated overfitting. Put differently, the model would memorize the pattern in the training data instead of generalizing well. The MAPE and SMAPE gave error metrics in percentage form and complemented the model accuracy assessments even more, especially where the percentage deviation about actual values was more meaningful than the absolute errors.

Besides, graphical analysis was used in plotting the performance of the model across different iterations. Directional Accuracy and RMSE were plotted for both training/validation data and unseen data. Developing plots that show how accuracy changes with time helps in the identification of issues, which may include overfitting when the performance improves on the training data while deteriorating on unseen data. In the best case, the same RMSE and Directional Accuracy on both sets would have been indicative that the model was learning effectively without overfitting.

Another approach that could be followed to give a better view of the performance of the model on unseen data was taking the average of predictions over multiple iterations and plotting these against the actual stock prices. This allows for the smoothing of random

fluctuations in individual predictions and gives a better view of how well the model tracks the underlying trends in the stock prices. A plot was fitted comparing the averages of predicted prices against the actual unseen prices for a visual assessment of the accuracy of the model in real-world forecasting scenarios. This visualization easily allows for the identification of systematic prediction errors, such as consistent over- or under-prediction.
This section presents the in- results of CNN, LSTM, and combined CNN-LSTM models used in stock price prediction. All models were optimized regarding their hyperparameters and were retrained several times. Their hyperparameters had been found where the best performance was obtained. The performance was measured on both training/validation data and on previously unseen test data. Each of the CNN, LSTM, and CNN-LSTM models demonstrated very good performance concerning training and validation datasets, each one being very proficient in identifying strong patterns within the information of stock prices.

Chapter 4 Results

4.1 Model Performance

4.1.1 CNN

The best hyperparameters for the CNN model were as follows:

- First Layer: 32 filters, kernel size of 2
- Second Layer: 64 filters, kernel size of 2
- Third Layer: 256 filters, kernel size of 3
- Pooling Type: Global average pooling
- Dense Layer: 128 units
- Dropout Rate: 0.3
- Learning Rate: 0.001

These are the hyperparameters used for the CNN, reflecting that it is designed for the capturing of local patterns in the stock price data of the next day. Therefore, these kernel sizes (2, 3) will be relatively smaller, which helps the model focus on fine-grained temporal changes that could be crucial in giving the best prediction of stock prices based on a daily market trend. While going deeper into the network, increasing the number of filters in the deeper layers allows the model to extract more complex and abstract features from the input data. Furthermore, this is aided by the fact that a global average pooling layer will average the contribution of every neuron in the final convolutional layer, reducing overfitting. Moreover, a dropout rate of 0.3 has introduced regularization to prevent overfitting to this training data. Meanwhile, a learning rate of 0.001 tries to balance ensuring a model converges effectively during training and prevents overshooting into local minima on the loss.

The CNN model performed well during training. It yielded an average MAE of about 99.31 with a standard deviation of 16.03-a rather consistent range across the training iterations. The RMSE averaged 142.32, reflecting a good fit to the training data. It can be observed that the very high value of R-square, 0.9304, reflects that more than 93% of the variance of the stock price data has been explained by the model in training. It is also proved by MAPE and SMAPE that both are smaller than 1.5 percent, which suggests that the predicted prices were not more than 1.5 percent away from the actual on average. The model further returned a mean directional accuracy of 70.87%, which showed it could correctly forecast the direction of the stock price movement, which is usually the central issue in any investment. All of the metric results for train data in Table 1 in Appendix B.

However, he CNN model exhibited a widely deteriorated performance when exposed to data it had not seen before. The MAE went as high as an average of 436.21 with a high SD of 249.99, reflecting more variability in the prediction errors. The RMSE rose to 630.59 on average, and the R-squared value dropped to -2.0117, meaning that the model failed to generalize to unseen data. The MAPE increased from 1.49% during training to 5.86% on test data, while the SMAPE rose from 1.48% to 5.68%, which is 4% deviation that could prove considering the context of price prediction. These higher percentage errors reflect a decline in the model's relative accuracy when predicting the accurate prices of the unseen data. This is further reflected when examining the negative R-squared suggests that the model performed worse than a simple mean prediction on the test data, highlighting significant challenges in generalizing to new market conditions. Despite the decrease in performance, the directional accuracy remained relatively high at 67.25%, though slightly lower than during training, which indicates that the model despite being inaccurate in reflecting the prices has a considerable accuracy in predicting which direction the price is going to go the next day. he metric results for test data all in Table 2 in Appendix B. All the plots for the average predicted prices against the actual price and metrics results over iterations in Appendix B.

This decline in the test data can be explained by turmoil in the UK stock market during the test period influenced by fluctuating macroeconomic indicators post-COVID. Its focus on local patterns may not have made it adaptable to such large shifts, thus showing the problem of reliance on historical data when the market evolves dynamically. The increased MAE and RMSE, along with the negative R-squared, indicate that the model struggled to capture new patterns and volatility present in the unseen data.

4.1.2 LSTM

The following, on the other hand, are the best hyperparameters for the LSTM model:

- First LSTM Layer: 128 Units
- Return Sequence: False
- Dropout Rate, LSTM Layer: 0.4
- Dense Layer: 112 units
- Dropout Rate, Dense Layer: 0.4
- Learning Rate: 0.01

The selection of 128 units in the LSTM layer may indicate that this model required a pretty high number of units to learn from the long-term dependencies in the stock price data. Return sequence were False below reflects that this model has been designed to output one value per time step—in this case, the next day's stock price. Moreover, it indicates that the data the temporal dependencies can be captured using a simpler model. A daily stock price prediction design returning a full prediction sequence isn't needed when the interest is solely in the next price.

Also, the dropout in the LSTM and dense layer was increased to 0.4 for stronger regularization, to avoid overfitting; this is particularly important, considering both layers are relatively large, since larger networks are more at risk of overfitting. Here, the learning rate is higher than for the CNN model, 0.01, but this can reflect the need of faster convergence for the LSTM architecture, which sometimes needs more iterations to correctly adjust weights in an optimal way.

In the training phase, the LSTM model achieved an MAE of approximately 108.34 with a standard deviation of 10.41, indicating stable performance across iterations. The RMSE averaged 153.42, R-squared value was 0.9199, demonstrating that the model explained about 92% of the variance in the training data, MAPE and SMAPE also showed results of 1.6% also proving that the model is quite accurate in its predictions. The directional accuracy was 62.71%, slightly lower than the CNN model, suggesting that while the LSTM captured long-term trends, it was less adept at predicting short-term directional changes during training. The directional accuracy was 62.71%, slightly lower than the CNN model, suggesting that the LSTM was less adept at predicting short-term directional changes during training.

On unseen test data, the generalization of the LSTM model was better than that of the CNN. The MAE was lower, averaging 322.05 with a standard deviation of 182.53, while the RMSE averaged 471.81. Though the R-squared value remained negative at -0.8177, it was closer to zero than that derived from the CNN, hence implying a lesser degree of underperformance relative to the mean prediction. The test data had a directional accuracy of 57.81%, which is a challenge to predict the direction of the movement in stock price according to the volatility of the market.

This was most probably due to the ability of LSTM to model long-term dependencies in inputs, thus improving its performance on unseen data concerning MAE and RMSE. However, directional accuracy for LSTM is much poorer, supporting the fact that sudden market fluctuations regarding price directions cannot be modelled easily by this framework. The model is biased toward longer-term trends that might be too slow to catch sudden changes in the market. Additionally, even with the increased window size capturing the next day's price has shown to be less effective, which would mean that the LSTM had a poor generalisation even with the advantage of capturing long-term dependencies. This implies that the model was potentially overfitting.

4.1.3 CNN-LSTM

The CNN-LSTM model was supposed to combine the best of both the CNN and LSTM models with their respective best hyperparameters and further improve the predictability of the model. For that, the proposed hybrid model combined the convolutional layers of the CNN with the LSTM layer of the LSTM model. The resultant hyperparameters were as follows:

- Convolutional Layers:
  - First Layer: 32 filters, kernel size of 2
  - Second Layer: 64 filters, kernel size of 2
  - Third Layer: 256 filters, kernel size of 3
- LSTM Layer:
  - Units: 128
  - Return Sequences: False
- Dense Layer: 112 units, ReLU activation
- Dropout Rate: 0.4 (after LSTM and dense layer)
- Learning Rate: 0.01

The CNN-LSTM hybrid model is defined that emphasizes the strengths of CNN and LSTM model architectures that can learn both local patterns and long-term dependencies of stock prices. By blending convolutional layers with an LSTM layer, the model tended to result in better predictive capabilities. Regularization was achieved by dropout rates of 0.4, while the learning rate was set at 0.01 to facilitate quick and efficient training.

The CNN-LSTM model yielded an MAE of about 129.86 with a standard deviation of 29.70 during training, hence showing that this model is more volatile compared to each stand-alone model across iterations. The RMSE averaged at 187.81, while the value of R-squared was 0.8756; that is, about 87.56% of the variance in training data is explained by the model. In

particular, it achieved the highest directional accuracy in training of 71.55%, which reflected the goodness of performance on the direction of change in stock prices.
On the test data, the CNN-LSTM model maintained the high directional accuracy of 70.24%, well above that of both the CNN and LSTM models. While doing this, though, its MAE increased to 437.81, with the RMSE rising to 615.54, thus indicating that it turned out to be good at predicting the direction of price movements but could not provide very good predictions of exact price levels on unseen data. The R-squared further decreased to -1.6889, reflecting inability to capture the variability within new market conditions. This can partly be explained by the complexity of this model and overfitting towards the training data. However, its consistency in its directional accuracy really underlines the value of such a hybrid in aggregating local and long-term pattern recognition for volatility trend prediction.

4.2 The Dataset

In order to find out what could have been a potential cause for these models to do well on training data, but failing to reproduce the same results on the test data we delve into the market indicators that could have been drastically changed as hypothesized above. We have created charts for all of the market indicators for training and test data to see the difference between them graphically and analyse the difference. The graphs are all shown in Appendix B.

As observed from the graphs, in this trend of the training data, the inflation rate was changing over the years-first, a decrease from 3% to 0%, followed by an increase again to 3%. In the test data, we observe, first of all, a difference in scale between the training and test periods, with such a high increase in the inflation upward trend-from 0% to 10%-after which it gradually decreases to 4%. This constitutes a critical observation because the test data has a radically higher inflation rate compared with conditions to which the models were not fully adapted. Similarly, in the training data, the interest rate undergoes minor time variations, stabilizing at around 0.5%, peaking at 0.7%, and then falling to 0.1% afterwards. In contrast, the interest rate in the test data is considerably higher and increases systematically from 0% to

5.25%. This consequently causes a huge gap between the macro conditions in the training and test periods, posing a big challenge for model generalization.

The GDP growth rate starts with a really stable upward trend from 2013 to 2020, then suddenly has a massive drop of -20%, immediately followed by an increase of 17%. This kind of sudden movement at the far end of the training data is likely to impact performance. Conversely, in the test set, it happened in reverse: slight fluctuations in the previously stable sub-zero rate. For the exchange rate, the first rate, EUR/GBP, starts at 1.20 for the training data, increases up to 1.45, and then bounces around 1.12 with a difference of about 0.8. The test data follows this trend. On the other hand, the rate for USD/GBP starts out quite well at 1.6 for the training data but gradually shrinks toward 1.3 with minor fluctuations of 0.1. In the test data, the trend continues for a while before sharply plummeting to 1.05 and then steadily rising again to 1.3.The unemployment rate, in both the training and test data, follows a similar trajectory—decreasing initially, but rising slightly toward the end.

These trends are important because the models learn from a combination of these macroeconomic indicators. Even though we are using the sliding window approach—where the models use the last 20 days to predict the opening price of the next day—they inherently learn from the broader trends observed during training. Our approach thus captures both long-term trends, as the models learn from these windows of data, and short-term patterns, which help provide more accurate next-day predictions.

However, the difference in macroeconomic conditions between the training and test data—particularly the sharp rise in inflation and interest rates in the test period—can lead to overfitting. In this case, the models may have overfit the specific patterns in the training data, especially during periods of volatility. This can cause the models to struggle when exposed to new, unseen data with different trends. This is evident when looking at the graph (combined averages), where the price predictions follow the trend of the test data well at the beginning but fall out when the indicators have drastically changed which implies that the model is unfit to generalize in such cases.

4.3 Summarize

The models have performed well during the training and validation phases of the project with strong metrics on MAE, RMSE, and directional accuracy. Given that these models are optimized on their hyperparameters, such a result is usually expected. However, the large drop in performance on unseen data might also be indicative of overfitting, where the models have learned models that are overly specific to the training data and have failed to generalize well to new volatile market conditions. In general, poor generalization across all models suggests that they struggled to adapt to the evolving economic trends in the UK market during the test period. This is an unprecedented time for UK policymakers as such conditions have not been met before, showing the originality of the use of deep learning models in the stock market as we test on the extraordinary economic conditions in the UK.

