Stock market forecasting with optimal feature selection using neural networks

Alvie Mahmud

School of Computing
Final Year Research Project

May 20, 2022

Abstract

No more than 300 words summarizing this dissertation.

Consent to share

I consent for this project to be archived by the University Library and potentially used as an example project for future students.

Table of Contents

List of Tables

List of Figures

Acknowledgements

Thanks.

Chapter 1

Introduction

1.1 Overview

Stock markets are often thought to be irrational or in some cases random, which suggests that markets are not made based on fundamental factors alone. In a 1996 speech, former Federal Reserve chair, Alan Greenspan questioned "... how do we know when irrational exuberance has unduly escalated asset values ..." (Greenspan, 1996), suggesting that there may be psychological factors that affect markets such as positive feedback loops.

This project intends to identify if a stock market index's direction can be predicted ahead of time, and which factors contribute the most to an accurate prediction. The intended audience for such a project can vary from the institutional to the retail investor to identify opportunities in the market by using only publicly available data rather than proprietary information.

1.2 Aims

This project aims to produce an artificial intelligence model that attempts to predict the following day's price of a stock market index. This will allow

the project to research if a stock market index's price can be predicted from the day prior; the factors which affect a stock market index's price and which factors are most important in predicting a stock market index's price.

1.3 Objectives

The objectives are as follows:

- Identifying key factors that can affect stock market performance.
- Collecting above-mentioned factors, where the historical data is publicly available and free to use (either public domain, non-commercial licenses, etc.).
- Transforming collected data into normalised, or otherwise modified, data that can easily be used to identify patterns, either by humans or by computational neural networks.
- Identifying and applying correct techniques to train a computational neural network (or other suitable technology) to understand which factors are most important when predicting stock market performance.
- Filtering through training data to ensure there are no biases in play.
- Ensuring that the network will be efficient enough to be trained on hardware that is available today, and if possible on consumer hardware such as Nvidia RTX 3000 series GPUs.
- Understanding outputs of the computational neural networks and presenting the output data learned in a way that is clear.
- Interpreting overall results to identify if it is possible to predict a stock market's index direction and/or price for any given day
- Evaluating the results to identify accuracy, limitations, and biases

1.4 Constraints

- May be difficult / expensive to get historical data
- May be difficult / expensive to get reliable data
- Access to currently publicly available data may be removed, or otherwise unavailable
- May not have the correct technical skills to sufficiently carry out the projects
- May not have sufficient time to carry out the full objectives of the project
- Hardware available may not be feasible to run / train complex neural networks on

Even if the project was successful for now, it may not be suitable for these reasons:

- May not work in extreme market events
- May not work if new market measures are placed
- May not work if new economic measures are placed
- Access to reliable data may be restricted, removed or otherwise unavailable in the future

Chapter 2

Literature Review

Keywords. artificial intelligence, stock market

2.1 Introduction

As the world advances in technology, an increasing number of decisions are dependent on computational models; including financial decisions. This includes, but is not limited to, decisions for money management, risk management, economics, and investing. This literature review will discuss: the feasibility of predicting stock market prices; different artificial intelligence research methods used in financial modelling, particular those that relate to investing and stock market indices; and the different input features used in these models.

2.2 Feasibility of predicting stock market index prices

2.2.1 Efficient Market Hypothesis

Over the year, many people have attempted to predict stock markets in order to profit from price movements as they occur. However, some have theorised that it is not possible to accurately predict prices due to the there being an innumerable number of factors that can affect a stock market, and these factors would already be a function of the current market's price, therefore make it impossible to forecast the price. (Bachelor, 1900, p. 1) This is the basis of the Efficient Market Hypothesis, where an asset price fully reflects the information that is available at any given time.

2.2.2 Empirical evidence of market (in)efficiencies

A 1973 book explains the term 'random walk' as "A random walk is one in which future steps or directions cannot be predicted on the basis of past actions", and continues to describe how when applied to the stock market that "short-run changes in stock prices are unpredictable. [and] Investment advisory services, earnings forecasts, and complicated chart patterns are useless" (Malkiel, 1973, p. 24). A reflection paper by the same author 30 years on, attempts to validate that the efficient market hypothesis still holds true in the 21st century by comparing actively managed funds to tracker funds: suggesting that markets are likely efficient, due to the fact that most actively managed funds are outperformed by a index tracker fund that follows the S&P 500 index; the data presented suggests that most experts (68% to 90% over various time periods) cannot outperform the index whether it be from a one year, three year, five year, ten year or twenty year holding period leading up to 31st December, 2003 (Malkiel, 2005). The author suggests that if markets were inefficient, fund managers should easily be able to outperform the index.

In order to prove, or disprove, the Efficient Market Hypothesis, many studies have been carried out with varying results. In this literature review we will focus on the US markets and the different results the studies have found. Using data from 1964 to 2003 for the US stock market, one study has found markets to be efficient using a threshold autoregressive model and a unit root process (Narayan, 2006). Another empirical look into equity premium prediction, that looks specifically at the US S&P 500 index, using various regression models with additional variables described such as dividends, earnings price ratio, book value, net issuing activity and more has concluded that "the equity premium has not been predictable" (Goval & Welch, 2004, abstract) which would agree with the efficient market hypothesis.

However, other recent studies have suggested the efficient market hypothesis may not hold true. A paper that looks into returns of the US Dow Jones Industrial Average (DJIA) index from 1928 to 2012 showed that while for autocorrelation tests on daily and weekly intervals suggest market efficiency, the same cannot be said for monthly and annual intervals (although the degree of correlation is low). Furthermore, the paper posits that autocorrelation tests are not sufficient to determine dependencies, and puts forwards another type of test called 'run tests' to show that markets are also inefficient in daily and weekly intervals (Sewell, 2012). Another paper that analyses data from 1999 to 2007 of various markets, including the US as well as other developed and developing countries, using a unit root process has stated that their findings show that "real stock price indices are stationary processes that are inconsistent with the efficient market hypothesis" (Lee et al., 2010, p. 1).

2.3 Types of artificial intelligence methods used

2.3.1 Description of artificial intelligence methods

Various artificial intelligence methods have been utilised in order to predict daily direction as well as price of stock markets, the following are described here:

- Multilayer Perceptrons (MLPs)
- Convolutional Neural Networks (CNNs)
- Support Vector Machines (SVMs)
- Long Short Term Memory (LSTM)
- Hybrid approaches

Multilayer Perceptrons (MLPs)

Multilayer Perceptrons (MLPs - Figure 2.1) are a class of artificial neural networks. They contain multiple layers of perceptrons, including an input layer, one or many hidden layers and an output layer. They are feed-forward networks meaning that data is only passed to the next layer and does not move backwards. It is an example of a supervised neural network as labelled data is used in training.

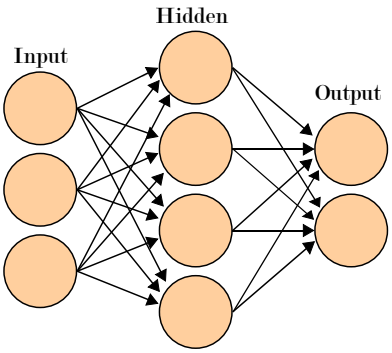

Figure 2.1: Diagram of a Multilayer Perceptron; adapted from Artificial neural network.svg, Wikimedia, Cburnett

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs - Figure 2.2) are a class of artificial neural networks. They contain multiple layers; including an input layer, one or many convolutional and pooling layers, and an output layer. The convolutional and pooling layers are inspired by biological processes; each layer extracts and summarises certain features and convolves the input and passes to the next layer. The pooling layers are using to reduce the spatial size of the data by combining outputs in order to optimise the network for performance.

They are feed-forward networks meaning that data is only passed to the next layer and does not move backwards. It is an example of a supervised neural network as labelled data is used in training.

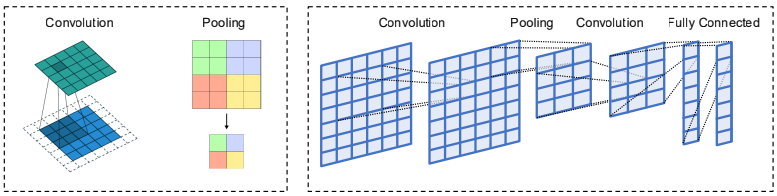

Figure 2.2: Diagram of a Convolutional Neural Network; (Maier et al., 2019)

Support Vector Machines (SVM)

Support Vector Machines (SVMs - Figure 2.3) are a class of machine learning models that are used in classification of data. In this model, the training data is used in order to identify a decision boundary, known as a hyperplane that separates the classifications of data. It is an example of a supervised machine learning model as labelled data is used in training.

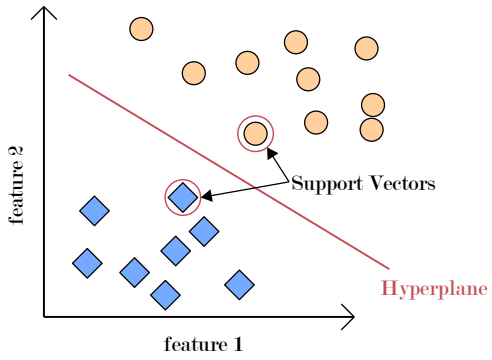

Figure 2.3: Diagram of a Support Vector Machine

Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM - Figure 2.4) is a class of artificial neural networks. They are recurrent networks meaning that data is not necessarily fed-forward as they allow for loops within the network. LSTMs contain cells with 'gates' that allow them to keep context and control the flow of data, as well as make decisions to keep or discard data in order to make predictions. It is an example of a supervised neural network as labelled data is used in training.

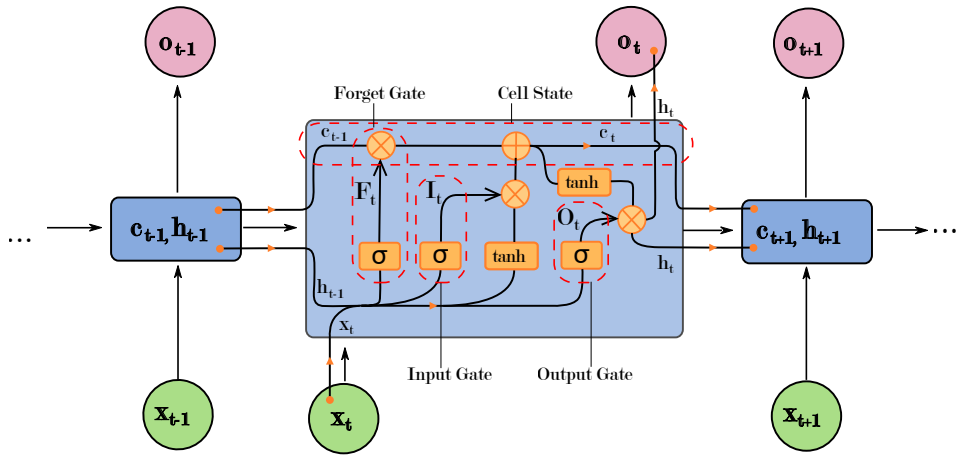

Figure 2.4: Diagram of a Long Short Term Memory Unit; adapted from Long Short-Term Memory.svg, Wikimedia, fdeloche

Hybrid Approaches

Hybrid approaches can be utilised in order to overcome shortcomings in certain models. A hybrid approach can utilise any number of different models; for example a CNN-LSTM hybrid model will use CNN and LSTM layers as a CNN may be a good model for classification problems, an LSTM is better suited to time series data.

2.3.2 Artificial intelligence methods used in previous studies

Different studies have utilised different artificial intelligence models to predict the price of stock market indices, with varying degrees of accuracy.

One study using multilayer perceptrons (MLP) to predict the daily direction of an exchange traded fund (ETF) that tracks the US S&P 500 stock market index suggests that the model has an accuracy of up to 60% (Zhong & Enke, 2019). Whereas another study has suggested they can predict the price, not just direction, of the same stock market index with an accuracy of 76% using support vector machines (SVMs) and reinforcement learning

(Shen et al., 2012). As there is a difference of 16%+ in of accuracy in the same stock market index (S&P 500), and the higher accuracy model suggests that support vector machines may provide an improvement over feed-forward neural networks such as MLPs. Another study comparing the performance of MLPs and SVMs agrees that SVMs are superior as, even though both models were able to predict the direction of the S&P 500 index, the MLP model had a maximum error difference of 15% over a 45 day period, whilst the SVM model had a maximum error difference of 6% in their test cases(Sheta et al., 2015). This is further supported by another study comparing various artificial intelligence methods on the Indian Nifty50 index with the SVM model outperforming a back-propagation neural network model, which is somewhat similar to the MLP model, by 5.51% (Kumar & Thenmozhi, 2006).

Further studies have been done on using hybrid approaches to predict stock markets. One study using an approach with a convolutional neural network (CNN) and three long short-term memory (LSTM) networks found that the accuracy of predicting weekly directions of the S&P 500 index was 66.6%, which was greater than with SVMs or CNNs alone, with those models achieving 62.0% and 59.3% respectively (Hao & Gao, 2020). Another study agrees with these findings, with their results also suggesting a model built with CNNs and bi-directional LSTMs are able to outperform SVMs (Eapen et al., 2019). However, the same cannot be said for all types of hybrid neural network models; one study comparing MLPs and hybrid networks on the US Nasdaq index has found that the MLP performs better, albeit slightly with a difference of 0.26% in mean absolute deviation, than an approach with a hybrid model of MLP and Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) (Guresen et al., 2011).

2.4 Input features for intelligence methods

The studies mentioned in the previous section use different input features for their studies; where the amount of input features ranges from just a single

type of input to many various macro-economic factors. One of the hybrid approaches has only stated using the 'daily closing price dataset of the S&P 500 index' in their CNN-LSTM model (Hao & Gao, 2020). Another study using the hybrid approach slightly expands on this by using additional factors related to the index: the opening and closing prices of the S&P 500 index, the low and high prices of the day, and the trading volume (Fitriyaningsih et al., 2019; Hu et al., 2018; Thakkar & Chaudhari, 2020). However, these studies have not considered any macro-economic factors. One study expands on this further by also including other indexes across the world as inputs, as well as various currency rates in addition to commodities such as oil and metals (Shen et al., 2012). Other studies expand on this further, for example one study with 27 input features also includes US Treasury yield rates and bond yields (Sheta et al., 2015) and another looks into additional factors for a total of 60 input features including certificate of deposit rates, and term and default spreads (Zhong & Enke, 2019) - though the latter study explains that with principal component analysis, the model peaked in terms of accuracy with 31 input features.

2.5 Conclusion

To conclude, there are differing opinions on whether markets are efficient, and therefore whether they are predictable. Some economists use the fact that experienced fund managers are rarely able to beat the market as proof that markets are efficient, and there are studies that agree with this hypothesis. However, there are also other studies that disagree presenting their findings that show there is some evidence of market inefficiencies in either the short term or long term.

Regardless of markets being efficient or not, numerous artificial intelligence methods — including MLPs, SVMs, CNNs as well as hybrid approaches using CNNs and LSTMs – have been employed to varying degrees of success. While some studies look at predicting just the direction, and others attempt

to predict the price, most have claimed to be able to do so with above 50% accuracy. From the studies researched here, it can be assumed that in general hybrid approaches involving CNNs and LSTMs outperform SVMs, which outperform MLPs.

These studies use a diverse set of input features in order to build their models, with some using features only directly related to the index and some using extended datasets including inputs from macroeconomic sources in order to build models that take account external factors that may affect the index's price, though there is evidence to suggest that not all of the input features are required to build a good model to predict the price of an index.

Chapter 3

Research Questions

3.1 Predicting a stock market index's direction

Question Is it possible to accurately predict a stock market index's (S&P 500) direction for the following day with an artificial intelligence model?
Sub Questions

- If so, what is the accuracy of the model?
- If not, what limitations affected the model to cause it to be inaccurate?
- Can the findings of this question be used to prove/disprove the efficient market hypothesis?

Importance The findings from this research question will have significant impact in making investment and trading decisions for participants of the market, potentially giving indicators of when to buy / sell into the stock market in order to outperform the market. Furthermore, depending on the accuracy of the model, it may be able to be used as evidence for or against the efficient market hypothesis.

3.2 Optimal amount of historical data as input

This question assumes that it is at least somewhat possible to accurately predict the stock market index's direction/price based on literature review and other studies.

Question How do different lengths of time as history for the input in the training data affect the model?

Sub Questions

- What is the ideal length of time to be used in the training data?
- What are potential reasons for this length of time being the most useful?

Importance The findings from this research question will allow optimal algorithms / models to be created in order to avoid wasting computational time that may cause the model to not improve, or even decrease in accuracy. Furthermore, this will be of help to participants of the market to make better informed manual decisions by understanding which time horizons to look at.

3.3 Most important input features

This question assumes that it is at least somewhat possible to accurately predict the stock market index's direction/price based on literature review and other studies.

Question What are the most important input features that affect the model?

Sub Questions

- How much do each of these input features contribute to the model?
- Do any of the input features identified have negligible impact to the model?

- What are the potential reasons these features do / do not impact the model?

Importance The findings from this research question will allow optimal algorithms / models to be created in order to avoid wasting computational time that may cause the model to not improve, or even decrease in accuracy. Furthermore, this will be of help to participants of the market to make better informed manual decisions by understanding which factors to look at.

Chapter 4

Research Design

4.1 Research methodology

The research methodology is one based on the literature review (chapter 2); the research already carried out in other studies informed the decision making process behind the requirements of the supporting artefact. Various artificial intelligence models have been analysed and critiqued; as well as the input features used within these models.

These aspects combined are used to inform the variables - such as model used, input features (and their sequence lengths of historical data) - to experiment with to better understand which variables are most important and have the ability to provide the most accurate results.

4.2 Research process

4.2.1 Comparison of AI models in previous studies

From the literature review, various AI models were identified. These include Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), Long Short Term Memory (LSTM) and

hybrid approaches. From this, we can understand which models can have greater accuracy. The review has found a hybrid approach has the greatest accuracy. CNNs and SVMs have a similar degree of accuracy and generally outperform MLPs.

Artificial Intelligence Model	Ease of Implementation	Accuracy of Output
Multilayer Perceptrons	5	3
Convolutional Neural Networks	3	4
Support Vector Machines	3	4
Hybrid (CNN + LSTM)	2	5

Table 4.1: Table rating different AI models on a scale of 1 to 5 based on ease of implementation as well as accuracy

4.2.2 Input features used in previous AI models

Currently, based on the studies in the literature review, it is difficult to suggest which input features are most important for an accurate output. This is due to the studies using more complex datasets as input features generally use a different research model.

This is one of the primary research questions identified, and this project intends to answer this question. The previous studies use the following input features; a subset of the features will be used and compared within this project.

- Daily Closing Price
- Low Price
- High Price
- Volume

- Currency Rates
- Commodities (e.g. Oil)
- Treasury Rates
- Stock indices of other countries
- Individual Stocks
- Certificate of Deposits
- Term spreads

4.3 Expected research outcomes

As CNN+LSTM hybrid approaches have already been proven in previous studies to be an accurate research model, this model will be researched within the artefact of this project. However, the effect of different input features on a single research model has not been well identified. This will be a primary area of focus for the supporting artefact to look into. The academic novelty exists within the fact that there is a unique set of input features on the CNN+LSTM model or other well-performing model; and also finding the optimal sequence and combination of input features. Ideally, this will allow the supporting artefact to have greater than 60% accuracy in predicting the daily price direction.

4.4 Requirements of the artefact

4.4.1 Functional requirements

Based on the literature review, the most accurate models for predictting stock prices had an accuracy of approximately 60% (Zhong & Enke, 2019). While there is an example of a study shown with a greater accuracy of 76% (Shen et al., 2012), another similar study places the accuracy of the model at

59.3% - albeit for weekly predictions rather than daily (Hao & Gao, 2020); thus 60% is considered the current baseline.
Also, due to the fact that this research project attempts to identify the most important input features, as well as the sequence length, they are key requirements of this project.
Predicting the exact return (daily relative price change) per day is an optional requirement due to two reasons: it does not significantly help market participants compared to knowing the direction alone; and it may be more cumbersome / time-consuming for a relatively low significant output for end users.

Necessary requirements

- Predict daily price direction with 60%+ accuracy
- Identify which input features are most important
- Identify what sequence length (number of days of history of each input feature) is optimal
- Present charts of accuracies and losses of each model

Optional requirements

Based on the literature review, a study suggested it had been able to accurately predict prices with a mean absolute deviation of 2.516% on the testing data set when compared to the true price.

- Predict daily price return with a mean absolute deviation of 2.5% or lower

4.4.2 Non-functional requirements

The results from the artefact are not time critical due to there being a large time period 17 hours and 30 minutes between market close (4:00 PM ET) and market open (9:30 AM ET) of stock exchanges in the USA; however this

may not be ideal for users thus the following non-functional requirements have been identified.

Necessary requirements

- The artefact should take less than one minute to run per model chosen (on modern Nvidia GPUs)

Optional requirements

- Front-end application that allows users to input features and sequence lengths to train different models
- User-facing documentation for artefact

Chapter 5

Artefact Design

5.1 Development decisions

5.1.1 Software development methodology

In the software development life cycle there are various stages: requirements gathering, analysis, design, development, testing, deployment and maintenance.

The Agile software development methodology will be utilised in this project; and this project will be separated into various iterations of design, development, and testing. This will allow for a more flexible approach to the task and the ability to make changes as the project progresses without requiring additional time and effort to fully test and deploy an inaccurate or unsuitable artefact. There are various iterations that will be used to gradually improve to a final version that will be used to form answers for the research questions.

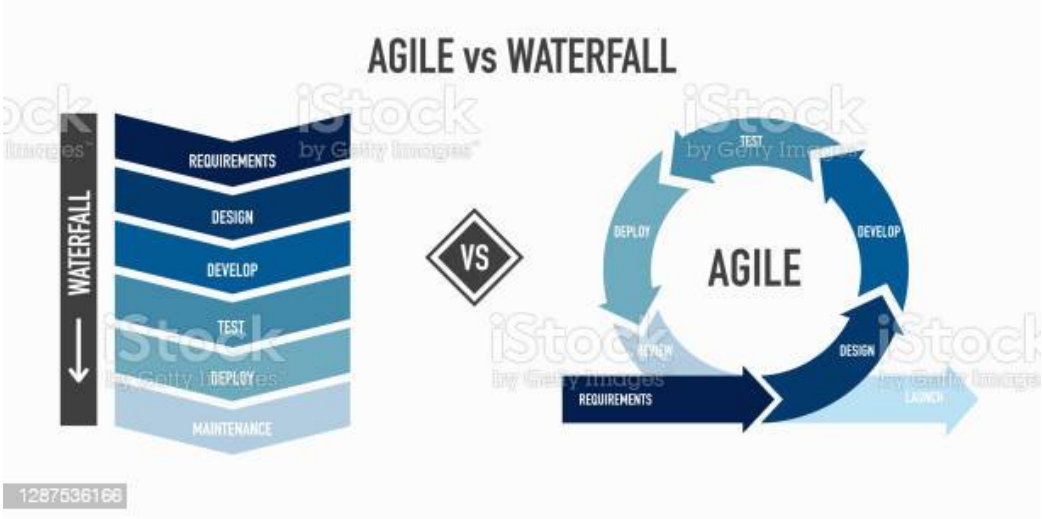

Figure 5.1: A comparison of the Agile and Waterfall methodologies (source: iStock, iam2mai) (replace as there is a copyright)

5.1.2 Software languages and libraries

Programming Language

Python has been chosen to be used in this project as it is ubiquitous for artificial intelligence related tasks. There are many resources readily available for Python with regards to its usage on artificial intelligence models and neural networks. There are various Python libraries for machine learning applications such as Tensorflow, an 'end-to-end open source machine learning platform' created by Google as well as PyTorch, ' open source machine learning framework' created by Facebook's AI Research Lab.

Python libraries for machine learning

As mentioned in the previous section there are various machine learning libraries for Python. Currently, the most popular is Tensorflow and has the most resources available. However, the usage of PyTorch has accelerated over the past years and is quickly becoming the preferred library as it is considered to be more 'pythonic' and often quicker to train neural networks.

Tensorflow will be used in the project due to its current ubiquity in this task, especially surrounding its applications for financial forecasting. However, PyTorch may be chosen for future iterations if the performance of Tensorflow is found to be insufficient and the performance of PyTorch is found to be greater.

5.2 Input features chosen and data collection

The input features chosen are an amalgamation primarily based on those found in the literature review. However, there are several input features that have previously not been used in such as repurchase agreements data, as well as sentiment data, etc. This represents a first time application of an existing research method to a novel case study for many of the models.

From the literature review, the input features have been classified into four categories:

- Stock market index
- Money availability
- Alternative portfolio allocations
- Sentiment

From this, the following sources for input features have been chosen; the reasoning are explained and the sources for used in the data collection process are also present within this list.

- Stock market index
  - Closing price
    Reasoning: It is believed that a sequence of closing price returns can be utilised to predict a future closing price
    Source: Yahoo Finance - SPY Historical Data:
    https://uk.finance.yahoo.com/quote/SPY/history

- Volume

  Reasoning: It is believed that the amount of stock traded can indicate confidence in that stock market index and potentially future price movements

  Source: Yahoo Finance - SPY Historical Data: https://uk.finance.yahoo.com/quote/SPY/history

- Volatility (based on VIX)

  Reasoning: It is believed that the volatility in the stock market can potentially indicate future price movements

  Source: Yahoo Finance - VIX Historical Data: https://uk.finance.yahoo.com/quote/%5EVIX/history

- Money availability

  - M1 Money Supply

    Reasoning: It is believed that a change in total money supply circulation can affect the amount of money allocated to the stock market, such as with measures related to quantitative easing to affect stock returns

    Source: Federal Reserve Economic Data - M1 Money Supply: https://fred.stlouisfed.org/series/WM1NS

  - GDP

    Reasoning: It is believed that the GDP can affect the money available to market participants and thus impact stock returns

    Source: Federal Reserve Economic Data - GDP: https://fred.stlouisfed.org/series/GDP

- Alternative portfolio allocations

  - Treasury Yields

    * 1Mo
    * 3Mo

    * 1Yr
    * 2Yr
    * 5Yr
    * 10Yr
    * 20Yr
    * 30Yr

    Reasoning: It is believed that the market participants often allocate their funds to treasury bonds / bills / notes for a guaranteed return compared to risk in the stock market. A change to yields may affect the money allocated in the stock market and thus affect the return of the stock market index.
    Source: US Dept. of the Treasury - Treasury Par Yield Curve Rates:

- Effective Federal Funds Rate
  Reasoning: EFFR is the interest rate banks charge one another for overnight loans, a change in EFFR may affect the loans banks make / take and thus affect the money the banks have available to allocate in the stock market; this may affect the return of the stock market index.
  Source: Federal Reserve Economic Data - EFFR:
  https://fred.stlouisfed.org/series/EFFR
- Repurchase / Reverse Repurchase Agreements
  Reasoning: A change in utilisation of repurchase agreements may indicate if banks expect a greater return elsewhere and how banks are allocating money. This may have an impact on the return of the stock market index
  Source: New York Fed - Repo:

https://www.newyorkfed.org/markets/desk-operations/repo

- Gold

  Reasoning: It is believed that a change in the prices of commodities such as gold may be an indicator of portfolio allocation changes; thus affecting the return of the stock market index
  Source: London Bullion Market Association - Precious Metal Prices:
  https://www.lbma.org.uk/prices-and-data/precious-metal-prices#/table

- Foreign Currency

  * USD-GBP
    Source: Federal Reserve Economic Data - GBP:
    https://fred.stlouisfed.org/series/DEXUSUK

  * USD-EUR
    Source: Federal Reserve Economic Data - EUR:
    https://fred.stlouisfed.org/series/DEXUSEU

  * USD-JPY
    Source: Federal Reserve Economic Data - GBP:
    https://fred.stlouisfed.org/series/DEXJPUS

  Reasoning: It is believed that the currency exchange rates may be an indicator of confidence of the country of the currency, thus a change in currency rate may result in a change of portfolio allocations of participants in that country's stock market.

- Sentiment

  - Put-to-Call Ratio

    Reasoning: It is believed that the market participants may use the options market to speculate on future returns of the stock market. The put to call ratio indicates the sentiment of the market participants and may have an affect on the return of the stock

market index.
Source: AlphaAlerts: Historical Equity Put/Call Ratio
https://www.alphalerts.com/live-historical-equity-pcr/

– Employment Rate
Reasoning: It is believed that market participants may react to changes of employment rate in making investment decisions. A change in nationwide employment may cause market participants to change their portfolio allocations due to lesser confidence in ability for companies in the country to produce and sell; thus potentially affect the return of the stock market index.
Source: Federal Reserve Economic Data - Employment Rate:
https://fred.stlouisfed.org/series/LREM64TTUSM156S

– Inflation Rate
Reasoning: It is believed that a change in inflation rate can cause market participants to change their portfolio allocations and thus affect the return of the stock market
Source: US Bureau of Labor Statistics - Inflation:
https://www.bls.gov/data/#prices

– Consumer Sentiment
Reasoning: It is believed that a change in consumer sentiment towards the economy can cause market participants to change their portfolio allocations and thus affect the return of the stock market
Source: University of Michigan - Consumer Sentiment:
https://fred.stlouisfed.org/series/UMCSENT

– Consumer Confidence
Reasoning: It is believed that a change in consumer confidence towards the economy can cause market participantsto change their portfolio allocations and thus affect the return of the stock market
Source: Federal Reserve Economic Data - OECD Indicator for

the US:
https://fred.stlouisfed.org/series/CSCICP03USM665S

5.3 Assumptions made

The primary assumptions are surrounding the trading days; whilst this varies per month / year, it is assumed that 1 week is equivalent to 5 trading days, 1 month is equivalent to 21 trading days, 1 quarter (3 months) is equivalent to 63 trading days, 1 half is equivalent to 126 trading days, and 1 year is equivalent to 252 trading days. These assumptions are made based on the NYSE trading hours that shows a month may have between 19 and 23 trading days, each quarter may have 61 to 64 trading days, generally resulting in a total of 252 trading days (NYSE, 2020).

5.4 Data preprocessing

5.4.1 Data normalisation

For each input feature, the percentage change over the previous period is calculated. For example if the feature is reported daily, the percentage change since the last day is calculated. For the closing price of the stock market index, the percentage change since the last two days, last week, last month and last quarter were also included.

Input Feature	Function Applied
SPY Closing Price	% Change vs Previous Day
	% Change vs Previous 2 Days
	% Change vs Previous Week
	% Change vs Previous Month
	% Change vs Previous Quarter
SPY Volume	% Change vs Previous Day
VIX (Volatility Index)	% Change vs Previous Day
M1 Money Supply	% Change vs Previous Week
GDP	% Change vs Previous Quarter
Treasury Yields (1 Month, 3 Months, 1 Year, 2 Years, 5 Years 10 Years, 20 Years, 30 Years)	% Change vs Previous Day
EFFR	% Change vs Previous Day
Repurchase Agreements	% Change vs Previous Day
Repurchase Agreements Rates	% Change vs Previous Day
Reverse Repurchase Agreements	% Change vs Previous Day
Reverse Repurchase Agreements Rates	% Change vs Previous Day
Gold Price	% Change vs Previous Day
GBP Exchange Rate	% Change vs Previous Day
EUR Exchange Rate	% Change vs Previous Day
JPY Exchange Rate	% Change vs Previous Day
Put-to-Call Ratio	% Change vs Previous Day
Employment Rate	% Change vs Previous Month
Inflation Rate	% Change vs Previous Month
Consumer Sentiment	% Change vs Previous Month
Consumer Confidence	% Change vs Previous Month

Table 5.1: Table showing input features for Iteration 1

Following this, the inputs are assumed to be normally distributed, and are scaled using the scale function of scikit-learn's preprocessing library. Based on values of that input feature, they are scaled with the following equation (Equation 5.1):

$$z = \frac{x - u}{s} \tag{5.1}$$

where x is the sampled value, u is the mean of all of the samples, and s is the standard deviation of the samples.

This results in a value known as the z-score, a measure of how many standard deviations a value is from the mean value. 99.7% of z-score should be between -3 and +3, but there may be outliers that exceed these values. In order to account for this, all values are clipped to be within the range -3 and +3, and then this range is scaled to be between -1 and +1 for optimised operation within the artificial intelligence models.