Chapter 5 Discussion

5.1 Overview

This project aimed to evaluate the predictive capabilities of three distinct deep learning models—CNN, LSTM, and CNN-LSTM—in forecasting daily stock prices, with the inclusion of a variety of macroeconomic indicators such as inflation, interest rates, GDP growth, and exchange rates. By application in both training and validation against unseen test data, we are able to demonstrate the efficacy in the capture of sophisticated market dynamics with a review of their limitations when it comes to volatile market conditions.

The central objective of this project was therefore to identify which model may serve most appropriately as a means for the daily stock price predictions on FTSE 100, based on the same macroeconomic factors described above. Thus, our analysis came to focus on which model showed the highest accuracy in prediction at generalization to new, unseen data. The models' performance was matched against their ability to minimize error metrics such as MAE, RMSE, MAPE, SMAPE, additional metrics that look at the variance and precision, and also to capture the directional trends in stock price movement, which is a very critical feature for decision-making in financial markets.

5.2 Performance of the Models

5.2.1 Performance of CNN

It should be mentioned that the CNN model had a very high R-squared value, amounting to 0.9304 with low MAE of 99.31 during training. This is reasonable and aligns with the previous studies by Long et al. (2019) and Chen et al. (2020), considering the power of CNN in learning locally restricted patterns in short-term time-series data. Thus, this ability to capture local features made CNN well-suited for detecting small temporal fluctuations, an aspect considered germane in the daily forecasting of stock prices. However, when the market conditions changed during the test period, especially post-COVID volatility, the generalization capability of the CNN was seriously affected and it resulted in an R-squared value of -2.0117. The drop of the R-squared here indicates that CNN had a very poor performance in modeling the new volatility of the market-a common issue when these deep learning models are trained on relatively stable historical data and employed for volatile real-world conditions (Hoseinzade & Haratizadeh, 2019).

5.2.2 LSTM Performance

On the other hand, it could be depicted that the LSTM model performed well in generalizing on unseen data. This is probably because of its architectural nature, which efficiently allowed it to learn temporal dependencies for larger horizons. Thus, this model turns out to be better in terms of MAE (322.05) and RMSE (471.81) with respect to the test data compared to CNN. This agrees with Fischer & Krauss (2018) who concluded that the ability of LSTM to the long-sequence memory rendered it robust to dynamic and changing market conditions. With that said, however, the directional accuracy was a bit lower for the LSTM model, 57.81%, compared to the CNN. It thus meant that though it was better in terms of prediction of absolute price levels, it showed poor predictive capabilities regarding the sudden changes within the direction of the stock price movement limitation earlier noted by researchers in financial time series forecasting.

Most likely, the volatility of the macroeconomic indicators within the test period-especially the interest rates and inflation-posed problems for the LSTM. Even with the advantage of modelling long-term dependencies, the rapid post-COVID changes of these indicators likely outpaced the ability of the model to adapt since its training data did not take into account such dramatic changes. This reflects a broader limitation of LSTM models in financial forecasting: in as much as they are able to grasp long-term trends, they might be too slow to adapt to sharp, unpredictable changes in market dynamics (Pang et al., 2018).

5.2.3 CNN-LSTM Performance

The CNN-LSTM hybrid model was developed so as to leverage the strengths of the CNN in capturing short-term patterns and the power of the LSTM in modelling long-range dependencies. This model, during training, demonstrated the best directional accuracy, 71.55%, of the three, showing that the hybrid was able to capture local fluctuations and market-wide trends more effectively by combining the two architectures. It did considerably well on the direction of price movements-70.24% directional accuracy. However, it only had an MAE of 437.81 and an RMSE of 615.54 on test data.

This is consistent with Cao & Wang (2019), who, while observing that hybrid models with more complexity compared to standalone models are better at learning complex temporal relationships, may consequently overfit to volatile market conditions. More precisely, the CNN-LSTM performed comparatively poorly on test data, indicating overfitting to the patterns in the training data and generalization performance in different macroeconomic conditions. This became especially prominent in the post-COVID period when the inflation and interest rates surged to values never faced by the model during training.

5.3 Influence of Macroeconomic Indicators and Market Volatility

The overall fluctuations in the macroeconomic indicators, especially those related to inflation and interest rate, during the test period, hit deep into the results which were produced by all three models. The inflation rate increased from about 0% to over 10%, and the interest rate was supposedly stuck at about 0.5% and increased to 5.25%. With these changes, along with a seesawing GDP growth rate and exchange rate, came new complexities the models had not seen in training.

This is in line with findings that explain how stock prices are strongly influenced by macroeconomic variables, especially during periods of economic turmoil. Similarly, Fama (1981) and Ioannidis & Kontonikas (2008) indicated that inflationary pressures and interest rates affect the stock market, as it was the case in our test data likely introducing new correlations between those signals and stock prices that models could not fully capture as they had been trained only on pre-pandemic data. This suggests that during this period, the models lack the generalization capability and hence point to the limitations in using historical data for financial market forecasting in fast-changing market conditions.

Besides, the oscillations of economic indicators during the test period show the rough sides of deep learning models for stock price prediction in real market conditions. While the model performed very well during training, the inability of the models to generalize on new market conditions may indicate that the models over-relied on the historical pattern, which became less relevant as the market evolved. In this case, sudden inflation and interest rate increase during the test period most probably served to overfit the models to the training data. The CNN, LSTM, and CNN-LSTM models could be fit to find out the relationship between macroeconomic indicators and stock prices in a stable environment. However, sudden changes in such macroeconomic indicators during the test period presented conditions that were not present during the training of the models. The intuition is that the deep learning models, while very powerful, may be quite difficult to adapt to new market conditions, especially in cases in which those conditions have to do with rapid shifts in key macroeconomic indicators. This accords with what was seen in the literature.

5.4 Confidence in Results

These are robust results, considering the amount of hyperparameter tuning and time series cross-validation applied to prevent overfitting. However, the performance of models on test data reflects the usual failure of deep learning models in highly volatile markets. Even though the model performed well on training data, inability of a good generalization suggests overfitting might occur during training for the case of the CNN-LSTM model.

However, the sliding window approach may not have been ideal in capturing long-term trends covering several hundreds of days. Therefore, this might be one point of improvement, since an increase in the size of the window-or the addition of further layers to the LSTM model-may resolve this issue to some extent. Besides, the choice of the macroeconomic indicators, though comprehensive, needs further refinement for better capturing the nuances of the test period in light of large changes in inflation and interest rates.

5.5 Implications and Scope

The key findings from the study go to the very core of deep learning model applications in stock price prediction. Although the CNN, LSTM, and CNN-LSTM models are very powerful in making good predictions when market conditions are stable, there is still a need to find the ideal solution that could handle the volatility of the market, which is indicative of their challenging performance. Therefore, although deep learning models can be a good tool for financial forecasting, tuning up and careful evaluation in light of changing market conditions will have to be considered.

Including further macroeconomic indicators or using adaptive learning techniques, like reinforcement learning, may help the model generalize better. Therefore, ensemble methods-where multiple models' predictions are combined into one final prediction-would be of use when considering overfitting, since it puts together strengths and mutes weakness.

Chapter 6 Evaluation, Reflections and Conclusion

6.1 Overview

This project, therefore, estimated the performance of three deep learning models regarding prediction, using a comprehensive dataset of macroeconomic indicators along with stock market data. The in-depth literature review, implementation of advanced machine learning techniques, and point-by-point evaluation of the results throughout this project have presented a number of insights into how deep learning models can perform in such a complex and volatile environment as finance. The key findings from this chapter will be reflected upon, decisions made will be critically evaluated, and areas for improvement highlighted. Possible future research and application directions are considered.

6.2 Evaluation of Project Objectives

The primary objective of the project was to determine which deep learning model best suits the prediction of daily stock prices for both the FTSE 100, integrating a range of macroeconomic indicators. The results clearly show the strengths of each model in the process. The CNN model was relatively good at finding the local patterns, while the LSTM model was better at representing the long-term dependency of the data. The most complex, hybrid CNN-LSTM had the best directional accuracy in defining the overall trend of the stock price movements.

Objectives set out for this project were largely achieved, while the most important one was that of demonstrating how these models perform under different market conditions. No model was able to generalize well when facing such dramatic shifts in macroeconomic indicators, especially those that pertain to inflation and interest rates, as analyzed by the unseen test data representative of a volatile post-COVID market. This casts serious doubt on the suitability of traditional deep learning models for real-world financial forecasting tasks where abrupt changes may come at any time without prior warnings.

Project in the following has identified the potential and limitations of the CNN, LSTM, and hybrid CNN-LSTM models to predict stock prices. On the other hand, it has also pointed out the shortcomings of these models to generalize well in a new, unseen condition of the market; hence, calling for further refinement in the strategies of modeling or adoption of other methods against overfitting to enhance generalization.

6.3 Reflections on the Literature and Methodology

The literature review thus set a wide backdrop for the current study, underlining the emerging interest in the application of deep learning models in financial time series forecasting. Whereas CNNs have proven their capabilities to detect short-term patterns, LSTM networks can grasp long-term dependencies. Therefore, the idea of joining these strengths in one hybrid CNN-LSTM model, as mentioned in recent studies, seemed the most promising.

While the literature provided general support for the approach adopted in this project, some methodological choices made a difference in results that was not well anticipated. The sliding window approach was quite effective to train models on sequential data, allowing the models to capture sequences of 20 days for the next day price prediction. However, the size of the window itself may have been too small to capture broader trends, especially in light of large macroeconomic changes such as inflation spikes. Expanding the window size or adopting less rigid methods, like dynamic windowing, could have captured both short- and longer-term trends by these models.

Also, although theoretically justified, the set of macroeconomic indicators of interest rates, inflation, GDP growth, exchange rates, and unemployment rates may be extended further, possibly by adding market sentiment measures or volatility measures-such as VIX-which would provide more complete information for the model about the state of the market, especially in unstable periods.

6.4 Reflection and Learning

Considering the perspective of the whole project, there are a couple of lessons that appear to be quite clear. What is most crucial is the generalization of models. Whereas deep learning models are performing marvellously on the training dataset, there should be a guarantee that their performance will be preserved when new, unseen data comes in financial forecasting tasks. This project has shown just how challenging it may be to apply the deep learning model to volatile markets and pointed out further refinement areas, in particular concerning regularization and handling macroeconomic shocks, still being at hand.

This further reinforced the concept that the most important concept in any model evaluation is the hyperparameter tuning. We achieved very strong performances when training all models with great care for tuning the parameters of each model. The test data results bring forth the point that even the best-tuned models are prone to performances when exposed to conditions outside the envelope with which they have been trained, and as such, other techniques would be necessary for robust models, like ensemble learning or reinforcement learning.

Last but not least, the project has indicated the inclusion of various macroeconomic indicators into the financial forecasting models. This fact was very helpful in putting the models into context but simultaneously meant dealing for the first time with the problem of modeling dynamic dependencies among those indicators and stock prices. Further work may well investigate more advanced methods, such as attention mechanisms, that enable the model to focus its attention on the most relevant indicators at any particular time.

6.5 Future Work

It is recommended to add more indicators to the model. Expanding the present set of macroeconomic indicators by market sentiment indices, volatility measures such as the VIX, or sector-specific data may provide a more accurate and fine-grained detail on the stock price movements. As a result, the model will be able to perform much better, especially during highly volatile periods when the additional indicators capture the critical factors driving market behavior.

In this respect, ensemble methods can be useful. In such a framework, several combined models can overcome overfitting that has taken place in this project. To leverage the strength of different models; for example, the capacity of CNN to capture the pattern in the short term, and LSTM to keep the memory for a long time. Ensemble methods result in robust, reliable, more certain predictions.

It is further recommended that adaptive learning techniques be studied. The adaptation methods will have the ability to get improved by reinforcement learning, which allows the models to adapt to the evolution of market conditions. These would significantly limit the problems arising from sudden changes in macroeconomic indicators and improve the generalization capability of models across diverse market scenarios.

Lastly, the work should be oriented toward model interpretability. Most of the deep learning models were normally of very high predictive accuracy, yet their mode of operation-"black box"-may turn out to be one of the limiting factors in a number of fields, including financial ones, where at least some kind of transparency and interpretability are integral. Attention mechanisms or explainable AI (XAI) techniques may be used in future work, enhancing the interpretability of models while guaranteeing more clarity about how the model will make predictions and makes its output understandable to users.

References

Apergis, N. and Eleftheriou, S. (2002) ‘Interest rates, inflation, and stock prices: The case of the Athens Stock Exchange’, *Journal of Policy Modeling*, 24(3), pp. 231–236. doi:10.1016/s0161-8938(02)00105-9.

Atsalakis, G.S. and Valavanis, K.P. (2009) ‘Surveying stock market forecasting techniques – part II: Soft computing methods’, *Expert Systems with Applications*, 36(3), pp. 5932–5941. doi:10.1016/j.eswa.2008.07.006.

BERNANKE, B.S. and KUTTNER, K.N. (2005) ‘What explains the stock market’s reaction to Federal Reserve Policy?’, *The Journal of Finance*, 60(3), pp. 1221–1257. doi:10.1111/j.1540-6261.2005.00760.x.