5.4.2 Training / validation split

The dataset includes data from November 2006 to March 2022; the dataset is split 80:20 between training data and validation data. This means March 2019 is the cut off period betweeen the training and validation datasets. This results in 2730 samples for the training data and 560 samples for the validation data after accounting for some values that are removed due to them having N/A values. There are no overlaps between the training data and the validation data; all validation data samples are from a date after the last date of the training data samples.

5.4.3 Data balancing

In order to ensure the artificial intelligence models do not have any biases associated with them, each of the training and validation datasets will be

shuffled to ensure the model is able to generalise rather than learn. Furthermore, once shuffled, the number of sequences that result in a positive trading day are counted as well as the number of sequences that result in a negative day. The minimum of the two are chosen to ensure there is an even split of inputs that result in a positive trading day, and inputs that result in a negative trading day. Some data balancing code and concepts were adapted from an online tutorial series of creating neural networks (Kinsley, 2018).

5.5 Model fitting

Each of the models in this project were fit using the Adam optimiser with a learning rate of 0.001 and a decay of 1e-6. Adam is a popular and efficient algorithm for gradient descent in deep learning models. The "Sparse Categorical Crossentropy" loss function was utilised in the fitting of the model due to there being multiple label classes. This calculates the loss as the negative sum of the differences between the true label and the log of the softmax probability.

5.6 Training optimisations

In order to improve efficiency in training neural networks the EarlyStopping callback was used in the Tensorflow training process. This allowed the model to be stopped further training if there was not found to be any improvement with specific factors. In the models in this project, a baseline of 0.5 was set for validation accuracy; and a 'patience' of 12 was set meaning that if there were no improvements from the previous best validation accuracy for 12 consecutive epochs, that model will stop early and not train further. This may present itself in accuracy and loss charts with some lines not carrying on for as many epochs as others.

```
early_stopping = EarlyStopping(monitor='val_accuracy',
                               baseline=0.5,
```

```
                                patience=12)
```

5.7 Model accuracy visualisation

In order to visualise the accuracies and losses, as well as time taken for each model, the Tensorboard call back was used. This provided an easy way to filter through data and understand which models performed well, which models were overfitting. Moreover, it aided in identifying consistently well suited parameters to create better models in the future. All of the accuracy and loss charts were created in Tensorboard.

```
tensorboard = TensorBoard(
                    log_dir=f'lstmiteration1logs-{time_at_start}/{name}')
```

5.8 Iteration 1

5.8.1 Artificial intelligence model used

In the first iteration, the LSTM model was utilised in order to create the foundations for the CNN-LSTM model. The LSTM consists of LSTM layers and Dense layers, with the final Dense layer having an output of 2, representing one output for a negative day (0) and another for a positive day (1), and utilised the 'softmax' activation function to predict the probabilities of each output. The LSTM and Dense layers utilised the 'tanh' (hyperbolic tangent) and 'relu' (REctified Linear Unit) activation functions respectively.

5.8.2 Sequence length used

A sequence length of 21 days (approximately 1 month) was used for each of the input features of this model. This is a figure that has not in this iteration been tested to be the optimal sequence length, but with a short test of various sequence lengths, it has been found to be sufficient to produce values over 50% accuracy.

5.8.3 Input features

The input features for this model included all of the input features mentioned in Table 5.1.

5.8.4 Layers and layer sizes

Various amounts of LSTM layers and Dense layers, as well as layer sizes for each were tested. The accuracy of each model varied significantly. For each LSTM layer, there was also a:

- Dropout layer of 0.4 meaning a set of neurons may be deactivated 40% of the time at random; it is a method of regularisation to prevent overfitting.
- Batch normalisation layer meaning the outputs of that layer were scaled to aid efficiency of the model.

The following models were created and compared with the following variables:

Layer Type	Number of Layers	Layer Size
LSTM	1, 2	32, 64
Dense	2, 3	32, 64

Table 5.2: Table showing layers and layer sizes for Iteration 1

5.8.5 Diagram of model used

The diagram below (Figure 5.2) shows the best model identified in iteration 1 of this project. There are two LSTM layers with a layer size of 64; and two Dense layers, one with a layer size of 32 and one with a layer size of 2. For the purposes of this diagram, the 'None' value of each layer's shape can be ignored.

From the diagram, it is shown that the input shape the first LSTM layer takes: (None, 21, 31); this represents the fact there is a sequence length of 21 (trading days) and 31 input features. This LSTM layer returns a sequence, so its output has a shape of (None, 21, 64); it is fairly abstract but represents that there are 64 neurons in the next layer. As the sequences are returned, the next layers also have input and output shapes of (None, 21, 64). The following LSTM layer however does not return a sequence, so its output shape is (None, 64). Dropouts and Batch Normalisation layers are added to aid with regularisation and optimisation of the model. The next layer is a Dense layer, with an input shape of (None, 64) and an output of the shape (None, 32). An additional dropout layer is applied here before transitioning to the final Dense layer that has an output shape of (None, 2) which represents the fact there are two potential outputs for the inputs: whether the following day is an positive or negative trading day.

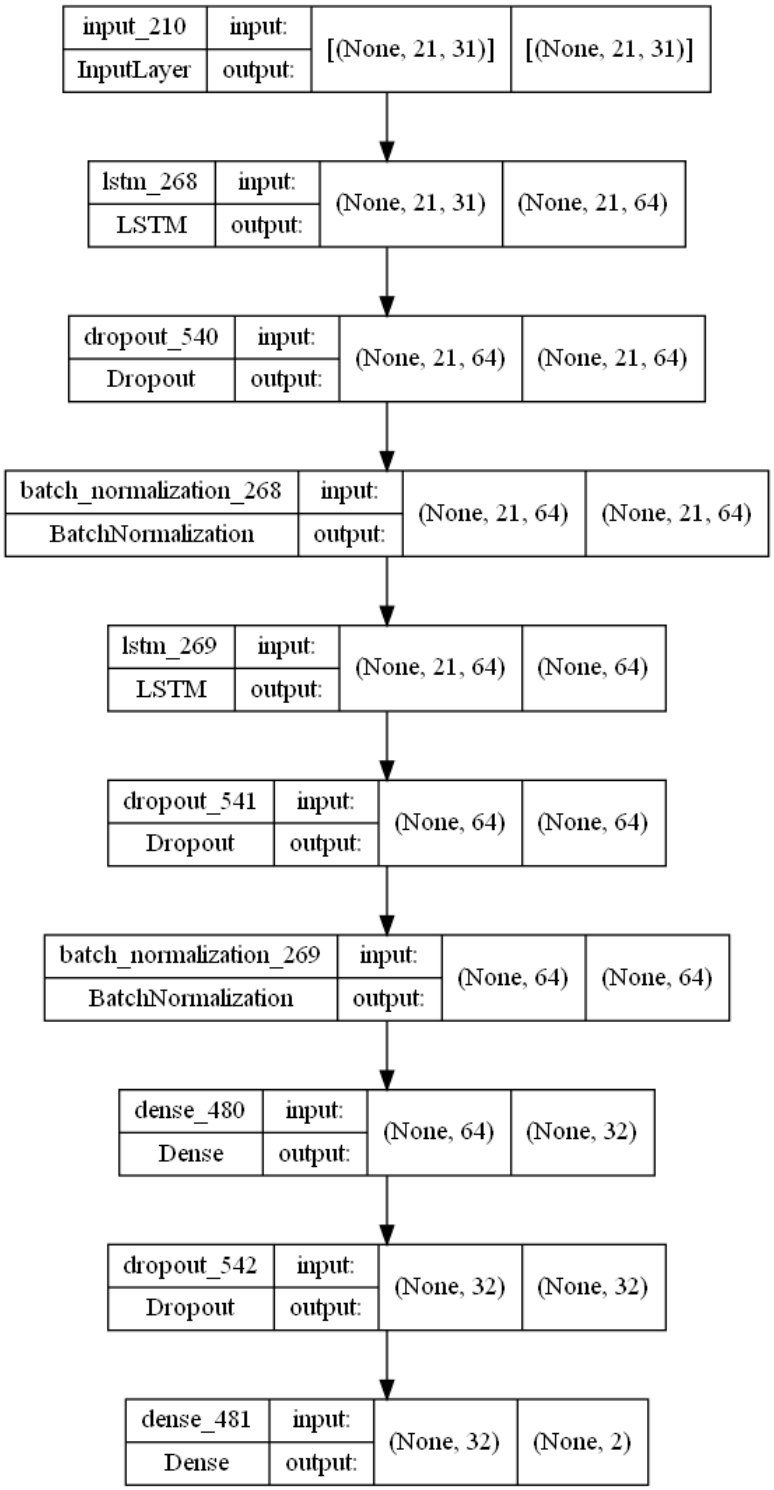

Figure 5.2: Diagram of iteration 1 layers

5.9 Iteration 2

5.9.1 Artificial intelligence model used

In the second iteration, the CNN model was utilised in order to create the foundations for the CNN-LSTM model. The CNN consists of Conv1D layers and Dense layers, with the final Dense layer having an output of 2, representing one output for a negative day (0) and another for a positive day (1), and utilised the 'softmax' activation function to predict the probabilities of each output. The Conv1D and Dense layers utilised the 'relu' (REctified Linear Unit) activation function.

5.9.2 Sequence length used

Contrary to the LSTM model, the CNN model does not use a sequence of historic data; it only uses the previous day's data as the input to the model.

5.9.3 Input features

The input features for this model included all the input features mentioned in Table 5.1.

5.9.4 Layers and layer sizes

Various amounts of Conv1D layers and Dense layers, as well as layer sizes for each were tested. The accuracy of each model varied significantly. For each Conv1D layer, there was also a MaxPooling1D layer following it; this created a downsampled feature map which aids in optimising the model's efficiency.

Following the Conv1D layers, there was also a Flatten layer applied to ensure the the following Dense layers were able to accept the shape of the output of the Convolutional layers.

The following models were created and compared with the following variables:

Layer Type	Number of Layers	Layer Size
Conv1D	1, 2	16, 32
Dense	1, 2	32, 64

Table 5.3: Table showing layers and layer sizes for Iteration 2

5.9.5 Diagram of model used

The diagram below (Figure 5.3) shows the best model identified in iteration 2 of this project. There is one Conv1D layer with a layer size of 32; and one Dense layer with a layer size of 2. For the purposes of this diagram, the 'None' value of each layer's shape can be ignored.

From the diagram, it is shown that the input shape the first Conv1D layer takes: (None, 1, 31); this represents the fact there are 31 input features. This Conv1D layer has an output shape of (None, 1, 32); it is fairly abstract but represents that there are 32 neurons in the next layer. This output is then passed to the MaxPooling1D Layer which downsamples it to create a feature map. A Flatten layer is applied to mutate the shape from (None, 1, 32) to (None, 32) before transitioning to the final Dense layer that has an output shape of (None, 2) which represents the fact there are two potential outputs for the inputs: whether the following day is an positive or negative trading day.

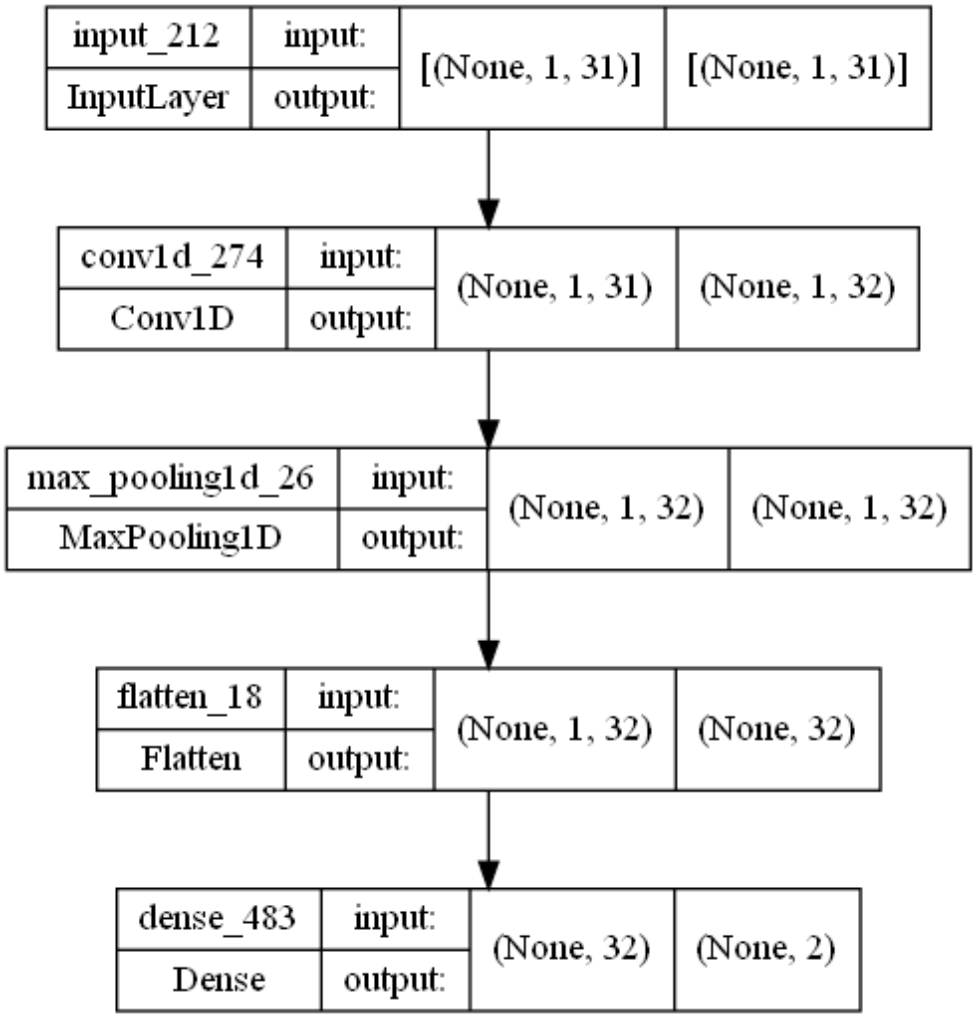

Figure 5.3: Diagram of iteration 2 layers

5.10 Iteration 3

5.10.1 Artificial intelligence model used

In the third iteration, a hybrid approach of the previous models were used to create a CNN-LSTM model with the layers applied sequentially. In this case Convolutional (specifically, Conv1D) layers were applied first and then LSTM layers in the sequential model. This model consists of Conv1D layers, LSTM layers and Dense layers with the final Dense layer having an output of 2,

representing one output for a negative day (0) and another for a positive day (1), and utilised the 'softmax' activation function to predict the probabilities of each output.

5.10.2 Sequence length used

Similarly to the LSTM model, this CNN-LSTM hybrid model does take a sequence as an input.

A sequence length of 21 days (approximately 1 month) was used for each of the input features of this model. This is a figure that has not in this iteration been tested to be the optimal sequence length, but with a short test of various sequence lengths, it has been found to be sufficient to produce values over 50% accuracy.

5.10.3 Input features

The input features for this model included all of the input features mentioned in Table 5.1.

5.10.4 Layers and layer sizes

Various amounts of Conv1D layers, LSTM layers and Dense layers, as well as layer sizes for each were tested. The accuracy of each model varied significantly. For each LSTM layer, there was also a:

- Dropout layer of 0.4 meaning a set of neurons may be deactivated 40% of the time at random; it is a method of regularisation to prevent overfitting.
- Batch normalisation layer meaning the outputs of that layer were scaled to aid efficiency of the model.

The following models were created and compared with the following variables:

Layer Type	Number of Layers	Layer Size
Conv1D	1, 2	16, 32
LSTM	1, 2	16, 32
Dense	2, 3	32, 64

Table 5.4: Table showing layers and layer sizes for Iteration 3

5.10.5 Diagram of model used

The diagram below (Figure 5.4) shows the best model identified in iteration 3 of this project. There are two Conv1D layers with a layer size of 32, 1 LSTM layer with a layer size of 16 and two Dense layers, one with a layer size of 64 and the final Dense layer having an output of 2, representing one output for a negative day (0) and another for a positive day (1), and utilised the 'softmax' activation function to predict the probabilities of each output.