Borovykh, A., Bohte, S. and Oosterlee, C.W. (2018) ‘Dilated convolutional neural networks for time series forecasting’, *Journal of Computational Finance* [Preprint]. doi:10.21314/jcf.2019.358.

Cong, S. and Zhou, Y. (2022) ‘A review of convolutional neural network architectures and their optimizations’, *Artificial Intelligence Review*, 56(3), pp. 1905–1969. doi:10.1007/s10462-022-10213-5.

Dornbusch, R. and Fischer, S. (no date) *Exchange Rates and the Current Account*. Available at: http://www.mit.edu/~14.54/handouts/dornbusch80.pdf (Accessed: 10 October 2024).

Fischer, T. and Krauss, C. (2018) ‘Deep learning with long short-term memory networks for financial market predictions’, *European Journal of Operational Research*, 270(2), pp. 654–669. doi:10.1016/j.ejor.2017.11.054.

Hochreiter, S. and Schmidhuber, J. (1997) ‘Long short-term memory’, *Neural Computation*, 9(8), pp. 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Humpe, A. and Macmillan, P. (2007) ‘Can macroeconomic variables explain long term stock market movements? A comparison of the US and Japan’, *SSRN Electronic Journal* [Preprint]. doi:10.2139/ssrn.1026219.

Ioannidis, C. and Kontonikas, A. (2008) ‘The impact of monetary policy on stock prices’, *Journal of Policy Modeling*, 30(1), pp. 33–53. doi:10.1016/j.jpolmod.2007.06.015.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) ‘Deep learning’, *Nature*, 521(7553), pp. 436–444. doi:10.1038/nature14539.

Long, W., Lu, Z. and Cui, L. (2019) ‘Deep learning-based feature engineering for stock price movement prediction’, *Knowledge-Based Systems*, 164, pp. 163–173. doi:10.1016/j.knosys.2018.10.034.

Moore, T. and Wang, P. (2014) ‘Dynamic linkage between real exchange rates and stock prices: Evidence from developed and emerging Asian markets’, *International Review of Economics & Finance*, 29, pp. 1–11. doi:10.1016/j.iref.2013.02.004.

Nelson, D.M., Pereira, A.C. and de Oliveira, R.A. (2017) ‘Stock market’s price movement prediction with

LSTM neural networks', *2017 International Joint Conference on Neural Networks (IJCNN)* [Preprint]. doi:10.1109/ijcnn.2017.7966019.

Rapach, D. and Zhou, G. (2012) *Forecasting stock returns* . Available at: http://apps.olin.wustl.edu/faculty/zhou/Rapach_Zhou_HB_2013_WP.pdf (Accessed: 16 October 2024).

Chen, N., Roll, R. and Ross, S.A., 1986. Economic forces and the stock market. *The Journal of Business*, 59(3), pp.383-403. Available at: http://www.jstor.org/stable/2352710 [Accessed 16 October 2024].

Fama, E.F., 1981. Stock returns, real activity, inflation, and money. *The American Economic Review*, 71(4), pp.545-565. Available at: http://www.jstor.org/stable/1806180 [Accessed 16 October 2024].

Geske, R. and Roll, R., 1983. The fiscal and monetary linkage between stock returns and inflation. *Journal of Finance*, 38(1), pp.1-33.

Srivastava, N. *et al.* (2014) *Dropout: A simple way to prevent neural networks from overfitting*. Available at: https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf (Accessed: 16 October 2024).

Liu, J. and Zhao, Y. (2022) 'Improved generalization performance of convolutional neural networks with Lossda', *Applied Intelligence*, 53(11), pp. 13852–13866. doi:10.1007/s10489-022-04208-6.

Patro, G.K. and Sahu, K.K. (2015) *Normalization: A Preprocessing Stage*. Available at: https://arxiv.org/abs/1503.06462 (Accessed: 10 October 2024).

MAGHAYEREH, A. (2003) 'Causal relations among stock prices and macroeconomic variables in the small, open economy of Jordan', *Journal of King Abdulaziz University-Economics and Administration*, 17(2), pp. 3–12. doi:10.4197/eco.17-2.1.

Kyereboah-Coleman, A. and Agyire-Tettey, K.F. (2008) 'Impact of macroeconomic indicators on stock market performance', *The Journal of Risk Finance*, 9(4), pp. 365–378. doi:10.1108/15265940810895025.

Reddy, L. (2012) *Impact of inflatation and GDP on stock market ...* Available at: https://garph.co.uk/IJARMSS/Dec2012/9.pdf (Accessed: 16 October 2024).

Pan, W.-F. (2018) 'Does the stock market really cause unemployment? A cross-country analysis', *The North American Journal of Economics and Finance*, 44, pp. 34–43. doi:10.1016/j.najef.2017.11.002.

Gonzalo, J. and Taamouti, A. (2017) 'The reaction of stock market returns to unemployment', *Studies in Nonlinear Dynamics & Econometrics*, 21(4). doi:10.1515/snde-2015-0078.

Chen, C. *et al.* (2020) 'Financial quantitative investment using convolutional neural network and Deep Learning Technology', *Neurocomputing*, 390, pp. 384–390. doi:10.1016/j.neucom.2019.09.092.

Hoseinzade, E. and Haratizadeh, S. (2019) ‘CNNpred: CNN-based stock market prediction using a diverse set of variables’, *Expert Systems with Applications*, 129, pp. 273–285. doi:10.1016/j.eswa.2019.03.029.

Cao, J. and Wang, J. (2019) ‘Stock price forecasting model based on modified convolution neural network and financial time series analysis’, *International Journal of Communication Systems*, 32(12). doi:10.1002/dac.3987.

Sherstinsky, A. (2020) ‘Fundamentals of Recurrent Neural Network (RNN) and long short-term memory (LSTM) network’, *Physica D: Nonlinear Phenomena*, 404, p. 132306. doi:10.1016/j.physd.2019.132306.

Pang, X. *et al.* (2018) ‘An innovative neural network approach for stock market prediction’, *The Journal of Supercomputing*, 76(3), pp. 2098–2118. doi:10.1007/s11227-017-2228-y.

Rundo, F. *et al.* (2019) ‘Advanced markov-based machine learning framework for making Adaptive Trading System’, *Computation*, 7(1), p. 4. doi:10.3390/computation7010004.

Ding, G. and Qin, L. (2019) ‘Study on the prediction of stock price based on the Associated Network Model of LSTM’, *International Journal of Machine Learning and Cybernetics*, 11(6), pp. 1307–1317. doi:10.1007/s13042-019-01041-1.

Ghosh, A. *et al.* (2019) ‘Stock price prediction using LSTM on Indian share market’, *EPiC Series in Computing* [Preprint]. doi:10.29007/qgcz.

Selvin, S. *et al.* (2017) ‘Stock price prediction using LSTM, RNN and CNN-sliding window model’, *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* [Preprint].

Gers, F.A., Schmidhuber, J. and Cummins, F. (2000) ‘Learning to forget: Continual prediction with LSTM’, *Neural Computation*, 12(10), pp. 2451–2471. doi:10.1162/089976600300015015.

Kingma, D.P. and Ba, J. (2017) *Adam: A method for stochastic optimization*, *arXiv.org*. Available at: https://arxiv.org/abs/1412.6980 (Accessed: 16 October 2024).

Prechelt, L. (1998) ‘Early stopping - but when?’, *Lecture Notes in Computer Science*, pp. 55–69. doi:10.1007/3-540-49430-8_3.

Brownlee, J. (2019) *CNN long short-term memory networks*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/cnn-long-short-term-memory-networks/ (Accessed: 16 October 2024).

Data compiled from:

Official Bank Rate History (no date) *Bank of England*. Available at: https://www.bankofengland.co.uk/boeapps/database/Bank-Rate.asp (Accessed: 16 October 2024).

team, G.D.P. (2024) *Gross domestic product: Quarter on quarter growth: CVM SA %*, *Gross Domestic Product: Quarter on Quarter growth: CVM SA % - Office for National Statistics*. Available at: https://www.ons.gov.uk/economy/grossdomesticproductgdp/timeseries/ihyq/pn2 (Accessed: 16 October 2024).

Published by D. Clark and 18, S. (2024) *UK inflation rate 2024*, *Statista*. Available at: https://www.statista.com/statistics/306648/inflation-rate-consumer-price-index-cpi-united-kingdom-uk/ (Accessed: 16 October 2024).

Published by Statista Research Department and 10, S. (2024) *UK unemployment rate 2024*, *Statista*. Available at: https://www.statista.com/statistics/279898/unemployment-rate-in-the-united-kingdom-uk/ (Accessed: 16 October 2024).

GBP to USD exchange rate: British pound sterling to US dollar (no date) *Investing.com UK*. Available at: https://uk.investing.com/currencies/gbp-usd (Accessed: 16 October 2024).

GBP EUR historical data: Exchange rate history (no date) *Investing.com UK*. Available at: https://uk.investing.com/currencies/gbp-eur-historical-data (Accessed: 16 October 2024).

Stooq (no date) *UKX - FTSE100*. Available at: https://stooq.com/ (Accessed: 16 October 2024).

Appendix A Original project proposal

1. Introduction

1.1 Purpose and Objectives

The stock market is a dynamic and complex system dependent on multiple factors, economic indicators, trading volumes, global events, and political situations. Accurate prediction of stock prices is essential for making rational investment decisions in a timely manner for investors and financial institutions in order to maximize returns on their portfolios. The main aim of this project is to test the effectiveness of deep learning models—Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM), and an integrated CNN-LSTM model—on stock price prediction. This research aims to establish which of these models provides accurate and reliable forecasts of stock prices by applying the models in the financial indices, namely the S&P 500 and the FTSE 100 from the start of 2022 to the end of 2023, with relevant economic indicators. The purpose of using two different indices is to test the generalisability and test in different conditions as the two markets are separate with their own market conditions and regulation. The importance of this project is to understand the relevant factors and to be able to informatively understand the models. Although, they are not independent of each other.

The primary objectives are:

1. Develop CNN, LSTM, and CNN-LSTM models using historical data from these indices.
2. Compare and evaluate their performance.
3. Identify which of these models is the best and evaluate why it is the case.
4. Measure the effectiveness based not only on the precise accuracy but also measure the accuracy of the predictions in a particular range.

1.2 Products and Beneficiaries

The output that is to be expected from this project:

1. A dataset of the indices with tailored macroeconomic indicators developed through the literature.
2. Deep learning models for stock price forecasting.
3. A comparative analysis of the performance of these models through visualization methods and tables.

The primary beneficiaries of this project are professionals from the financial industry: financial analysts, traders, portfolio managers, risk managers, financial engineers, etc. Moreover, data scientists who have aspirations of joining a financial market as this project is intended to also deliver an understanding of economics concepts.

2. Critical Context

2.1 Research question and scope

In this project, we ultimately want to find out which Deep Learning approach has the best performance across different evaluation metrics. While addressing this question, we also address questions regarding the selection of indicators, apart from the baseline features of stock opening and closing prices and trade volume, that should be incorporated into our models to obtain a comprehensive understanding of the market and performance of these models under different market conditions. This research focuses on analyzing data from two prominent financial indices: the S&P 500 and FTSE 100 with macroeconomic indicators. Both indices consist of the aggregate prices of the largest corporations in the United States and the United Kingdom, respectively.

2.2 Literature Review

2.2.1 Economic indicators

The stock market of a particular country is closely intertwined with the various economic indicators that signal the health of the trajectory of the economy. Participants in the stock market firmly examine economic indicators to make an informed decision. Thus, there is a need to explore the effects of economic indicators on the stock market. (Kyereboah-Coleman & Agyire-Tettey, 2008) have conducted a study to find how the macroeconomic indicators affect the Ghana Stock Exchange. They have found that the lending rate is a strong indicator of the direction of the stock market as the policy changes significantly affect the businesses in Ghana. Moreover, inflation was also found to have a strong impact on the stock market performance, as they are determinants of money flow, but noted that the effect is lagged. Additionally, in the study conducted by Maghayereh (2003) on the Jordanian Stock Exchange between the period 1987-2000, interest rate and inflation have been found to have a notable relationship with the long-term stock price index. In another comparative study of the effects of the macroeconomic indicators on the stock market between US and Japan conducted by (Humpe & Macmillan, 2009), they found that the interest rates and inflation have affected both the stock markets. However, they note that not all the tested indicators affected the markets to same degree. Moreover, money supply had no effect on the US market and a sizeable effect on the Japanese market. This highlights the importance of using multiple indicators as some indicators might not be as effective as others. This is further highlighted by findings from conducted by Gan et al. (2006) about the relationship between the indicators and the New Zealand Stock Exchange. It was found that on top of the already mentioned interest rate and money supply, real GDP has been found to have a noteworthy effect on the performance of the New Zealand stock market. Moreover, they have concluded that inflation rate and exchange rate do not affect the results as much as it could be from the more sizeable markets like in the US, Korea, and Japan.