For the purposes of this diagram, the 'None' value of each layer's shape can be ignored.

From the diagram, it is shown that the input shape the first Conv1D layer takes: (None, 21, 31); this represents the fact there is a sequence length of 21 and there are 31 input features. This Conv1D layer has an output shape of (None, 21, 32); it is fairly abstract but represents that a sequenced is passed to the next layer which has 32 neurons. This output is then passed to another Conv1D layer with the same Input and Output shapes. Following the Convolutional layers, there is an LSTM layer which takes the input (None, 21, 32). It does not return a sequence and also reduces the layer size to 16 to produce an output shape of (None, 16). Dropouts and Batch Normalisation layers are added to aid with regularisation and optimisation of the model. Following the LSTM layers, there are Dense layers, the first of which takes the input of shape (None, 16) from the LSTM layers and outputs a shape of (None, 64). Another Dropout layer is applied before transitioning to the final Dense layer that has an output shape of (None, 2) which represents the

fact there are two potential outputs for the inputs: whether the following day is an positive or negative trading day.

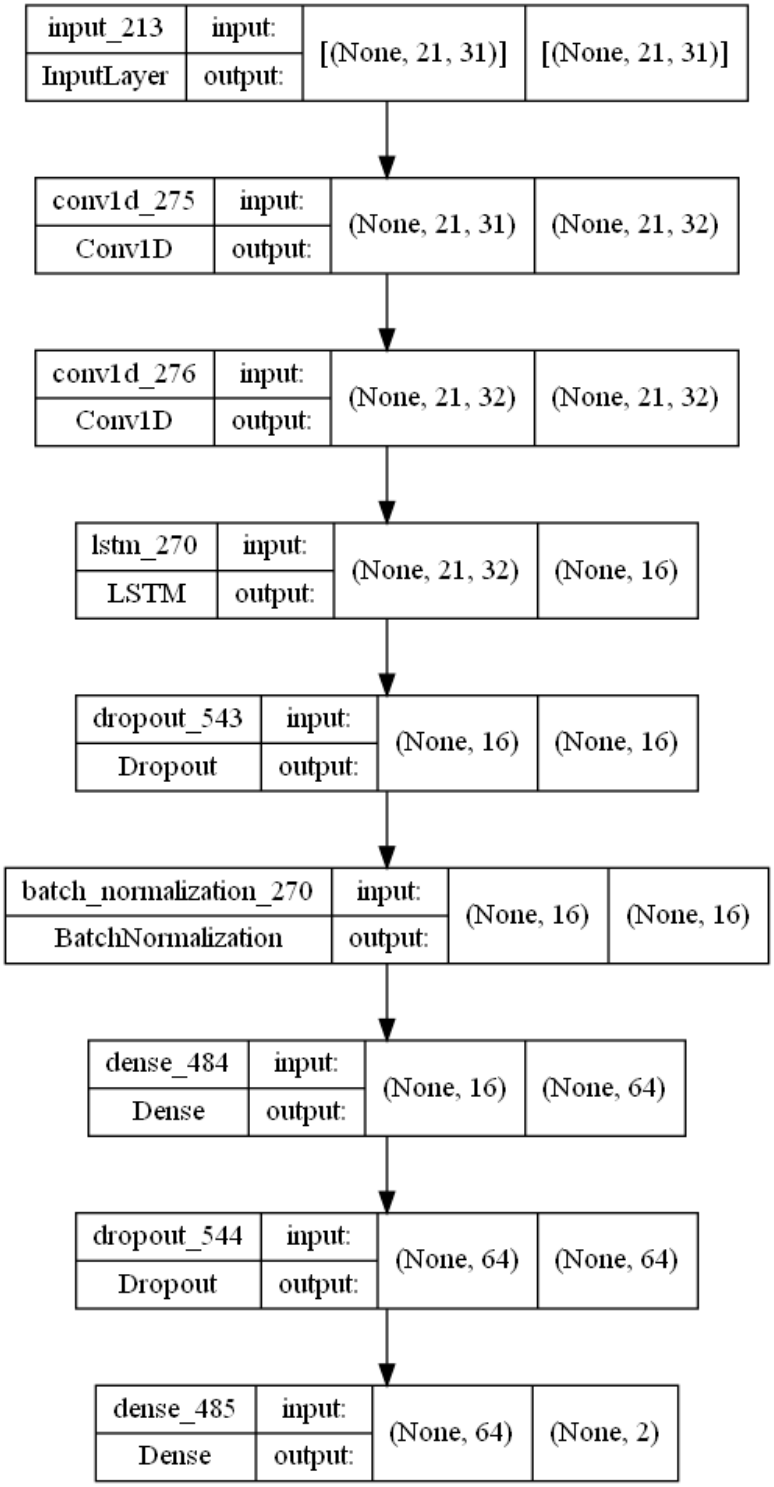

Figure 5.4: Diagram of iteration 3 layers

5.11 Iteration 4

5.11.1 Artificial intelligence model used

In the fourth iteration, a hybrid approach of the previous models was used to create a concatenated model of each of the three previous iterations in parallel. The layer amounts and sizes of each previous model were not modified, apart from the final Dense layer being removed from each model. After the three models were concatenated in parallel, an additional Dense layer was added to the model, and then the final Dense layer of output shape (None, 2) was returned to the model.

5.11.2 Diagram of model used

The diagram below (Figure 5.5) shows the concatenation of the previous three models (LSTM, CNN, CNN-LSTM) The Concatenate layer accepts the outputs from each of the three models and as such has an input shape of [(None, 32), (None, 32), (None, 64)]. The Concatenate layer has an output of shape (None, 128) which is passed to a Dense layer which outputs a shape of (None, 32) which is passed to the final layer which has an output shape of (None, 2).

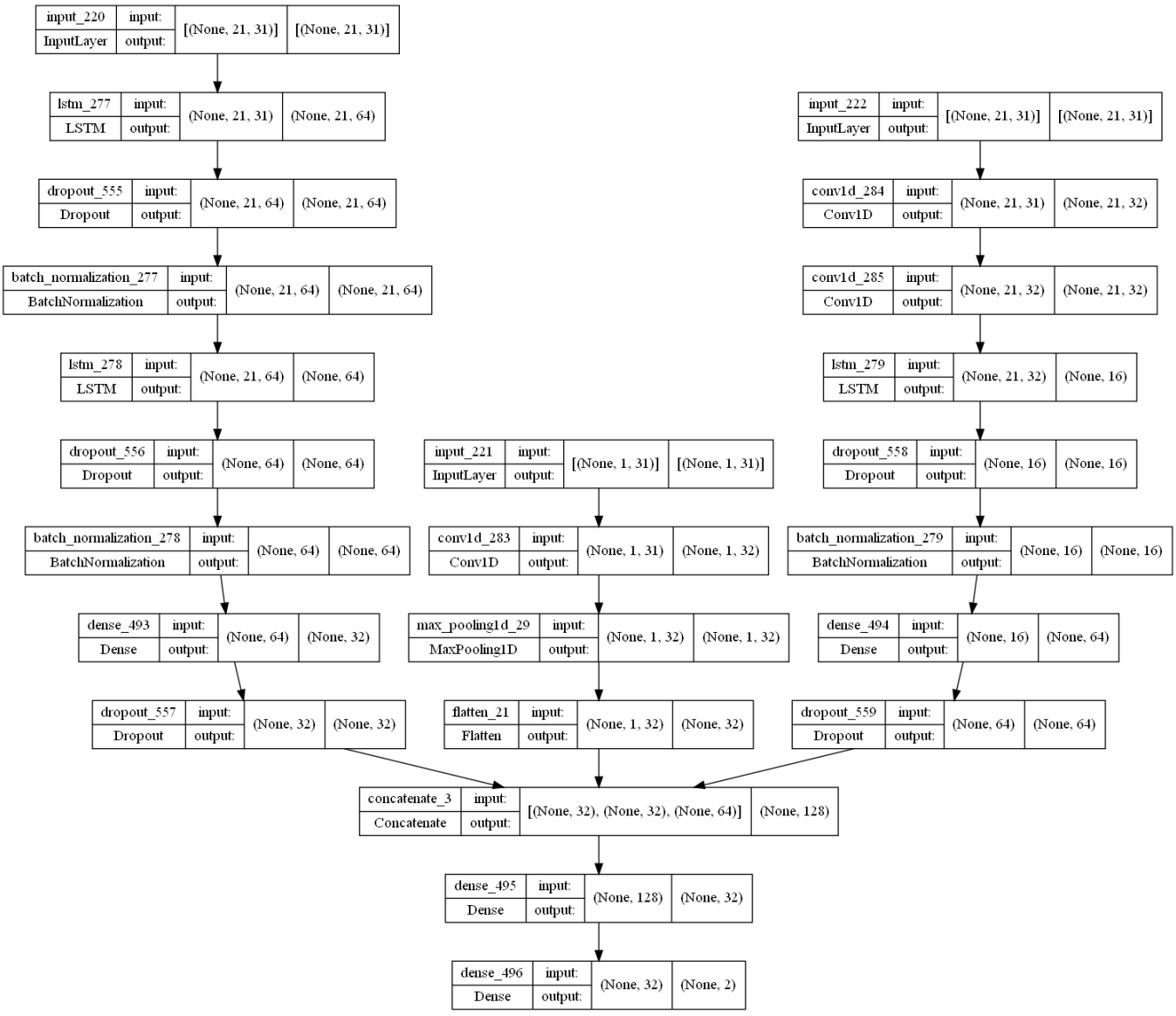

Figure 5.5: Diagram of iteration 4 layers

5.12 Iteration 5

This iteration is the final iteration for this project. It utilises the best model identified across the previous four iterations: the CNN-LSTM sequential model from Iteration 3. Different sets of input features and sequence lengths are utilised to help answer the research questions.

5.12.1 Artificial intelligence model used

In the fifth iteration, the previous four iterations were compared to identify which model had the best results. This was found to be the CNN-LSTM hybrid model from subsection 5.10.1 with an accuracy of 56.07%.

5.12.2 Sequence length used

Various sequence lengths were used in the testing of this model, including: previous two days, previous week (5 trading days), previous month (21 trading days), and previous quarter (previous 63 trading days). These all affect the input shape to the neural network, but layers following the input layer are unmodified from what can be seen in Figure 5.4 of Iteration 3.

5.12.3 Input features

Various combinations of input features were used in this iteration. All of the combinations of input features included the price percentage change vs the previous day. The combinations of input features are as follows:

- Price only
  - SPY price percentage change vs previous day
- Price extended set
  - SPY price percentage change vs previous day
  - SPY price percentage change vs previous 2 days

  - SPY price percentage change vs previous week
  - SPY price percentage change vs previous month
  - SPY price percentage change vs previous quarter
- Price + Volume
  - SPY price percentage change vs previous day
  - SPY volume percentage change vs previous day
- Price + Volatility
  - SPY price percentage change vs previous day
  - VIX (Volatility Index) percentage change vs previous day
- Price + M1 Money Supply
  - SPY price percentage change vs previous day
  - M1 Money Supply percentage change vs previous week
- Price + GDP
  - SPY price percentage change vs previous day
  - GDP percentage change vs previous quarter
- Price + Treasury Yields
  - SPY price percentage change vs previous day
  - 1 Month treasury yield percentage change vs previous day
  - 3 Month treasury yield percentage change vs previous day
  - 1 Year treasury yield percentage change vs previous day
  - 2 Year treasury yield percentage change vs previous day
  - 5 Year treasury yield percentage change vs previous day
  - 10 Year treasury yield percentage change vs previous day

  - 20 Year treasury yield percentage change vs previous day
  - 30 Year treasury yield percentage change vs previous day
- Price + Effective Federal Funds Rate (EFFR)
  - SPY price percentage change vs previous day
  - Effective Federal Funds Rate percentage change vs previous day
- Price + Repurchase Agreements
  - SPY price percentage change vs previous day
  - Repurchase Agreement Utilisation percentage change vs previous day
  - Repurchase Agreement Rate percentage change vs previous day
- Price + Reverse Repurchase Agreements
  - SPY price percentage change vs previous day
  - Reverse Repurchase Agreement Utilisation percentage change vs previous day
  - Reverse Repurchase Agreement Rate percentage change vs previous day
- Price + Gold
  - SPY price percentage change vs previous day
  - Gold Price percentage change vs previous day
- Price + Currency Exchange
  - SPY price percentage change vs previous day
  - USDGBP percentage change vs previous day
  - USDEUR percentage change vs previous day
  - USDJPY percentage change vs previous day

- Price + Options (Put-to-Call Ratio)
  - SPY price percentage change vs previous day
  - Put-to-Call ratio percentage change vs previous day
- Price + Inflation Rate
  - SPY price percentage change vs previous day
  - Inflation rate percentage change vs previous month
- Price + Employment Rate
  - SPY price percentage change vs previous day
  - Employment rate percentage change vs previous month
- Price + Other Sentiment Indicators
  - SPY price percentage change vs previous day
  - Consumer Sentiment (University of Michigan) percentage change vs previous month
  - Consumer Confidence (OECD) percentage change vs previous month

5.13 Benchmark application

This iteration is the final iteration for this project. It utilises the same model as specified in iterations 5 and 3. The main difference is regarding the set of input features which are described in subsection 5.13.3.

As a result, this model represents a benchmark application of the same existing research method (the CNN-LSTM sequential model) that is used with an existing case study.

5.13.1 Artificial intelligence model used

In the benchmark application, the CNN-LSTM sequential model is utilised as this is the one chosen in Iteration 5 after it had been found to be the best model in Iteration 3.

5.13.2 Sequence length used

Similarly to the models in the artefact, various sequence lengths were used in the testing of this model, including: previous two days, previous week (5 trading days), previous month (21 trading days), and previous quarter (previous 63 trading days). These all affect the input shape to the neural network, but layers following the input layer are unmodified from what can be seen in Figure 5.4 of Iteration 3.

5.13.3 Input features

Similarly to the artefact models, the benchmark dataset includes data from November 2006 to March 2022; the dataset is split 80:20 between training data and validation data. Furthermore, the preprocessing is applied in a similar manner.

The following input features are used in this model:

- SPY daily open price percentage change vs previous day
- SPY daily high price percentage change vs previous day
- SPY daily low price percentage change vs previous day
- SPY daily closing price percentage change vs previous day
- SPY daily volume percentage change vs previous day

The daily volume and closing price have been utilised in previous models for the artefact, but the open, high and low prices are unique to the benchmark application. This dataset was chosen as it has been found to be a popu-

lar dataset in previous studies (Fitriyaningsih et al., 2019; Hu et al., 2018; Thakkar & Chaudhari, 2020).

Chapter 6

Results and evaluation

The primary metric for the results of this project is the validation accuracy of the models. Validation accuracy is calculated as the amount of correct predictions divided by the total number of predictions in the validation dataset as can be seen in the equation below:

An additional metric specified is the loss of each model, which has been described in section 5.5 and was utilised to detect overfitting in the various models tested.

The results have also been compared to a benchmark application using an existing case study to the best performing existing research method as found in the results.

6.1 Table of all results

Iteration	Research Method	Best Validation Accuracy
Iteration 1	LSTM	55.54%
Iteration 2	CNN	53.57%
Iteration 3	CNN-LSTM (sequential)	56.07%
Iteration 4	LSTM + CNN + CNN-LSTM (parallel)	53.21%
Iteration 5	CNN-LSTM (sequential)	57.50%
Benchmark	CNN-LSTM (sequential)	54.55%

Table 6.1: Table showing the best model's validation accuracy of each iteration

6.2 Iteration 1 results

As mentioned in subsection 5.8.4, various combinations of LSTM layers and Dense layers were tested, each with various sizes. The results of all the models can be seen in Figure 6.1 and Figure 6.3.
The results of the best model can be seen in Figure 6.5.

6.2.1 Accuracies of all tests

In the charts below, each line represents a combination of different layer types and sizes of the Dense and LSTM layers of the chart. For both the training and validation accuracy charts, each line has a corresponding line in the loss charts with the same colour. There are 16 lines in total for this iteration as there were 2 LSTM layer amounts, 2 Dense layer amounts, 2 LSTM layer sizes, and 2 Dense layer sizes in the tests.