2.2.2 CNN

The Convolutional Neural Network (CNN), which is highly popular, has undergone significant advancements since its inception and is extensively utilized in various domains related to pattern recognition, including natural language processing, image analysis, and voice recognition (Albawi et al., 2017). Due to this, researchers have started utilizing CNN in various fields due to its exceptional pattern identification capabilities. One of the researches undertaken by Selvin et al. (2017) employed CNN to predict stock prices by analyzing minute-wise stock data from the National Stock Exchange (NSE) of India. The CNN model was applied using a sliding window approach, where each window contained a sequence of stock prices over a specified period, where it had effectively captured local patterns within these windows, demonstrating its strength in short-term prediction tasks. This is further reinforced in the study done by Chen et al. (2020), where they utilized the Shanghai and Shenzhen 300 stock index futures minute data between September 30, 2017 and June 30, 2018 coupled with the RSI, BIAS, KDJ, and RSI technical indexes to test CNN. They found that the accuracy of the CNN model was 58% but the cumulative yield was 11.2% meaning that the model was generating profit against the futures. This study advocates that the CNN model can be used for stock prediction to receive earnings by predicting the stock market. Furthermore, Hoseinzade & Haratizadeh (2019) used a wide range of variables within their CNN model, ranging from index-specific variables to economic indicators. They have found that with the extensive list of variables, the performance of the CNN substantially improves when tested on 5 different indices for the years 2010-2017 (S&P 500, DJI, NASDAQ, NYSE, RUSSELL). They argue that the CNN model should be further studied as their generated results have shown to beat the market. In the study conducted by Cao & Wang (2019), the five indices (HIS, TSEC, DAX, NASDAQ, S&P500) have been chosen due to the fact that they represent different market from Asia, Europe and the Americas. They evaluated the performance of the CNN model using the root MSE, correlation coefficient, and determinant coefficient. After tinkering with the model configurations, the best results have shown that CNN is incredibly efficient in predicting the stock prices. Moreover, they have indicated that if the model is paired with another deep learning algorithm it could generate better results.

2.2.3 LSTM

Just like the CNN, LSTM has also been improved over the course of 25 years prominently known for its exceptional results in the fields where the data is sequential such as speech-to-text transcription, language modelling, and machine translations (Sherstinsky, 2020). The excellent results in the field of sequential data make it an attractive method for application especially for stock prediction data where it is consistent of sequential pricing. In light of that, Pang et al. (2018) conducted a study where the performance of LSTM with an embedded layer and LSTM with an automated encoder were evaluated using the daily prices of Shanghai A-share composite index across 10 years. The findings showed that the accuracy was 57% which is considerably better than the stochastic method. Building on that, Rundo et al. (2019) used the historical data of banking stocks and corporates shares listed in the Milano Stock Exchange Market. The used framework consisting of two LSTM networks has been constructed: one for predicting trends and another for regressing the time-series of stock close prices. The initial implementation of the LSTM network is employed for predicting stock trends. It consists of an input layer that receives one stock close price value at a time, a hidden layer with 400 neurons, and a fully connected output layer with a single neuron. The second layer of LSTMs exhibits a similar structure, although with distinct configuration parameters and training updates. The reported accuracy ranged from 50%-90% and the model consistently generated profits. Additionally, Ghosh et al. (2019) had a different approach to testing the LSTM model. They have conducted a study on the Bombay Stock Exchange choosing different banking stocks and sectors comprising the biggest companies and testing them using daily prices in the span of 1,3,6 month and 1,3 years. The results have shown that with the increased time frame the error values have significantly decreased and provide exceptional results at range of less than 1 which was equivalent to 99% accuracy. Furthermore, (Ding & Qin, 2019) conducted a study where the LSTM models with different parameter settings were tested on the Shanghai Composite Index, PetroChina, and ZTE. The study shows that the best-performing LSTM model reached outstanding results, where it had an accuracy of the predicted value at 95%. This reinforces that the LSTM model when tested with different settings and on different stocks could have profound results.

3. Approaches

3.1 Data

The dataset for the daily stock opening and closing prices and volume will be obtained from Yahoo Finance and the macroeconomic indicators from the government resources. The closing price will serve as the prime point as we will measure the difference between the closing price with the predicted price. We will normalize the data to ensure all features are on a similar scale. The macroeconomic indicators we have chosen so far are interest rates, inflation (CPI), and GDP growth. As I read more of the literature, I will be subsequently adding more indicators. The data for indicators will be gathered from the US Bureau of Economic Analysis (BEA), Federal Reserve, Office for National Statistics (ONS), and Bank of England.

3.2 Methods

Firstly, before starting working on building the models I will review the influential papers for the model that I will be using. This will ensure that the fundamental idea of the models will remain at the heart of these models. Later on, I will be going through the specific models that have been applied to my domain to see how the best-performing models have been built, test them within the dataset that I have, and adapt my model to be the best version that it could be from the available literature. The literature outlined above is the baseline that I am looking to apply, and as more papers are reviewed new useful material will subsequently implemented. So far, from the reviewed literature I have come up with the next plan to carry out my methodology. The libraries that I intend to use are TensorFlow, Keras, NumPy, Pandas, Scikit-learn, Matplotlib, and Seaborn.

3.2.1 CNN

Convolutional Neural Networks (CNN) are particularly effective with respect to the local patterns and short-term trends in the financial time series data. Their specialty in handling the task is that they are very capable of auto-extracting meaningful features from the raw data, which can model complex patterns as stated above. Thus, they become very useful in prediction for short-term tasks where identification and processing of local patterns are required to understand the recent trends.
The approach we will be utilizing for CNN:

- **Windowing**: The data will be segmented into fixed-size windows, the daily rates over the years 2022-2023. Each window will serve as a single input to the CNN, which captures the short-term trends within that period.

- **Reshaping Data**: The data will be reshaped into a 3D tensor format since the CNN works with 3D. This will allow the convolutional layers to process the temporal data effectively.

- **Model Architecture**: Apply multiple convolutional layers to extract local features from the data windows. To downsample the features maps will utilize the pooling layers to reduce the dimensionality and keep the important information. I will then, flatten the output and pass it through dense layers to make the final prediction.

- **Feature Integration**: As described, I will implement the macroeconomic indicators directly into the CNN model as additional features. These indicators provide additional information that improves the model’s understanding of market conditions.

However, the application of CNN for stock prediction is not without its limitations. CNNs mostly capture local dependencies and may not do well in modeling long-term dependencies or temporal relationships beyond the scope of the sliding window. That could lead to a limited understanding of longer-term market trends, which are often quite important for the accuracy of stock price predictions. Besides, CNNs may be quite computationally expensive, thus requiring a lot of resources for training and tuning, especially with our large datasets that include many features and indicators.

3.2.2 LSTM

Long Short-Term Memory Networks (LSTM) can handle sequential data and learn long-term dependencies. LSTMs will be particularly amenable to learning from past sequences and maintaining that information over long periods.

The approach we will be utilizing for LSTM:

- **Data Modification**: The data will be organized into sequences of daily closing stock prices to serve as an input. Each of the sequences will be of important use to predict the next time step.

- **Model Architecture**: Construct the LSTM model with layers that can capture temporal dependencies. The architecture involves one or more LSTM layers followed by dense layers to output the prediction.

- **Feature Integration**: The implementation of the macroeconomic indicators as additional features in each of the time steps in the sequence. This integration allows the model to account for broader economic factors influencing stock prices.

- **Regularization**: To prevent overfitting, apply regularization techniques such as dropout, L2 regularization, and early stopping.

However, applying CNNs to stock prediction is not free of limitations. Mainly, CNNs have the ability to catch local dependencies and fail in modelling long-term dependencies or temporal relationships that are out of scope for the sliding window. This may result in limited comprehension of the longer-term market trends, very decisive in the precision of stock price predictions. Also, the CNNs are relatively computationally expensive. For our data, involving a great number of features and indicators, the training and fine-tuning of the models require massive resources.

3.2.3 CNN-LSTM

I will take the best model configurations from each of the best-performing models and apply them to build a hybrid CNN-LSTM model. This model will be built with the idea that it will cover the shortcomings of each of the models. Different methods can be implemented in the CNN-LSTM model (Rahman, 2023) :

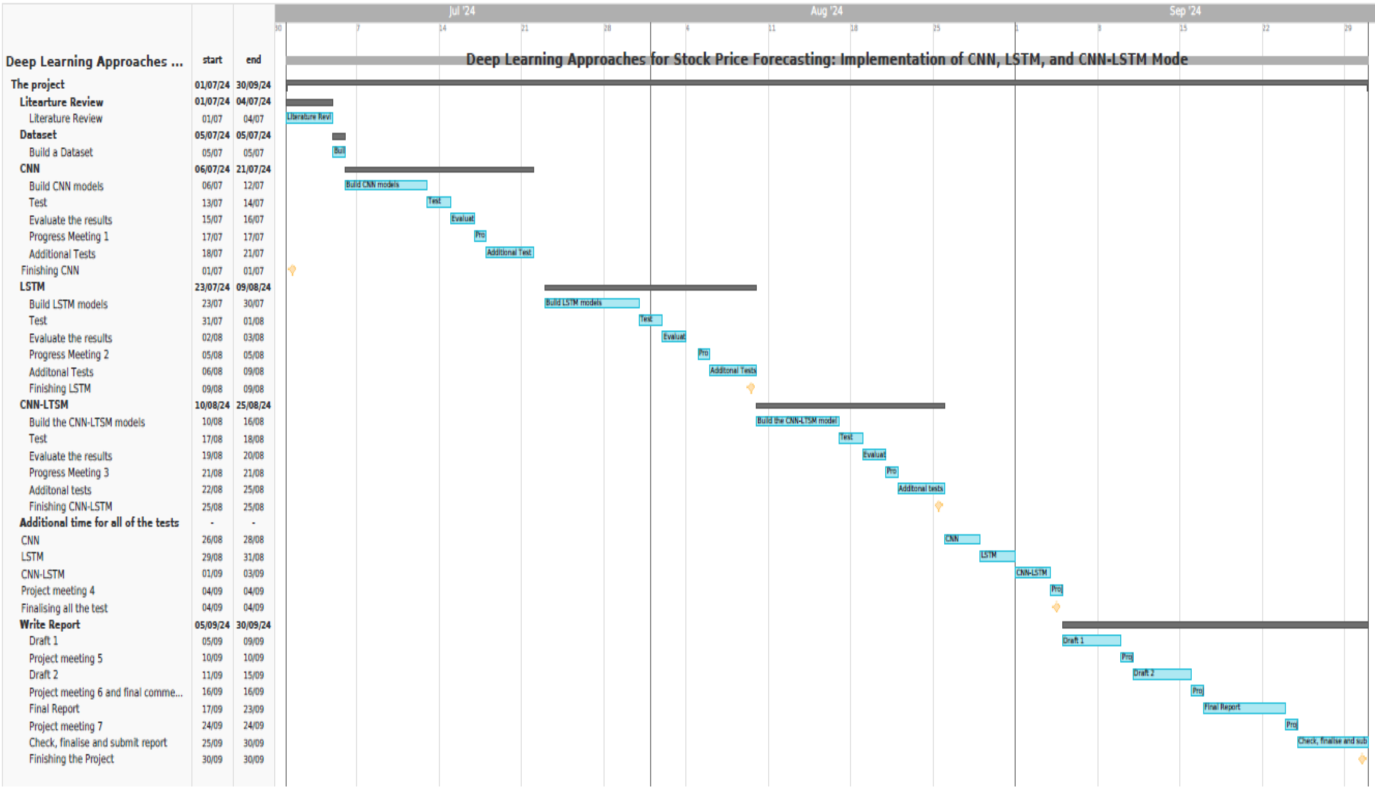
1. Utilise the output of the Convolutional Neural Network (CNN) as the input to the Long Short-Term Memory (LSTM) model. This enables the LSTM to acquire features from the input data that have been acquired by the CNN.
2. Utilise the output of the Long Short-Term Memory (LSTM) as the input to the Convolutional Neural Network (CNN). This enables the Convolutional Neural Network (CNN) to acquire features from the output of the Long Short-Term Memory (LSTM) model.

3. Employ a parallel architecture, wherein the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) function autonomously on the input data, and their outputs are merged and transmitted to the fully linked layer.

3.3 Evaluation

To ensure the generalization and reproducibility of my models, I will apply approaches such as partitioning the dataset into validation, training, and test sets. I will employ hyperparameter optimization to identify the optimal setup. In addition, we will employ k-fold cross-validation to assess the robustness of the model and evaluate it using metrics such as error percentage, RMSE, and primarily accuracy for each of the models. Moreover, we will be testing the variance of the predicted price and closing prices as another metric of evaluation. This is done to see if the predicted values are accurate within a range of values.

4. Work Plan



5. Risks

Description	Likelihood (1-3)	Consequence (1-5)	Impact (LxC)	Mitigation
Loss of the project	1	5	5	I will keep all of my progress on the data and the models on the One Drive to be able to

				store it securely.
Models take a long time to run	2	3	6	I will ask for the university resources as they provide computers that can generate results for time-consuming models quickly. Otherwise I will buy subscriptions to cloud services
Models are hard to build	2	5	10	I will be studying the models closely with online available resources and go through the learning them more in-depth in online classes. I already have the foundation from the course. I will also examine the codes available and adapt them.
Physical injury	1	4	4	I am a football player who enjoys playing a lot. This can results in an injury which could prevent me from doing my tasks. I will use all the necessary defensive equipment and do not engage in risky situations.
Mental health	1	5	5	Loss of motivation could majorly impact me and thus the project. I am very passionate about this project and unlikely to lose the motivation. I will be mindful of myself and take breaks.
Supervisor is no longer able to supervise the project	1	4	4	A new supervisor will be given by university. I will thoroughly write all the discussions to update a new supervisor
The models are not showing promising results	1	4	4	I will be using daily closing rates for the years 2022-2023. I will resort to using other data like hourly, monthly but shortening or imcreasing the time period.

6. References

1. Albawi, S., Mohammed, T.A. and Al-Zawi, S. (2017) 'Understanding of a convolutional neural network', *2017 International Conference on Engineering and Technology (ICET)* [Preprint]. doi:10.1109/icengtechnol.2017.8308186.
2. Cao, J. and Wang, J. (2019) 'Stock price forecasting model based on modified convolution neural network and financial time series analysis', *International Journal of Communication Systems*, 32(12). doi:10.1002/dac.3987.
3. Chen, C. *et al.* (2020) 'Financial quantitative investment using convolutional neural network and Deep Learning Technology', *Neurocomputing*, 390, pp. 384–390. doi:10.1016/j.neucom.2019.09.092.
4. Ding, G. and Qin, L. (2019) 'Study on the prediction of stock price based on the Associated Network Model of LSTM', *International Journal of Machine Learning and Cybernetics*, 11(6), pp. 1307–1317. doi:10.1007/s13042-019-01041-1.
5. Ghosh, A. *et al.* (2019) 'Stock price prediction using LSTM on Indian share market', *EPiC Series in Computing* [Preprint]. doi:10.29007/qgcz.
6. Hoseinzade, E. and Haratizadeh, S. (2019) 'CNNpred: CNN-based stock market prediction using a diverse set of variables', *Expert Systems with Applications*, 129, pp. 273–285. doi:10.1016/j.eswa.2019.03.029.
7. Humpe, A. and Macmillan, P. (2009) 'Can macroeconomic variables explain long-term stock market movements? A comparison of the US and Japan', *Applied Financial Economics*, 19(2), pp. 111–119. doi:10.1080/09603100701748956.