Training accuracies and losses

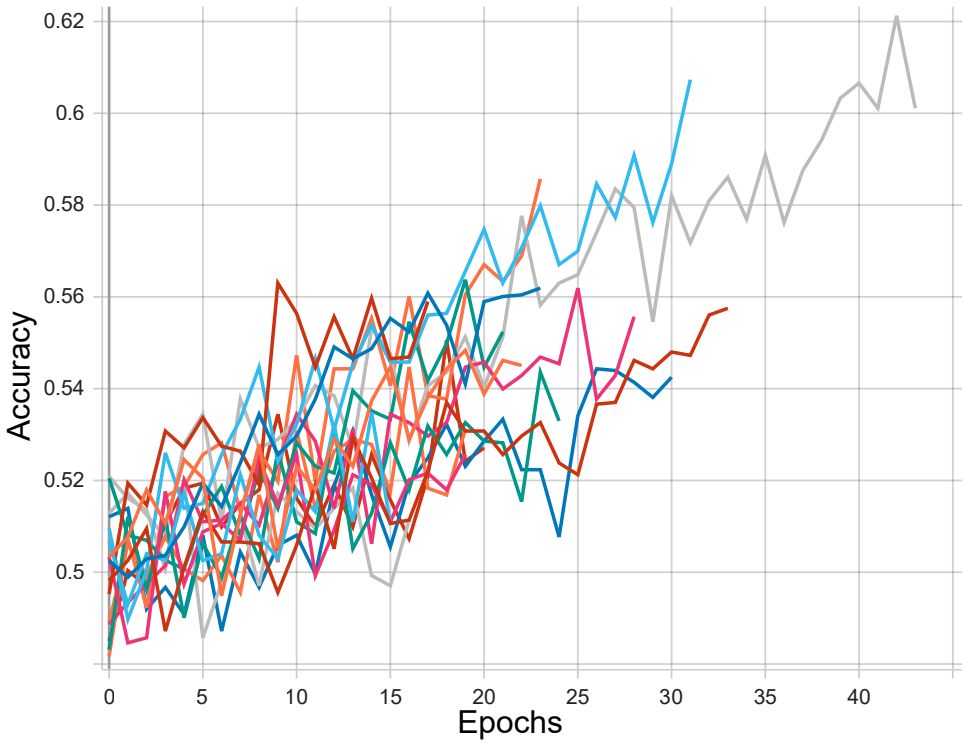

Figure 6.1: Figure of all training accuracies of the combinations of LSTM Layers and Dense Layers in Iteration 1

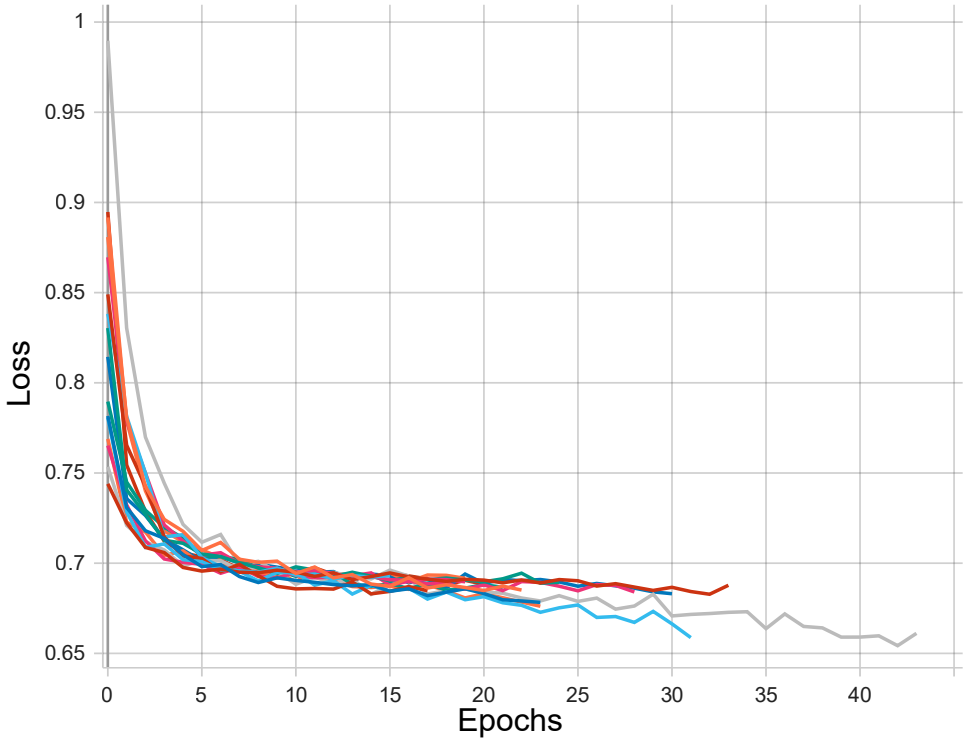

Figure 6.2: Figure of all training losses of the combinations of LSTM Layers and Dense Layers in Iteration 1

Validation accuracies and losses

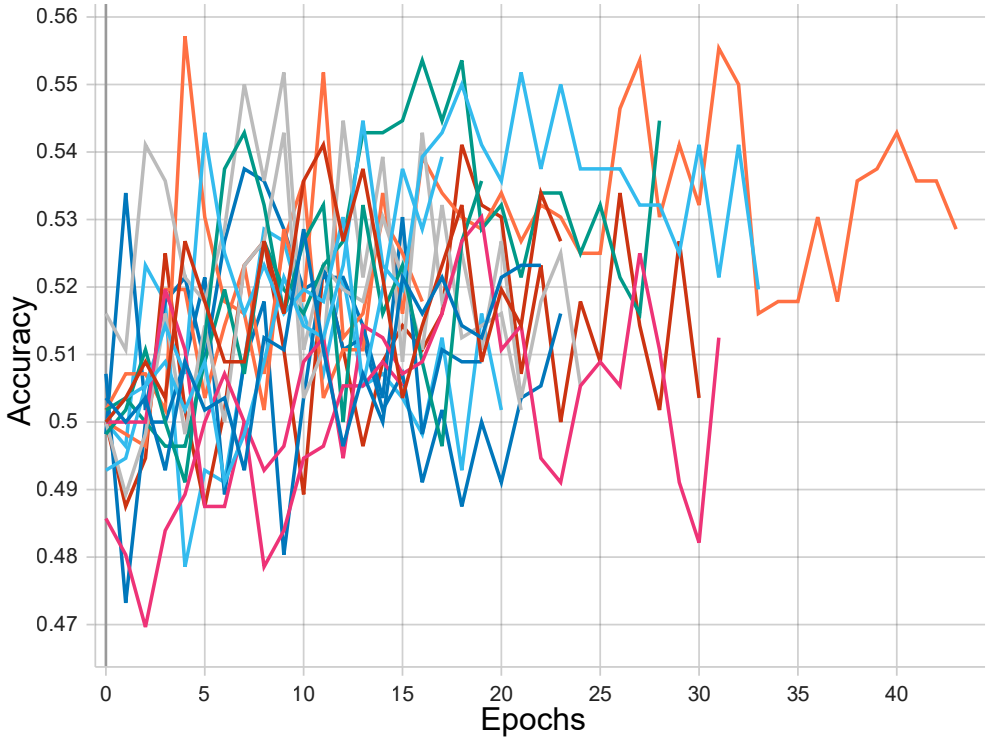

Figure 6.3: Figure of all validation accuracies of the combinations of LSTM Layers and Dense Layers in Iteration 1

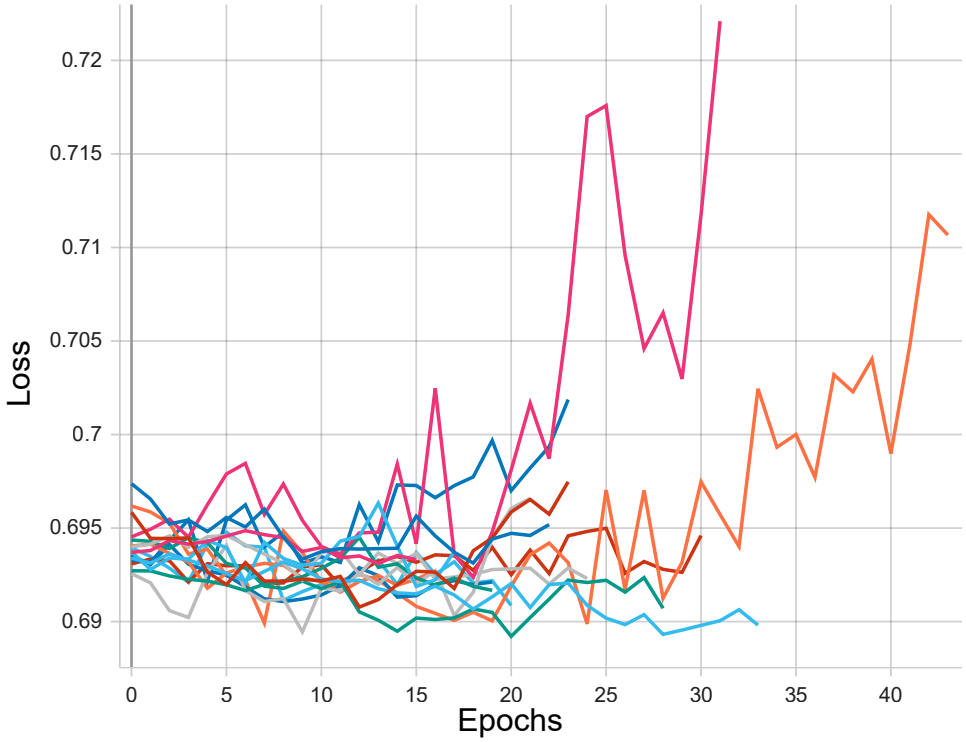

Figure 6.4: Figure of all validation losses of the combinations of LSTM Layers and Dense Layers in Iteration 1

There are varying degrees of success with different combinations of layers; some do not improve in accuracy and others do. The validation losses using the sparse categorical loss function as described earlier in section 5.5 were used to help identify which models had the lowest error rate; and as an increasing validation loss is an indicator of overfitting, it was used to filter models showing overfitting.

6.2.2 Accuracies of the best result

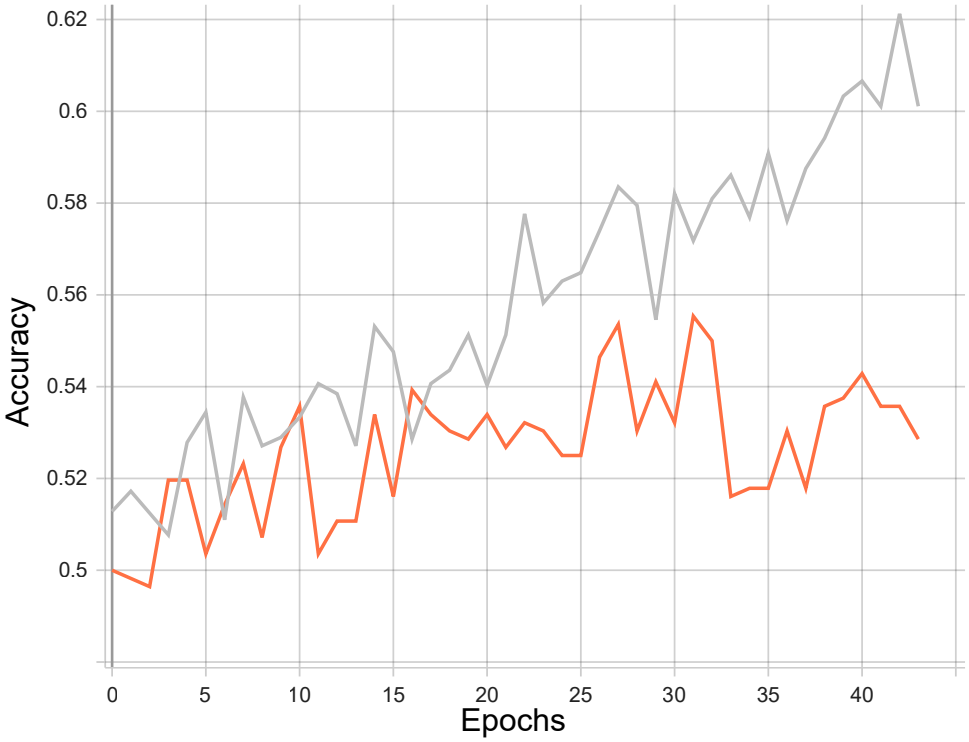

Figure 6.5: Figure of the best accuracy of the combinations of LSTM Layers and Dense Layers in Iteration 1

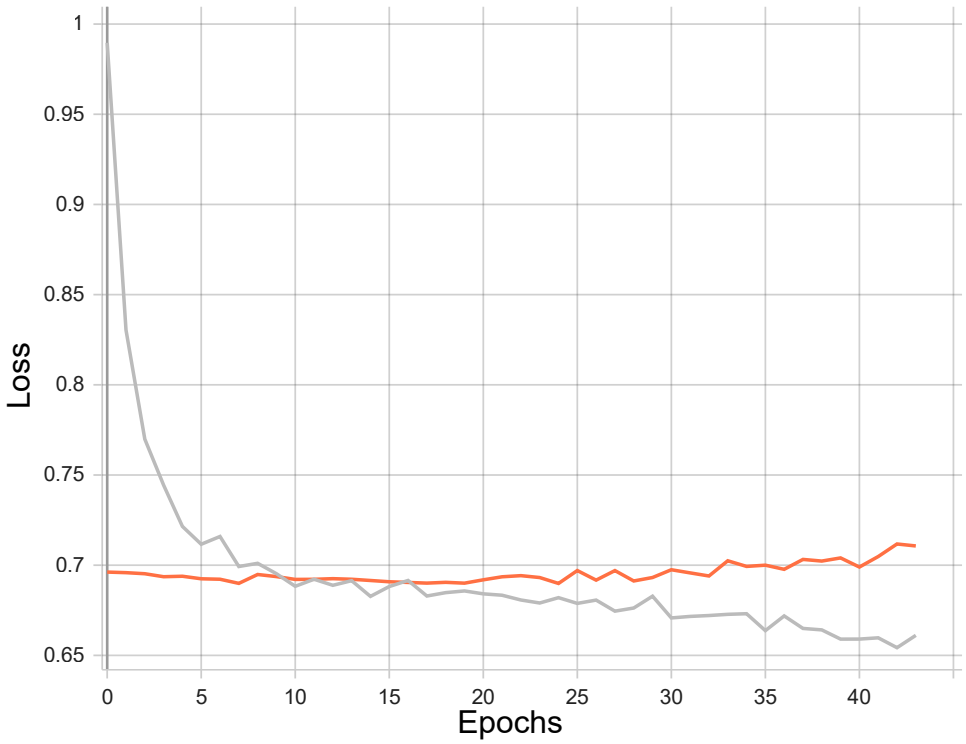

Figure 6.6: Figure of the best loss of the combinations of LSTM Layers and Dense Layers in Iteration 1

The best model found within this iteration was one with a two LSTM layers of size 64 followed by the final Dense layer of size 2. The grey line represents the training set and the orange line represents the validation set. The validation loss shows little overfitting as can be seen in Figure 6.6. At 31 epochs, the validation accuracy reaches its highest value of 55.54% whilst the training accuracy reached above 57.18%.

6.2.3 Evaluation of iteration 1

With a validation accuracy above 55% it suggests that there is improvement above a random choice. It forms a good basis for testing further AI models, specifically the CNN-LSTM hybrid model. There are various limitations regarding iteration 1. This iteration does not test various combinations of input features nor does it test various sequence lengths. Furthermore, there could be another number of layers or layer sizes that is better suited to the problem, but unfortunately many have not been tested due to time constraints.

6.3 Iteration 2 results

As mentioned in subsection 5.9.4, various combinations of Conv1D layers and Dense layers were tested, each with various sizes. The results of all the models can be seen in Figure 6.7 and Figure 6.9.
The results of the best model can be seen in Figure 6.11.

6.3.1 Accuracies of all tests

In the charts below, each line represents a combination of different layer types and sizes of the Dense and LSTM layers of the chart. For both the training and validation accuracy charts, each line has a corresponding line in the loss charts with the same colour. There are 16 lines in total for this iteration as there were 2 Conv1D layer amounts, 2 Dense layer amounts, 2 Conv1D layer sizes, and 2 Dense layer sizes in the tests.

Training accuracies and losses

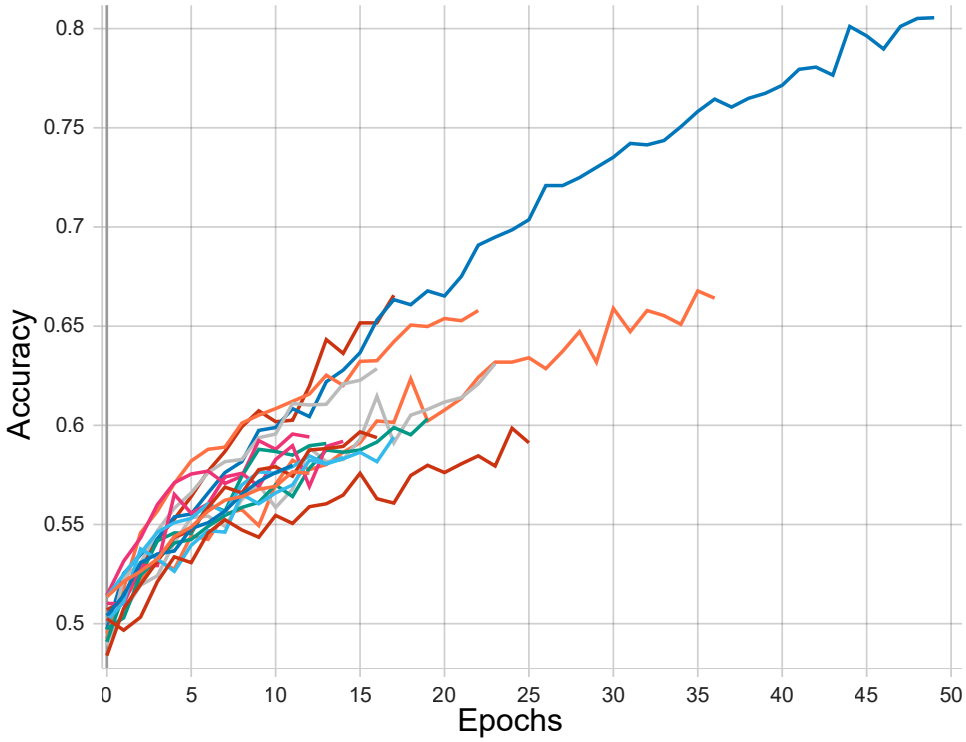

Figure 6.7: Figure of all training accuracies of the combinations of Conv1D Layers and Dense Layers in Iteration 2

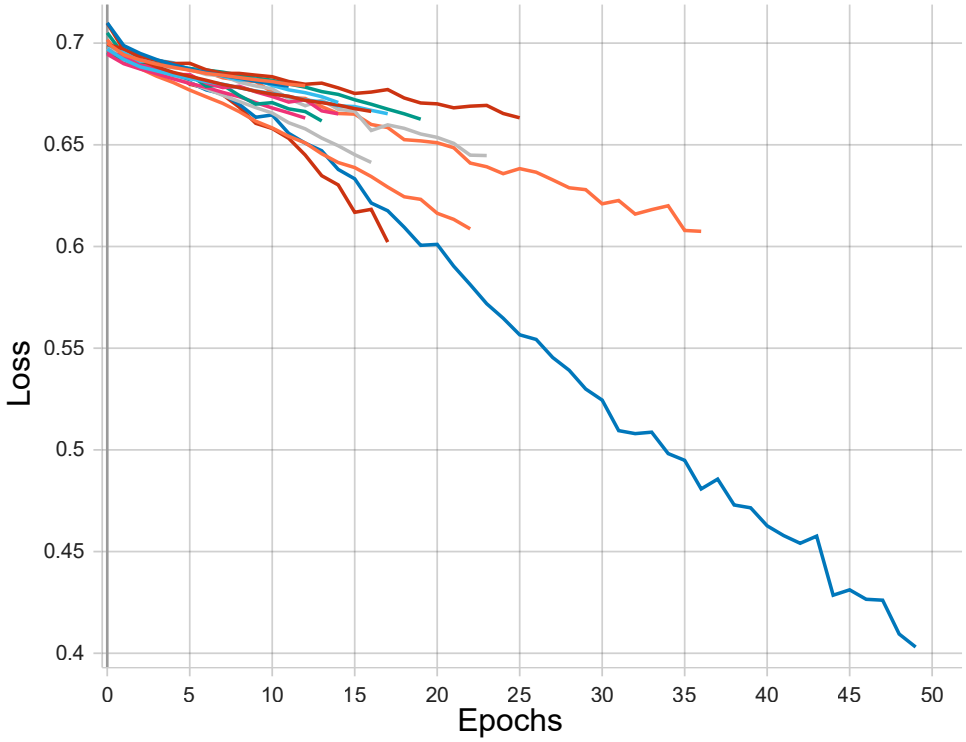

Figure 6.8: Figure of all training losses of the combinations of Conv1D Layers and Dense Layers in Iteration 2

Validation accuracies and losses

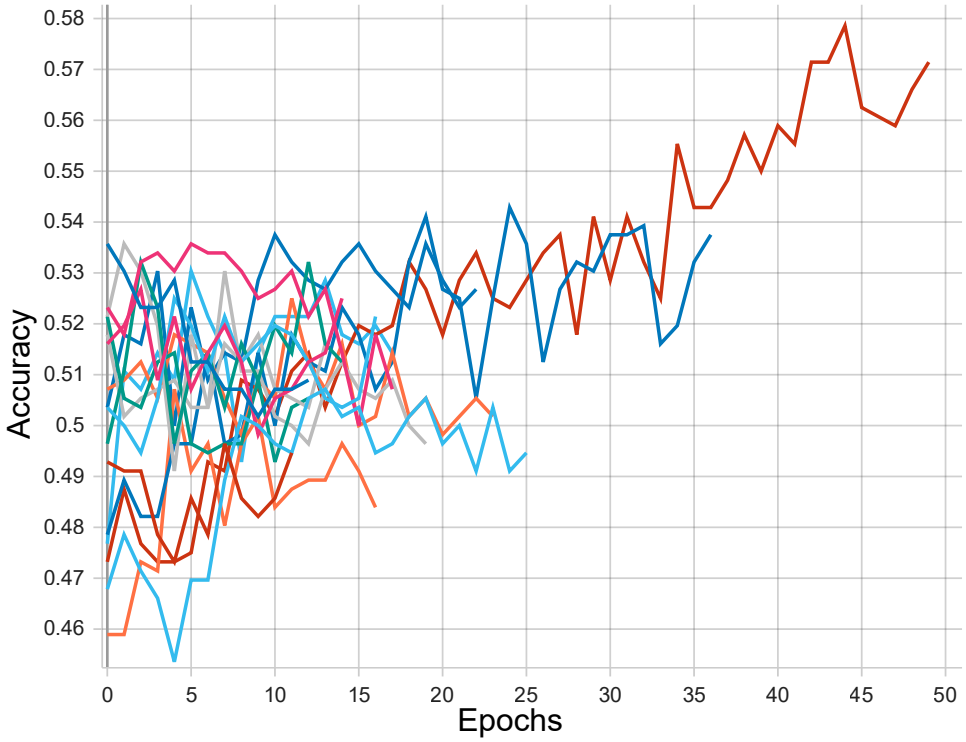

Figure 6.9: Figure of all validation accuracies of the combinations of Conv1D Layers and Dense Layers in Iteration 2

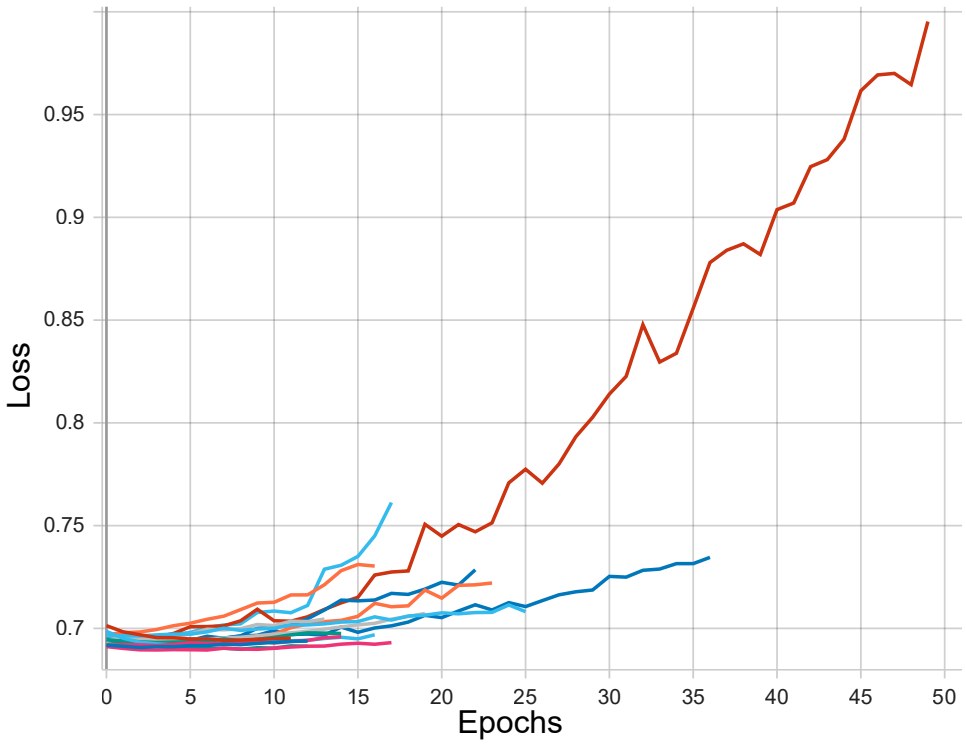

Figure 6.10: Figure of all validation losses of the combinations of Conv1D Layers and Dense Layers in Iteration 2

There are varying degrees of success with different combinations of layers; some do not improve in accuracy and others do. The validation losses using the sparse categorical loss function as described earlier in section 5.5 were used to help identify which models had the lowest error rate; and as an increasing validation loss is an indicator of overfitting, it was used to filter models showing overfitting.

6.3.2 Accuracies of the best result

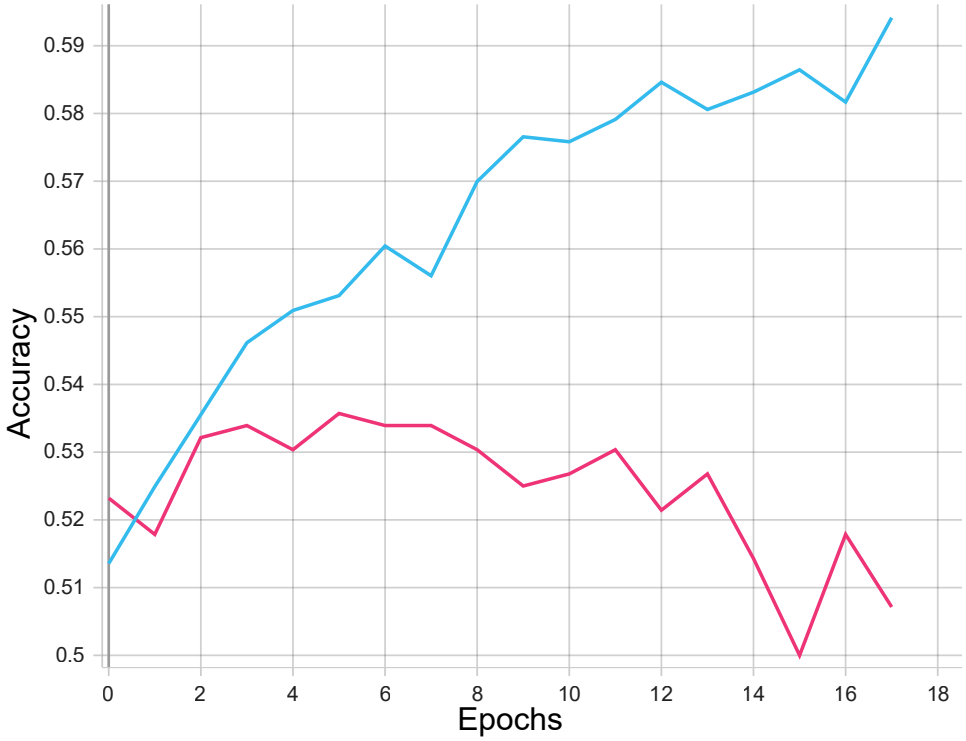

Figure 6.11: Figure of the best accuracy of the combinations of Conv1D Layers and Dense Layers in Iteration 2

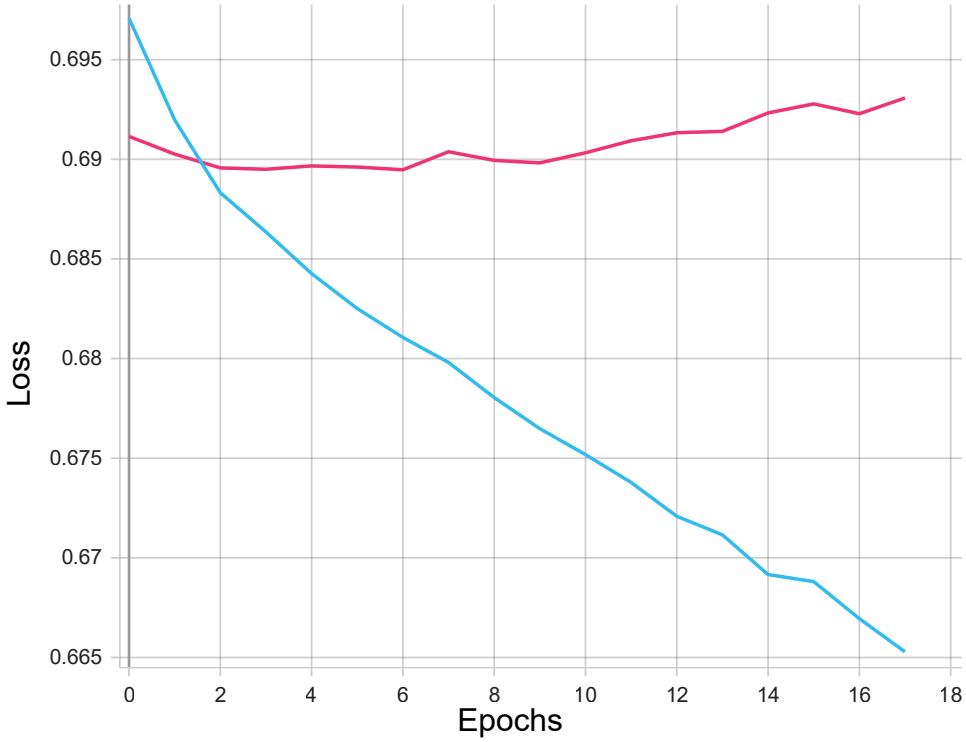

Figure 6.12: Figure of the best loss of the combinations of Conv1D Layers and Dense Layers in Iteration 2

The best model found within this iteration was one with a one CNN layer of size 32 followed by the final Dense layer of size 2. The blue line represents the training set and the pink line represents the validation set. The validation loss shows little overfitting as can be seen in Figure 6.12. At 5 epochs, the validation accuracy reaches its highest value of 53.57% whilst the training accuracy reached above 55.31%.

6.3.3 Evaluation of iteration 2

With a validation accuracy above 53% it suggests that there is only minimal improvement above a random choice. It forms a good basis for testing further AI models, specifically the CNN-LSTM hybrid model. There are various limitations regarding iteration 2. This iteration does not test various combinations of input features nor does it test various sequence lengths.

Furthermore, there could be another number of layers or layer sizes that is better suited to the problem, but unfortunately many have not been tested due to time constraints.

6.4 Iteration 3 results

As mentioned in subsection 5.10.4, various combinations of Conv1D layers, LSTM layers and Dense layers were tested, each with various sizes. The results of all the models can be seen in Figure 6.13 and Figure 6.15.
The results of the best model can be seen in Figure 6.17.