Kyereboah-Coleman, A. and Agyire-Tettey, K.F. (2008) 'Impact of macroeconomic

Appendix B

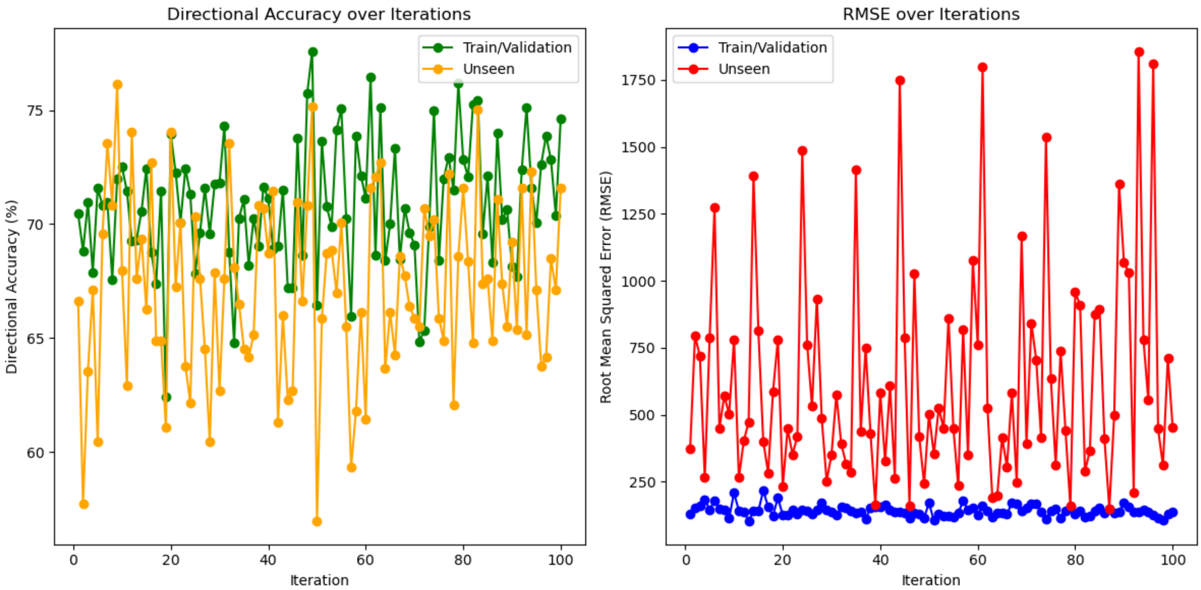


Figure 1 CNN

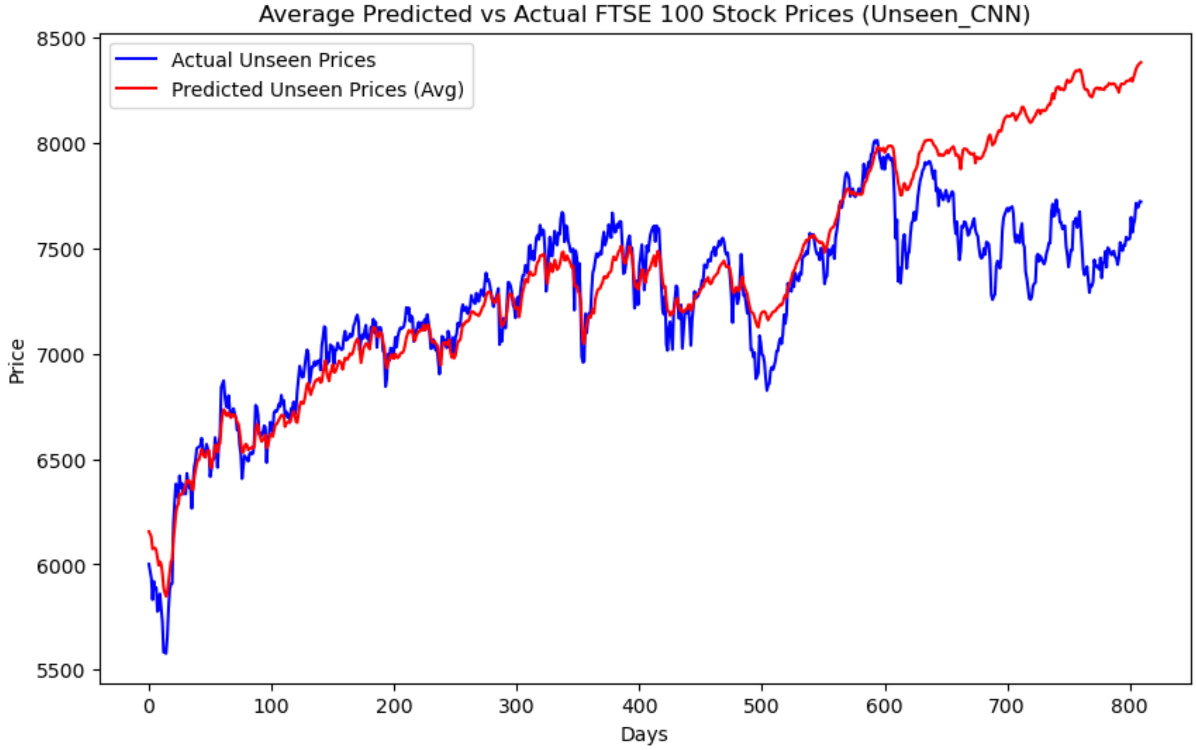


Figure 2 CNN

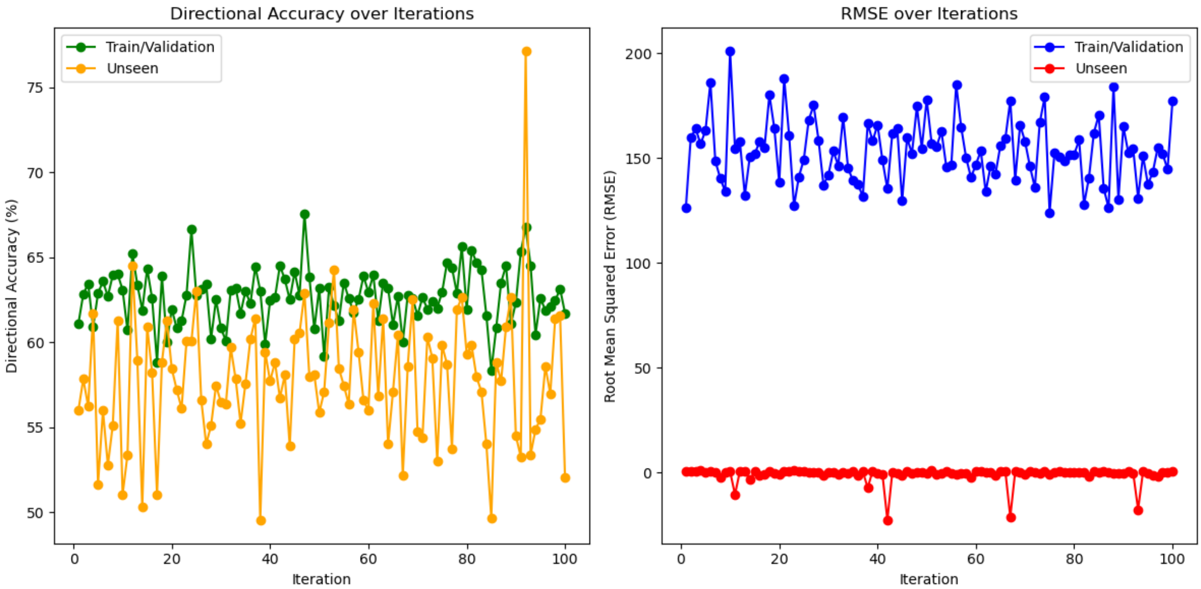


Figure 3 LSTM

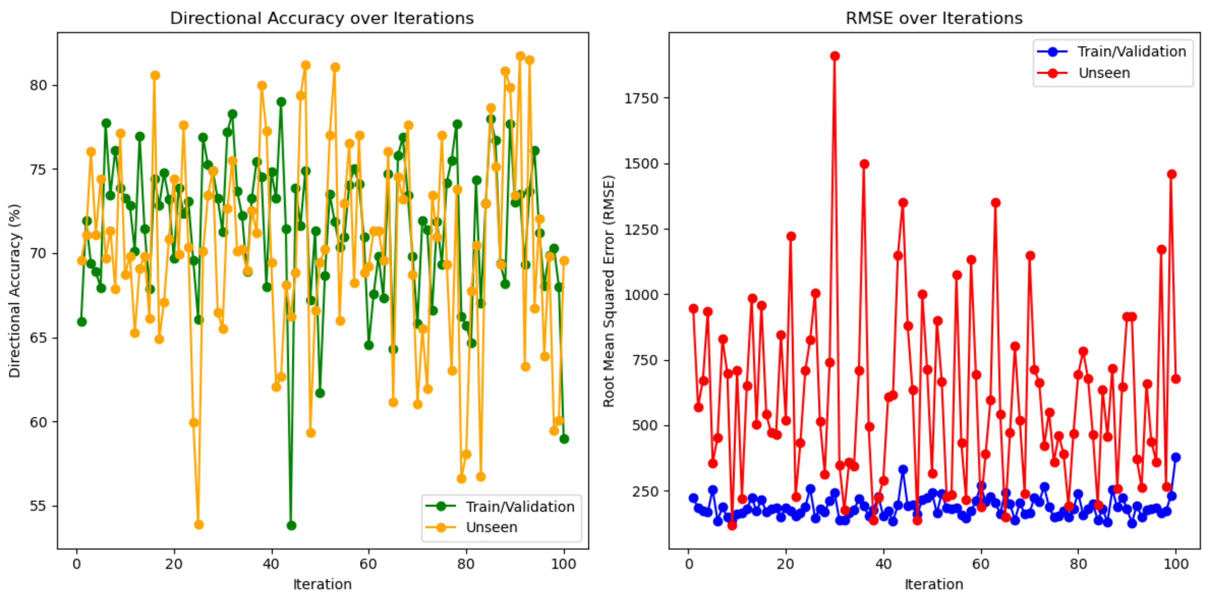


Figure 4 CNN-LSTM

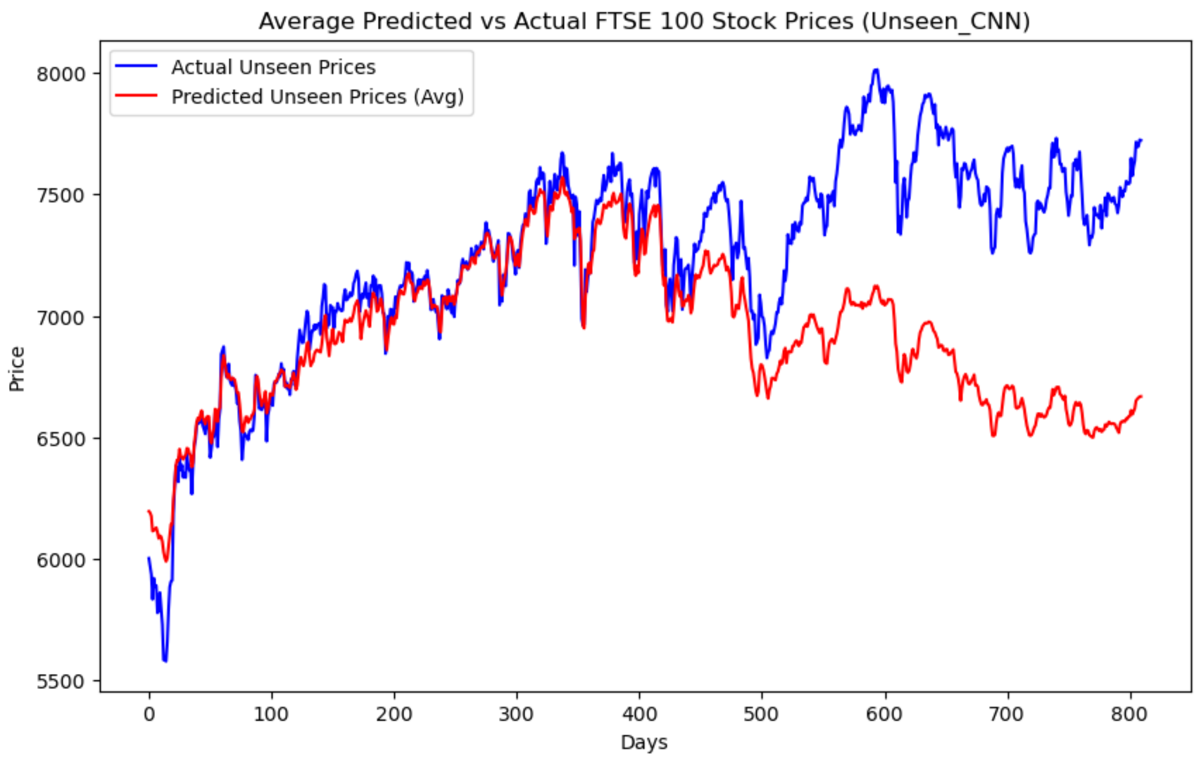


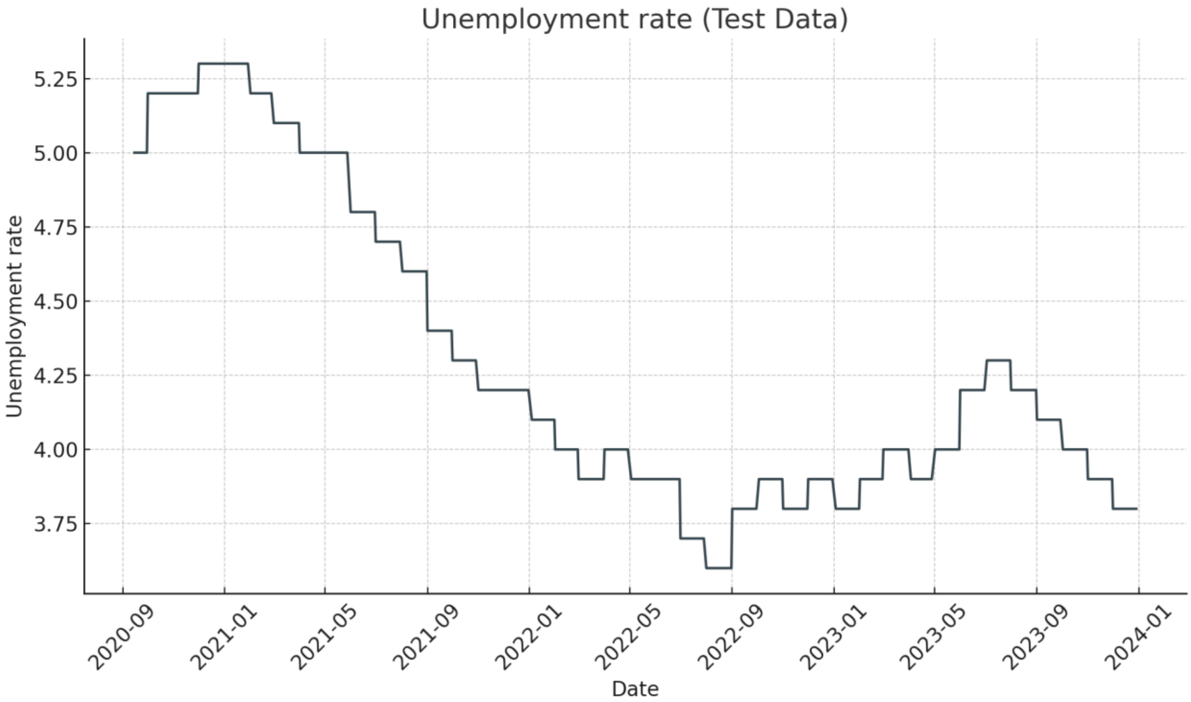
Figure 5 CNN-LSTM, please ignore the heading

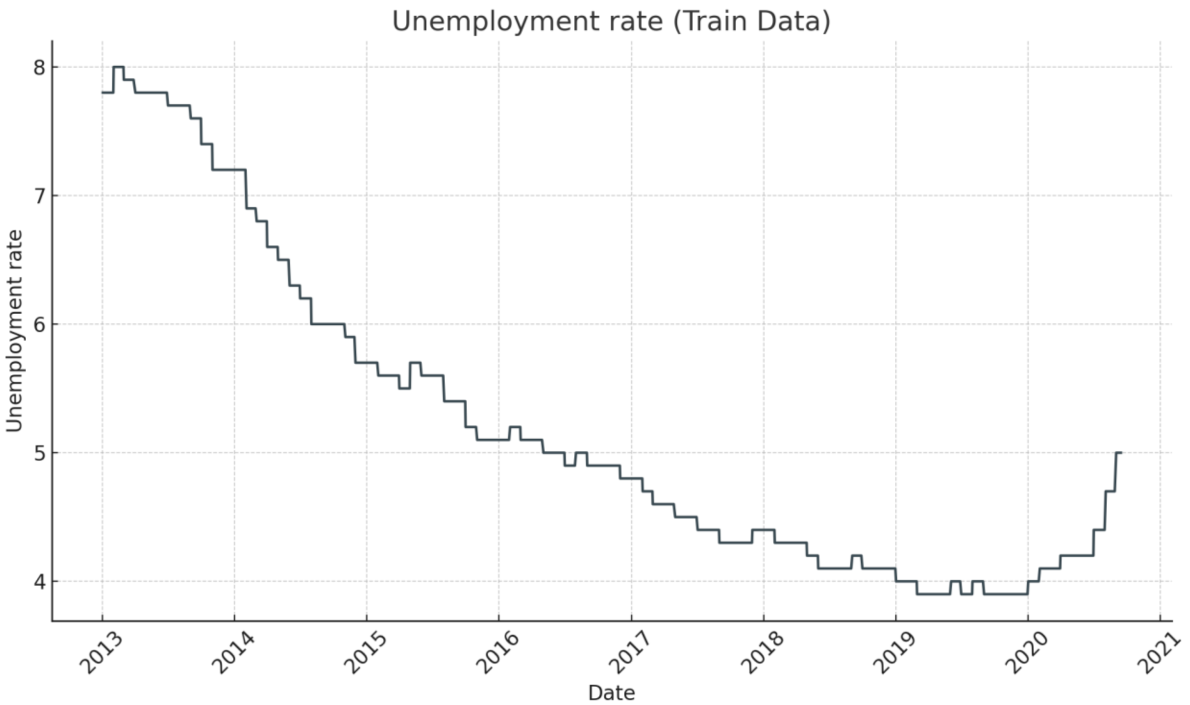
Metric	CNN			LSTM			CNN-LSTM		
	Mean	Median	Standard Deviation(SD)	Mean	Median	Standard Deviation(SD)	Mean	Median	Standard Deviation(SD)
Mean Absolute Error (MAE)	99.30	97.74	16.02	99.30	97.74	16.02	129.86	125.22	29.70
Mean Squared Error (MSE)	20,670	19880	6232	20, 670	19				
Root Mean Squared Error (RMSE)	142.31	140.99	20.3959						
R-squared (R^2)	0.9304	0.9331	0.0210						
Mean Absolute Percentage Error (MAPE)	1.49	1.47	0.23						
Symmetric Mean Absolute Percentage Error (SMAPE)	1.4843	1.4612	0.2310						
Explained Variance Score	0.9319	0.9357	0.0199						
Mean Bias Deviation (MBD)	3.2635	5.8834	21.0911						
Directional Accuracy	70.8678	70.8853	2.7655						
Accuracy within 1% tolerance	50.2530	50.7788	6.5546						
Exact Accuracy (within 50 units)									