6.4.1 Accuracies of all tests

In the charts below, each line represents a combination of different layer types and sizes of the Conv1D, LSTM and Dense layers. For both the training and validation accuracy charts, each line has a corresponding line in the loss charts with the same colour. There are 64 lines in total for this iteration as there were 2 Conv1D layer amounts, 2 LSTM layer amounts, 2 Dense layer amounts, as well as 2 Conv1D layer sizes, 2 LSTM layer sizes and 2 Dense layer sizes in the tests.

Training accuracies and losses

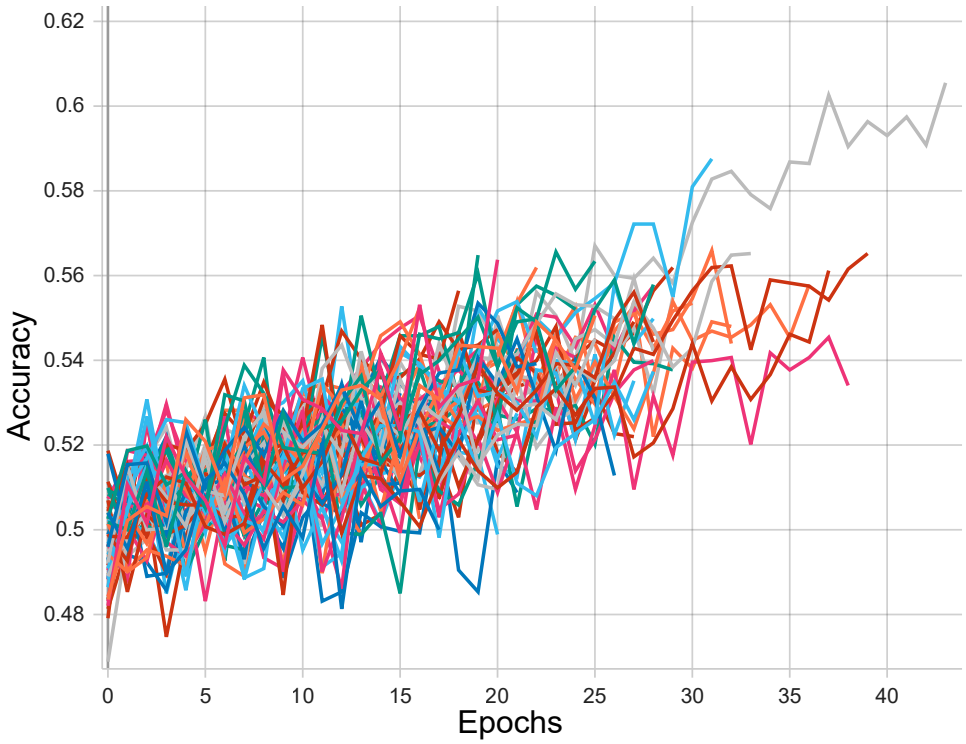

Figure 6.13: Figure of all training accuracies of the combinations of Conv1D, LSTM and Dense Layers in Iteration 3

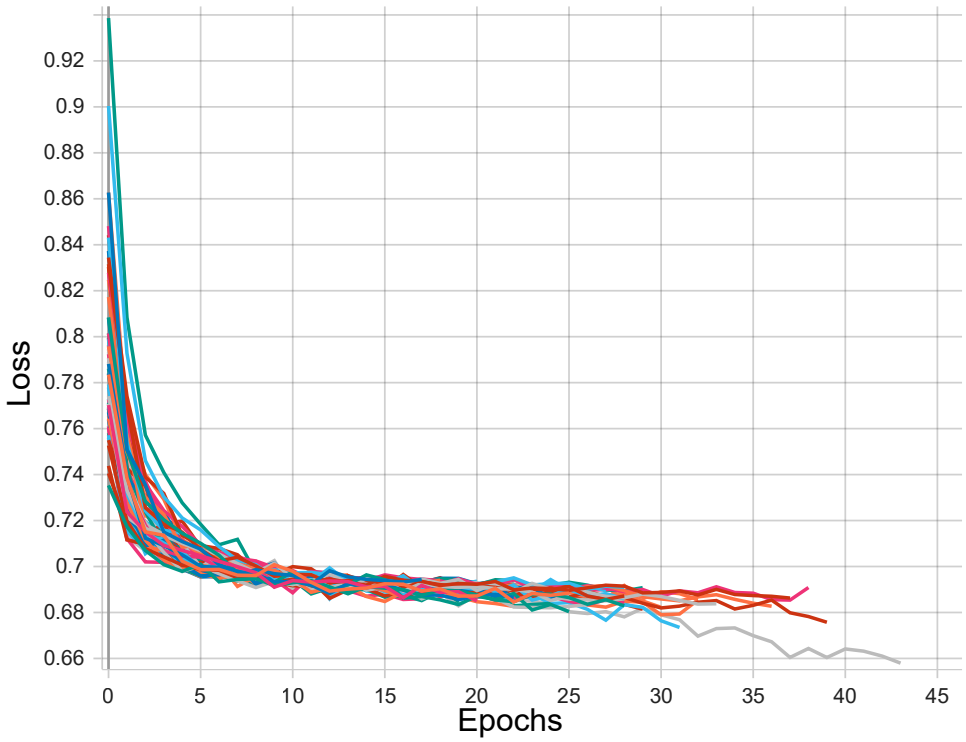

Figure 6.14: Figure of all training losses of the combinations of Conv1D, LSTM and Dense Layers in Iteration 3

Validation accuracies and losses

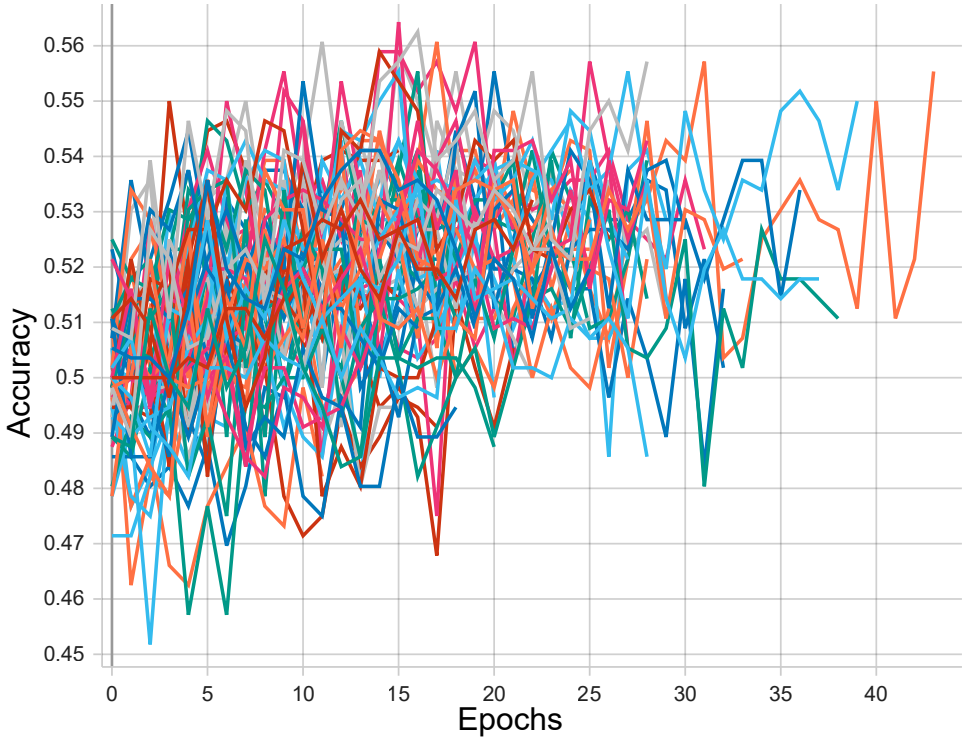

Figure 6.15: Figure of all validation accuracies of the combinations of Conv1D, LSTM and Dense Layers in Iteration 3

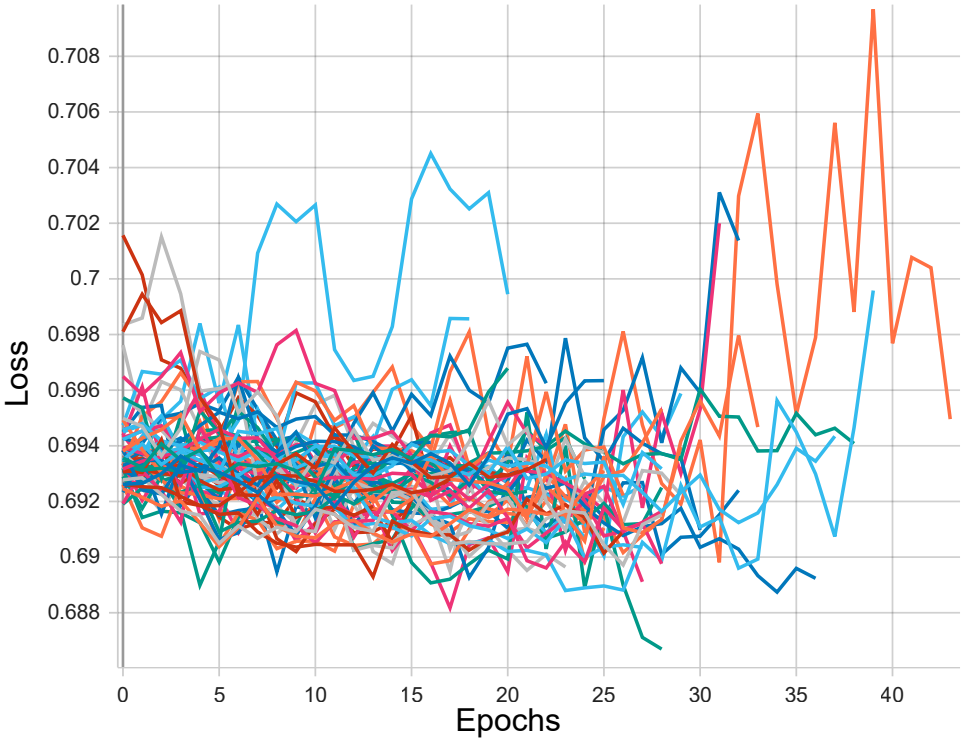

Figure 6.16: Figure of all validation losses of the combinations of Conv1D, LSTM and Dense Layers in Iteration 3

There are varying degrees of success with different combinations of layers; some do not improve in accuracy and others do. The validation losses using the sparse categorical loss function as described earlier in section 5.5 were used to help identify which models had the lowest error rate; and as an increasing validation loss is an indicator of overfitting, it was used to filter models showing overfitting.

6.4.2 Accuracies of the best result

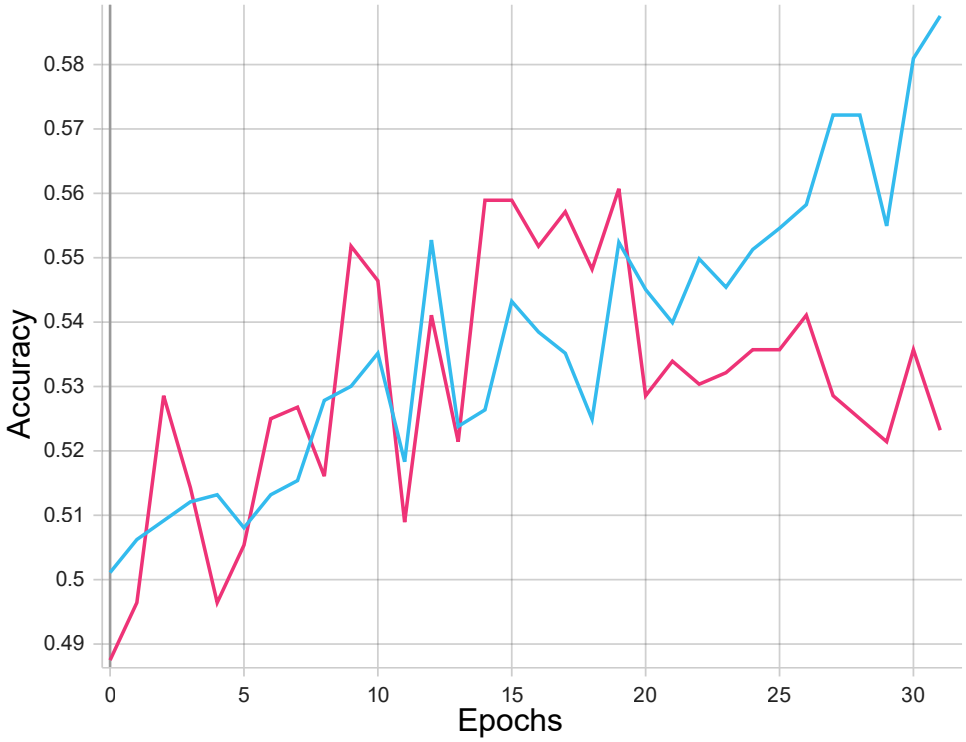

Figure 6.17: Figure of the best accuracy of the combinations of Conv1D, LSTM and Dense Layers in Iteration 3

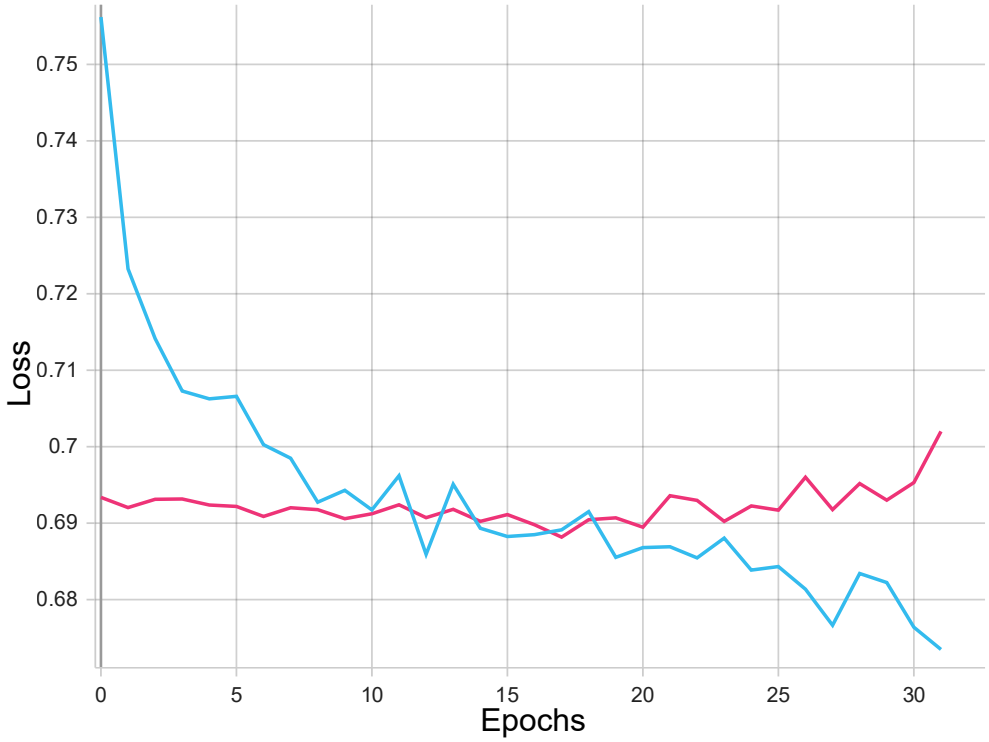

Figure 6.18: Figure of the best loss of the combinations of Conv1D, LSTM and Dense Layers in Iteration 3

The best model found within this iteration was one with a two Conv1D layers of size 32 followed by 1 LSTM layer of size 16, and a Dense layer of size 64 followed by a final Dense layer of size 2. The blue line represents the training set and the pink line represents the validation set. The validation loss shows little overfitting as can be seen in Figure 6.18. At 19 epochs, the validation accuracy reaches its highest value of 56.07% whilst the training accuracy reached 55.24%. The validation accuracy being somewhat greater than the training accuracy can be explained by the dropout layers within the model that can affect the results of the training set but does not affect the validation set.

6.4.3 Evaluation of iteration 3

With a validation accuracy above 56.07%, it is currently the best model when compared to the previous iterations. This suggests there is a significant advantage compared to randomly guessing the next trading day's direction. Furthermore, this aligns with the studies analysed in the literature review. However, this iteration alone cannot be used to answer the research questions as it does not test different sequence lengths or combinations of input features. Additionally, there could be a different number of layers or sizes of layers that perform better but unfortunately could not be tested due to time constraints.