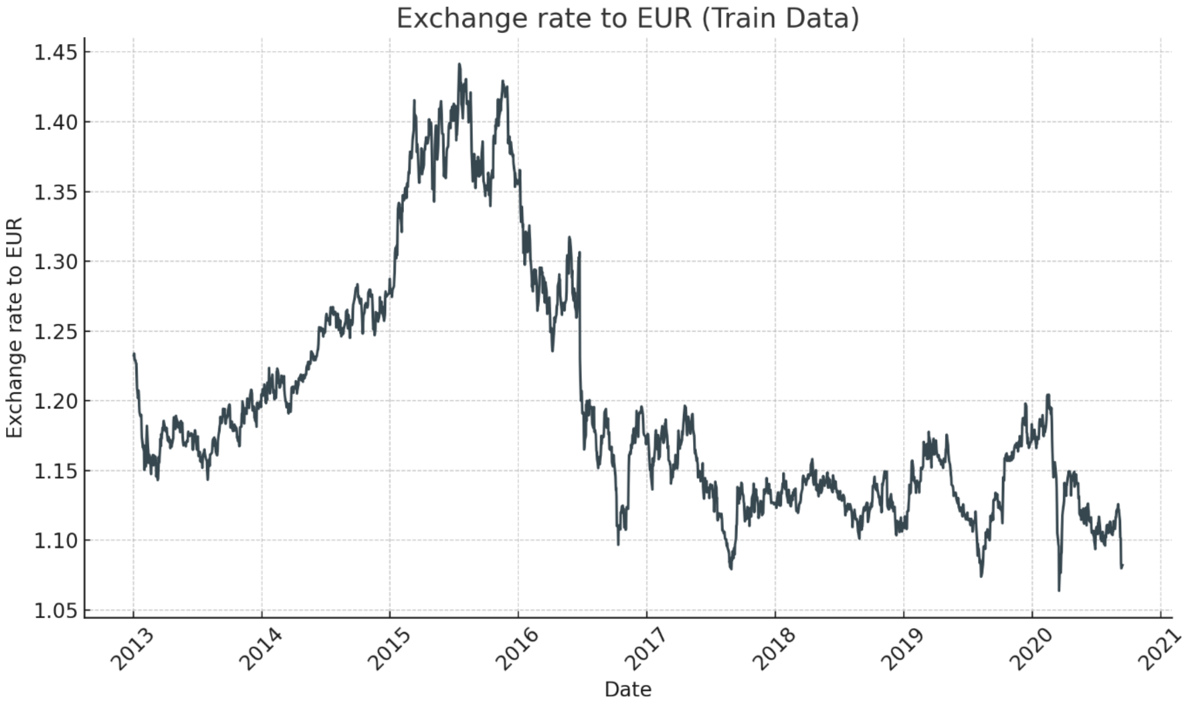
Table 1. Training Data

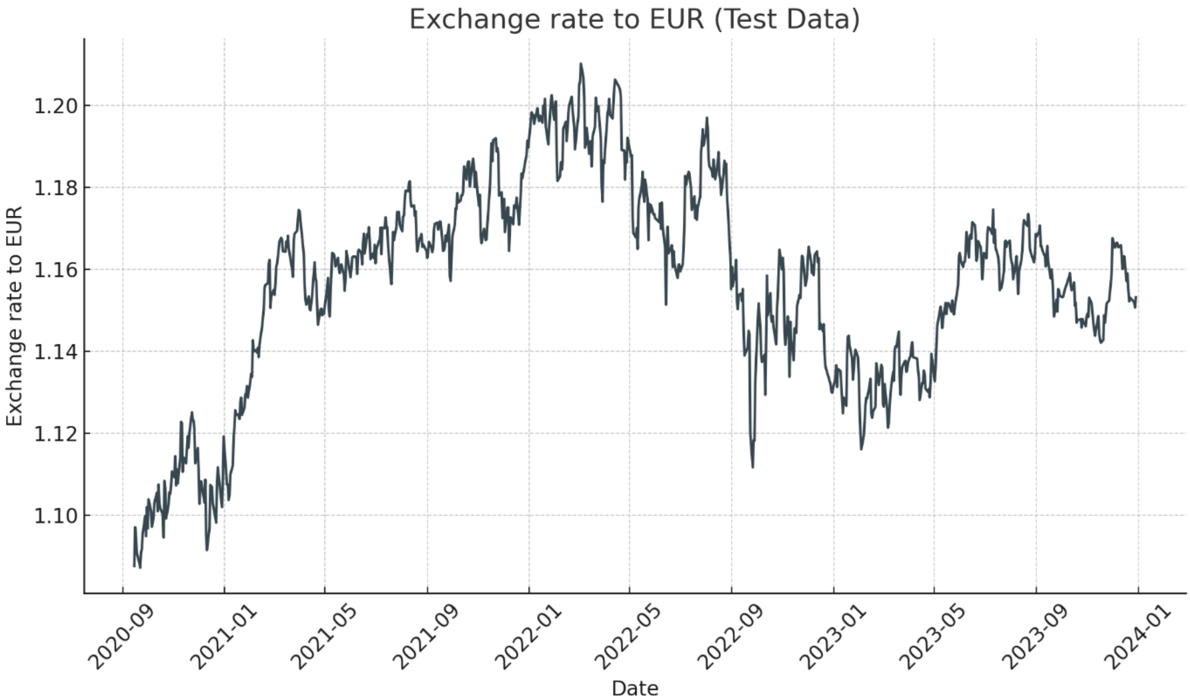
Metric	CNN			LSTM			CNN-LSTM		
	Mean	Median	Standard Deviation(SD)	Mean	Median	Standard Deviation(SD)	Mean	Median	Standard Deviation(SD)
Mean Absolute Error (MAE)	99.30	97.74	16.02	99.30	97.74	16.02	129.86	125.22	29.70
Mean Squared Error (MSE)	20,670	19880	6232	20, 670	19				
Root Mean Squared Error (RMSE)									
R-squared (R^2)									
Mean Absolute Percentage Error (MAPE)									
Symmetric Mean Absolute Percentage Error (SMAPE)									
Explained Variance Score									
Mean Bias Deviation (MBD)									
Directional Accuracy									
Accuracy within 1% tolerance									
Exact Accuracy (within 50 units)									

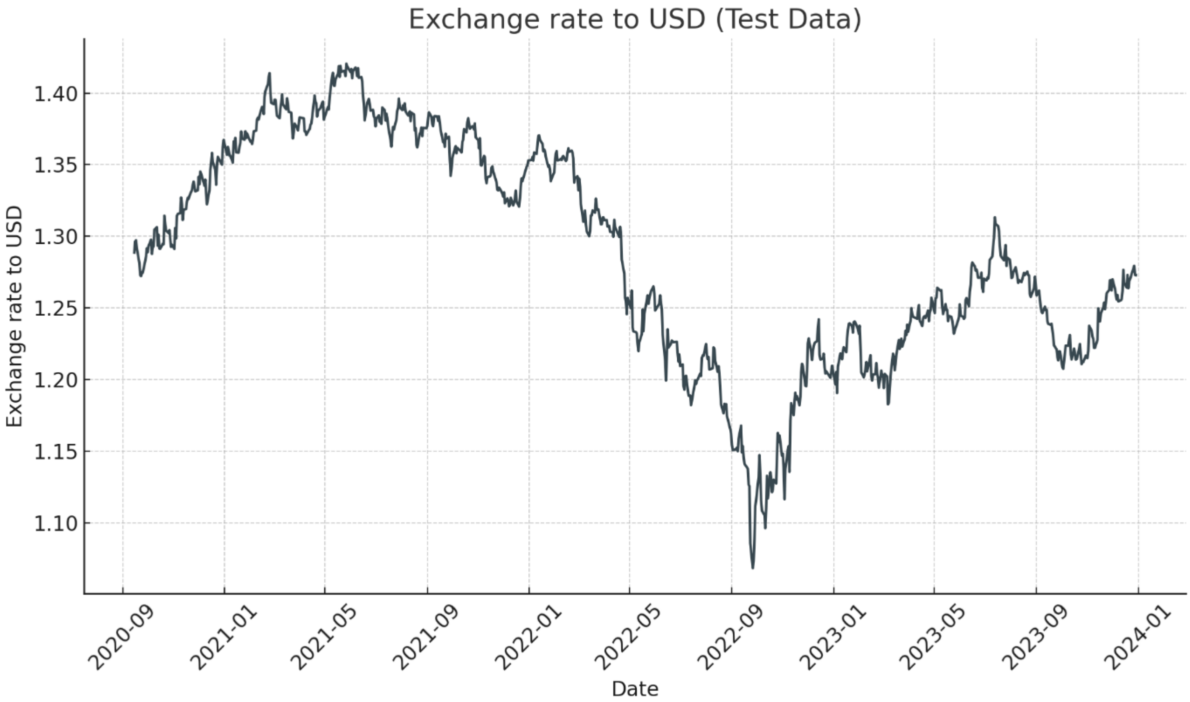
Table 2 Test Data

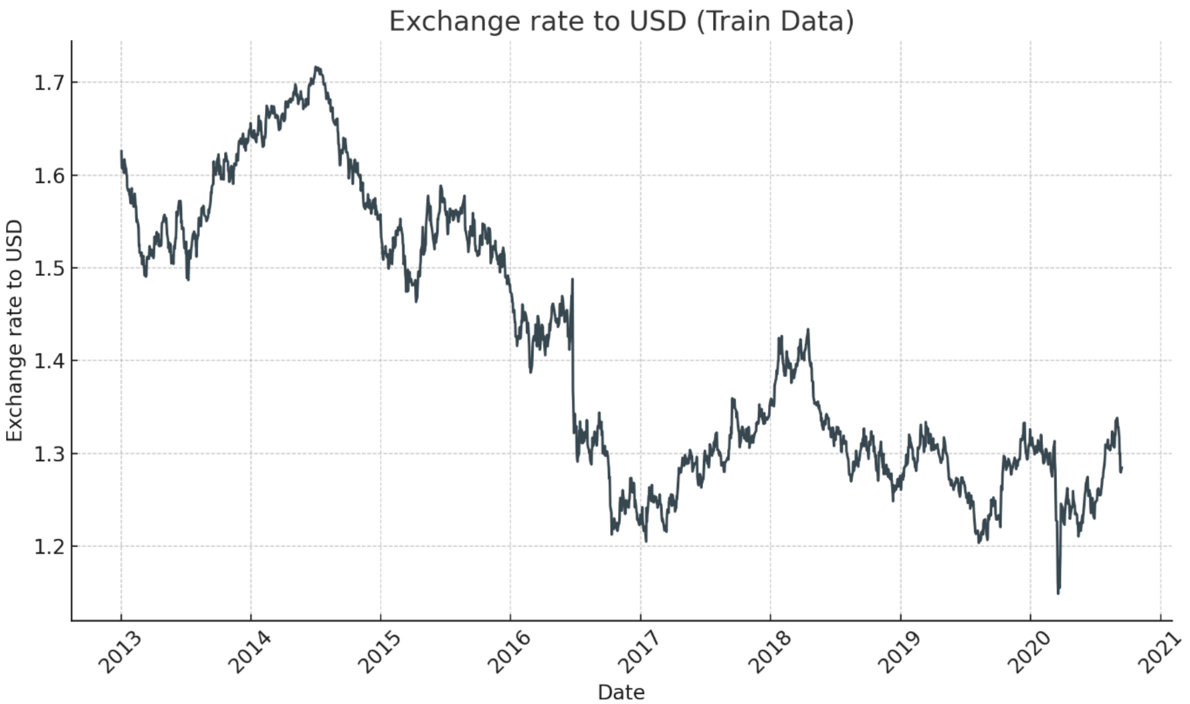

Unemployment rate (Test Data)
Unemployment rate
5.25
5.00
4.75
4.50
4.25
4.00
3.75
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
Date

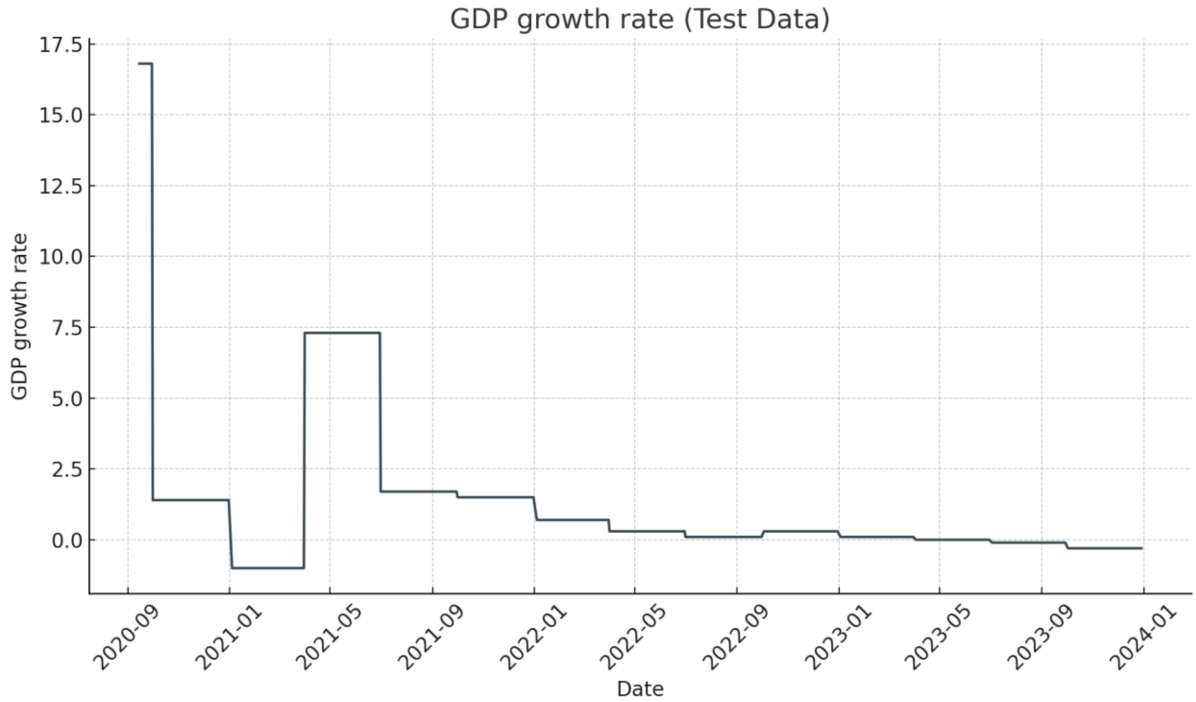

Unemployment rate (Train Data)
Unemployment rate
8
7
6
5
4
2013
2014
2015
2016
2017
2018
2019
2020
2021
Date

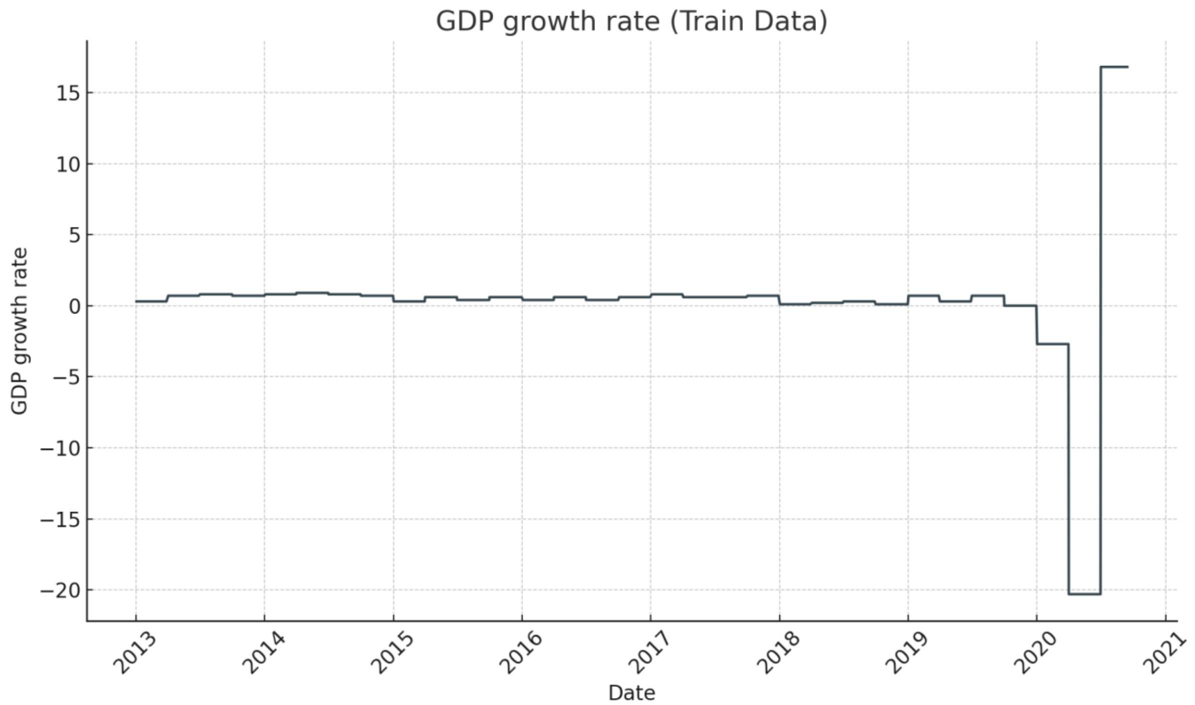

Exchange rate to EUR (Train Data)
Exchange rate to EUR
Date
1.45
1.40
1.35
1.30
1.25
1.20
1.15
1.10
1.05
2013
2014
2015
2016
2017
2018
2019
2020
2021

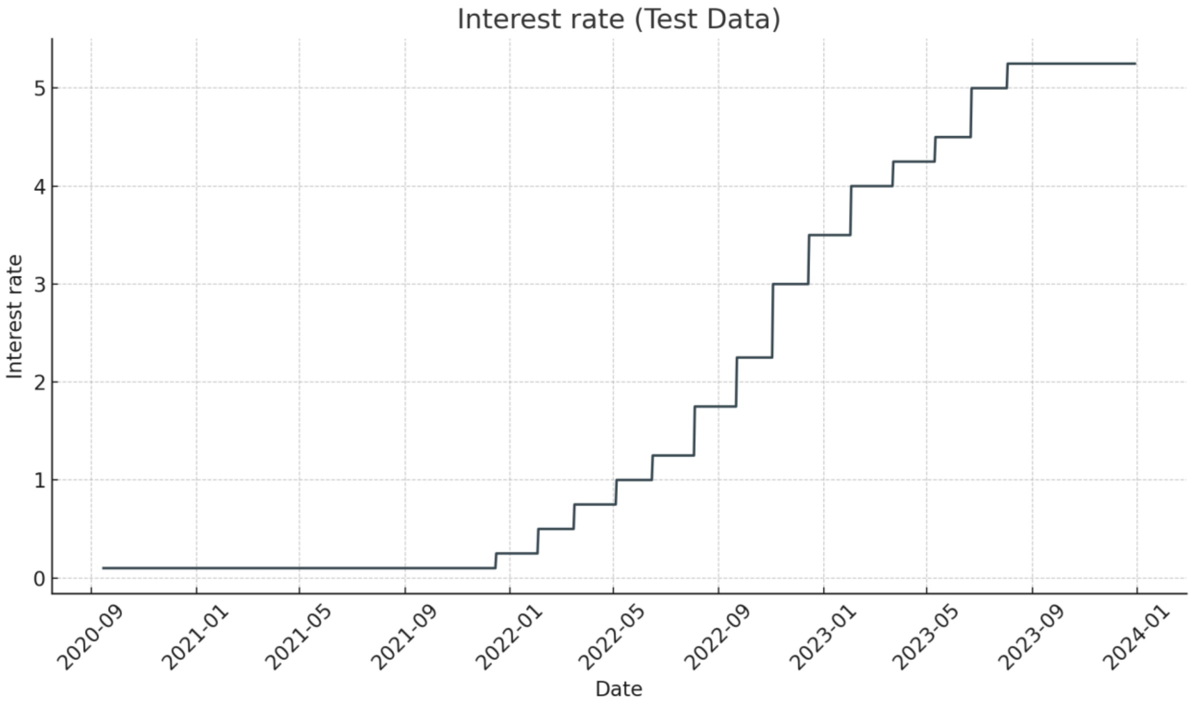

Exchange rate to EUR (Test Data)
Exchange rate to EUR
Date
1.20
1.18
1.16
1.14
1.12
1.10
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01


Exchange rate to USD (Test Data)
Exchange rate to USD
1.40
1.35
1.30
1.25
1.20
1.15
1.10
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
Date

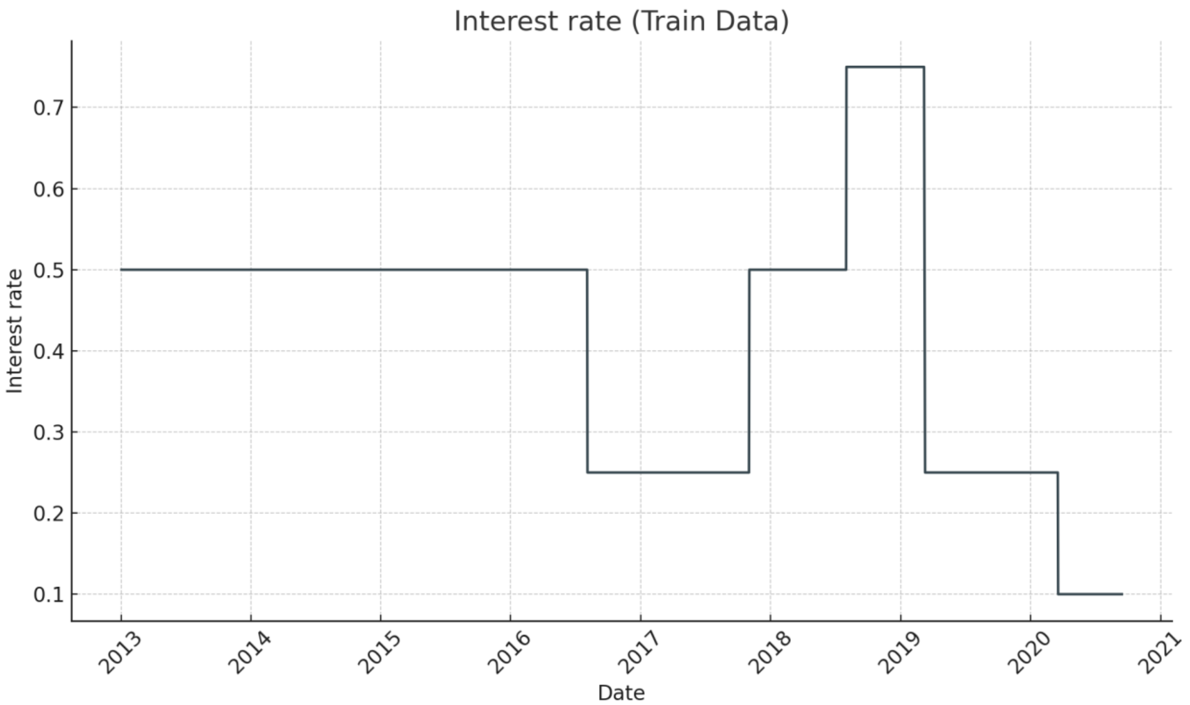

Exchange rate to USD (Train Data)
Exchange rate to USD
1.7
1.6
1.5
1.4
1.3
1.2
2013
2014
2015
2016
2017
2018
2019
2020
2021
Date


GDP growth rate (Test Data)
GDP growth rate
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
Date

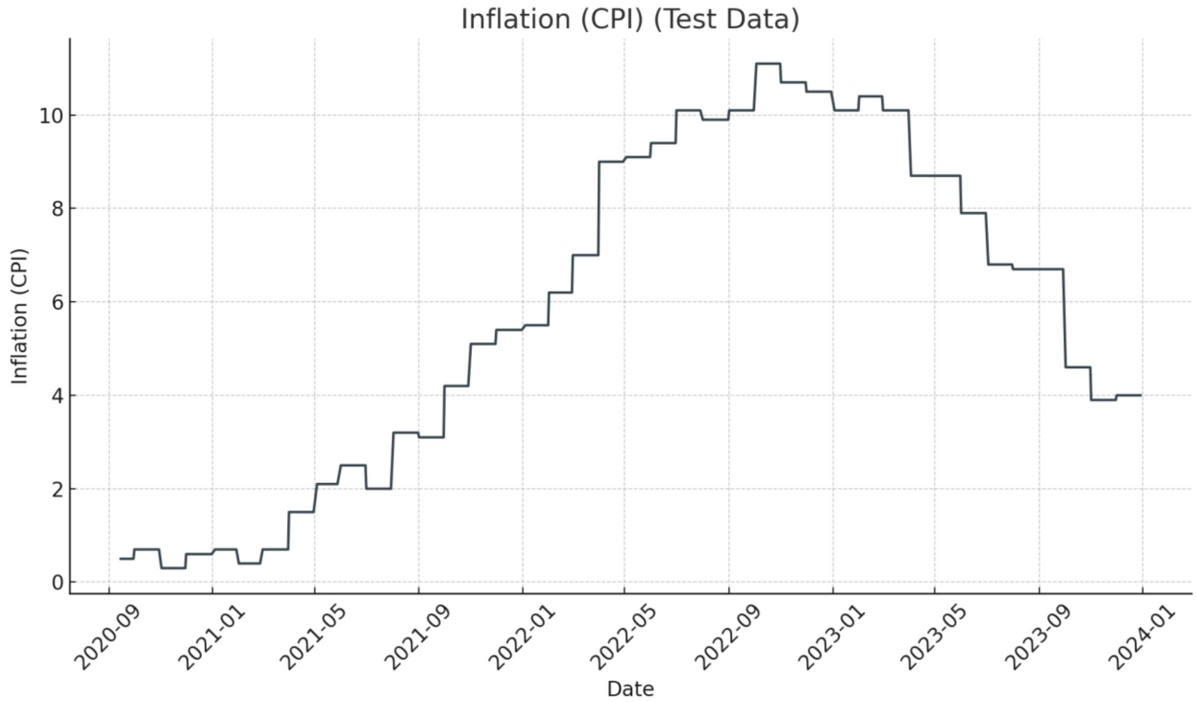

GDP growth rate (Train Data)
GDP growth rate
15
10
5
0
−5
−10
−15
−20
2013
2014
2015
2016
2017
2018
2019
2020
2021
Date



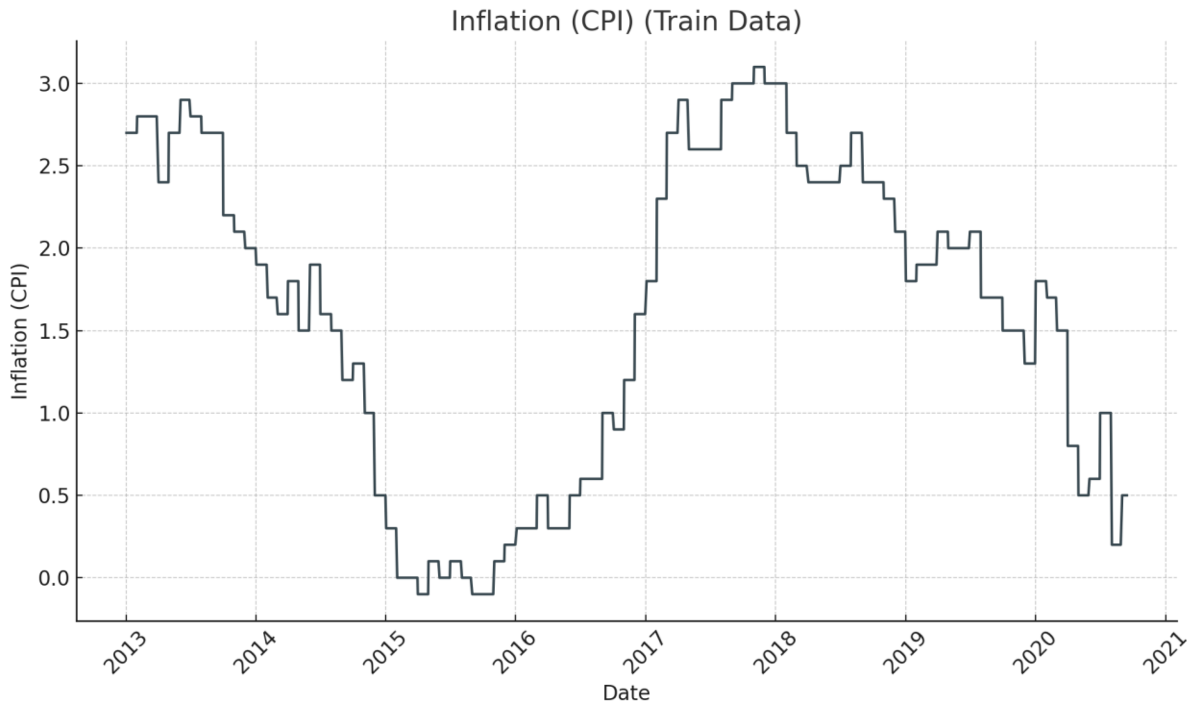
Interest rate (Test Data)
Interest rate
0
1
2
3
4
5
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
Date



Interest rate (Train Data)
Interest rate
0.1
0.2
0.3
0.4
0.5
0.6
0.7
2013
2014
2015
2016
2017
2018
2019
2020
2021
Date



Inflation (CPI) (Test Data)
Inflation (CPI)
Date
0
2
4
6
8
10
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01



Inflation (CPI) (Train Data)
Inflation (CPI)
Date
0.0
0.5
1.0
1.5
2.0
2.5
3.0
2013
2014
2015
2016
2017
2018
2019
2020
2021