6.5 Iteration 4 results

As mentioned in subsection 5.11.1, a concatenated model of the best models of the previous three iterations were combined in parallel. The results of this model can be seen in Figure 6.19.

6.5.1 Accuracies of iteration 4 model

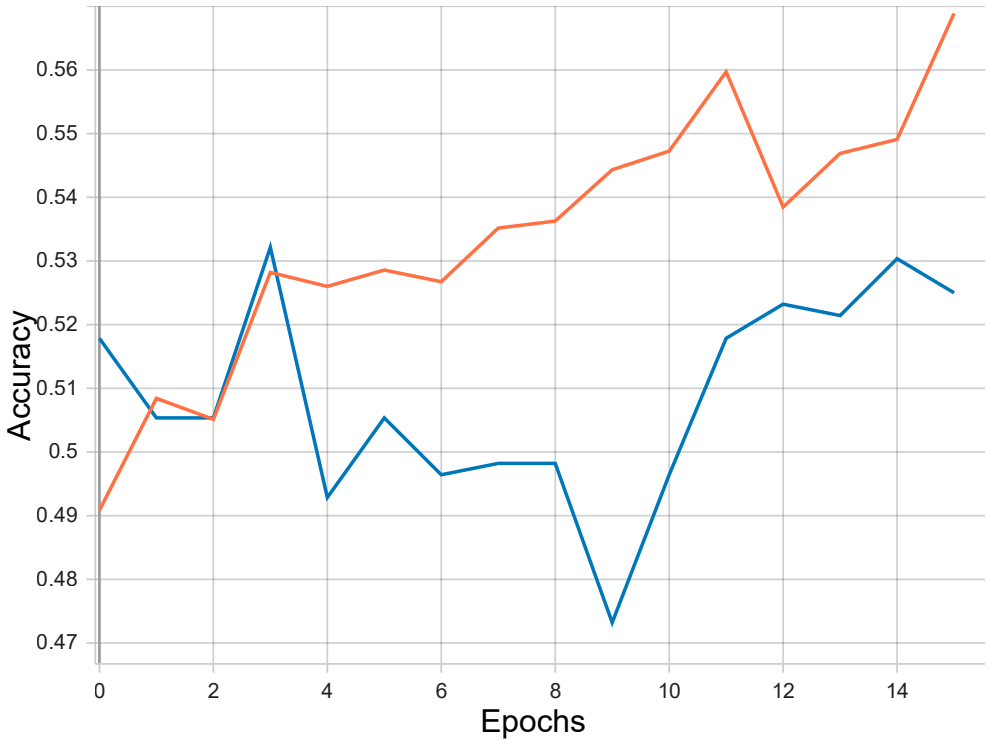

Figure 6.19: Figure of all training and validation accuracies in Iteration 4

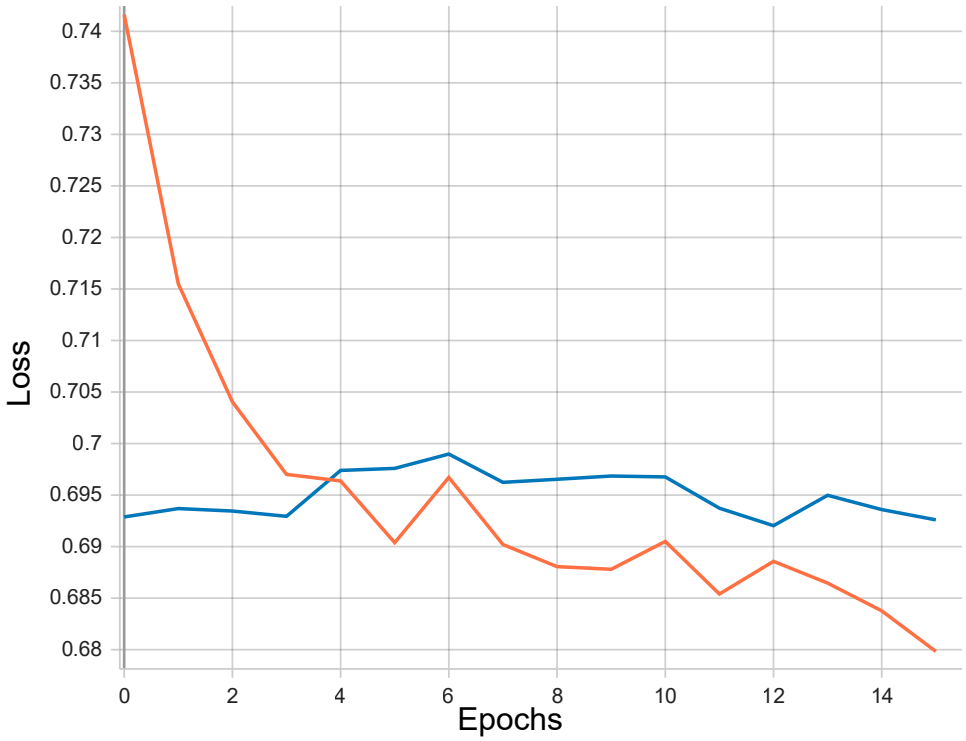

Figure 6.20: Figure of all training and validation losses in Iteration 4

The orange line represents the training set and the blue line represents the validation set. The validation loss shows little overfitting as can be seen in Figure 6.20 as the loss does not significantly increase. At 3 epochs, the validation accuracy reaches its highest value of 53.21% whilst the training accuracy reached 52.82%. The validation accuracy being somewhat greater than the training accuracy can be explained by the dropout layers within the model that can affect the results of the training set but does not affect the validation set.

6.5.2 Evaluation of iteration 4

With a validation accuracy above 53%, there is only a minimal improvement above a random guess of the direction of the next trading day. This model was included to test between a sequential hybrid model as seen in Iteration 3 and this parallel hybrid approach. However, this model does not offer any advantage over the CNN-LSTM sequential model alone; due to this, no further tests will be utilised with this model.

6.6 Iteration 5 results

This iteration primarily used a similar model to that in Iteration 3, however with varied input features and varied sequence lengths. There are various results that come from this model from identifying the best amount of historic data to identifying the most important inputs.

6.6.1 Accuracies of all tests

For each of the tests in Iteration 5, the charts contain 15 lines representing the different combinations of input features as mentioned in subsection 5.12.3. Each line in the accuracy chart has a corresponding line in the loss chart with the same colour. Each line in the training charts has a corresponding line in the validation charts with a different colour.

The best values of the validation accuracies of each model at any epoch are shown in subsection 6.6.2.

Two days of historic data accuracies and losses

Training accuracies and losses

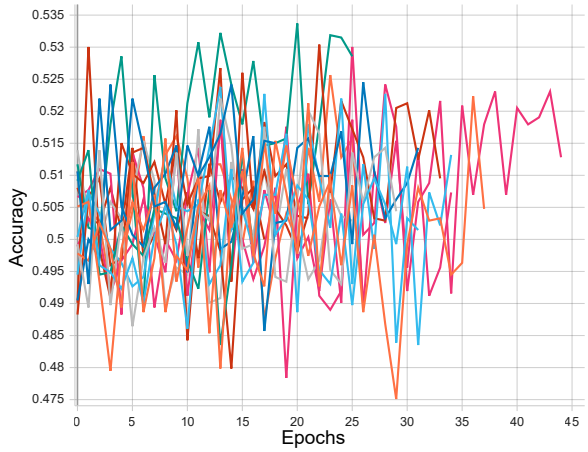

Figure 6.21: Figure of all training accuracies with two days of historic data in Iteration 5

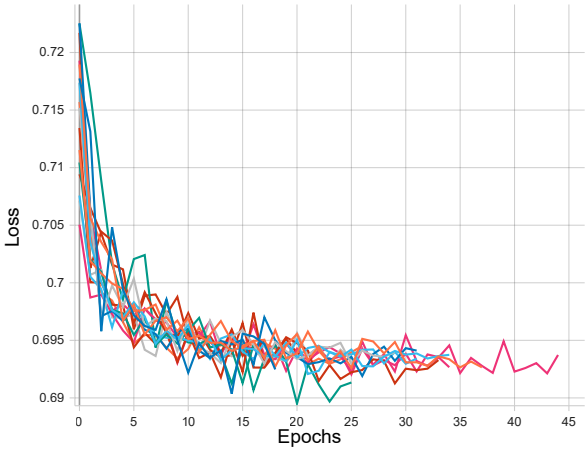

Figure 6.22: Figure of all training losses with two days of historic data in Iteration 5

Validation accuracies and losses

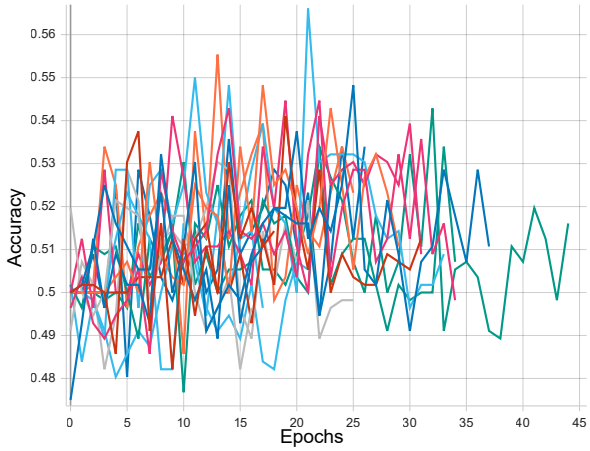

Figure 6.23: Figure of all validation accuracies with two days of historic data in Iteration 5

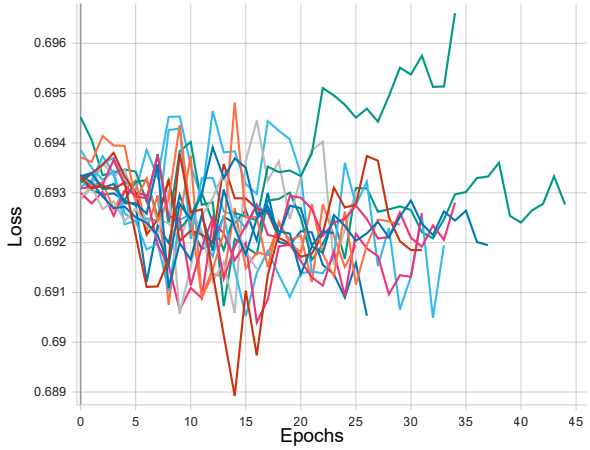

Figure 6.24: Figure of all validation losses with two days of historic data in Iteration 5

The greatest validation accuracy for the model with a sequence length of two days was 55.54% with the dataset using price and volatility data. While the chart above in Figure 6.24 may look erratic, the changes to the loss are minimal due to the scale as it only varies between 0.689 and 0.697. This suggests there is little overfitting as it does not increase significantly.

One week of historic data accuracies and losses

Training accuracies and losses

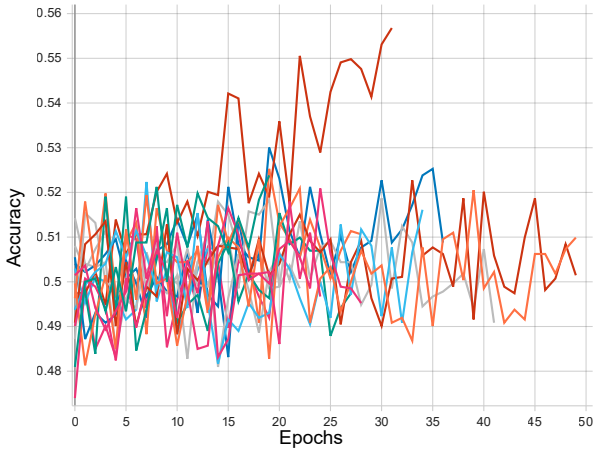

Figure 6.25: Figure of all training accuracies with one week of historic data in Iteration 5

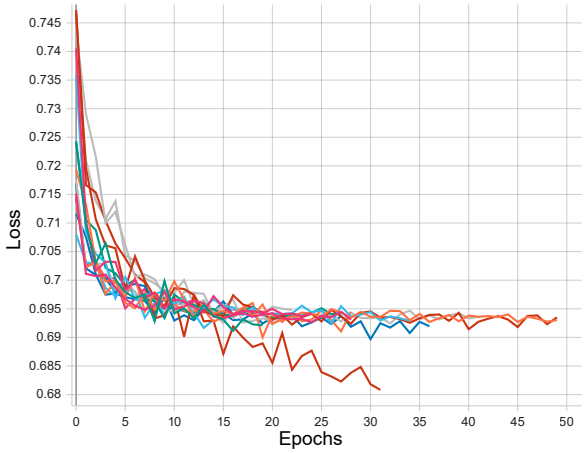

Figure 6.26: Figure of all training losses with one week of historic data in Iteration 5

Validation accuracies and losses

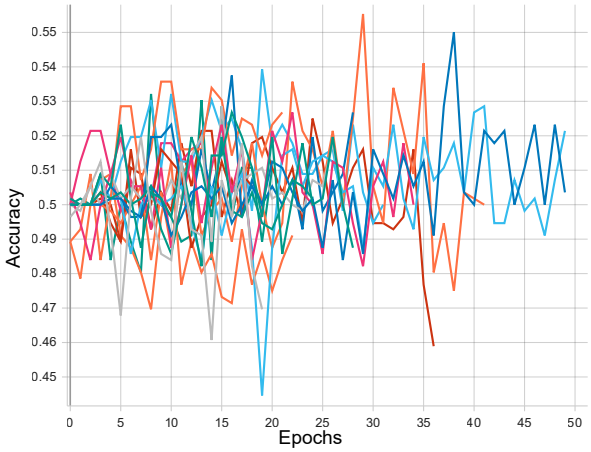

Figure 6.27: Figure of all validation accuracies with one week of historic data in Iteration 5

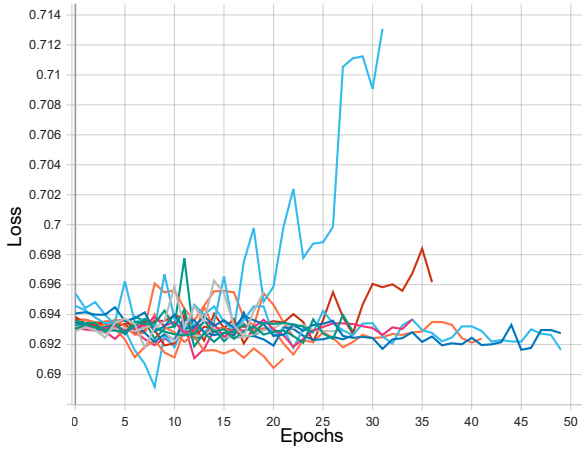

Figure 6.28: Figure of all validation losses with one week of historic data in Iteration 5

The greatest validation accuracy for the model with a sequence length of one week was 55.54% with the dataset using price and repurchase agreements (repo) data. While the chart above in Figure 6.28 may look erratic, the changes to the loss are minimal due to the scale as it only varies between 0.688 and 0.713. This suggests there is little overfitting as it does not increase

significantly.

Two weeks of historic data accuracies and losses

Training accuracies and losses

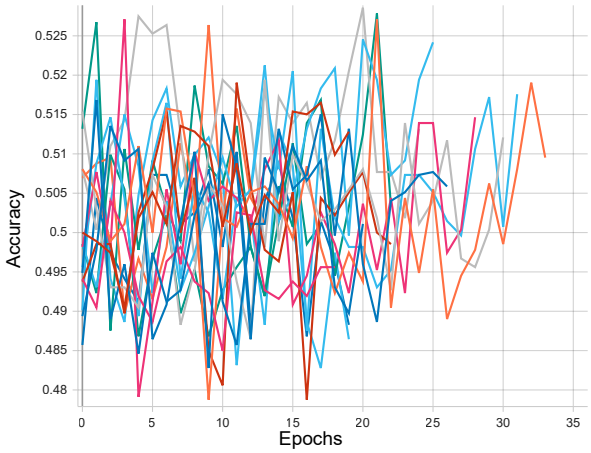

Figure 6.29: Figure of all training accuracies with two weeks of historic data in Iteration 5

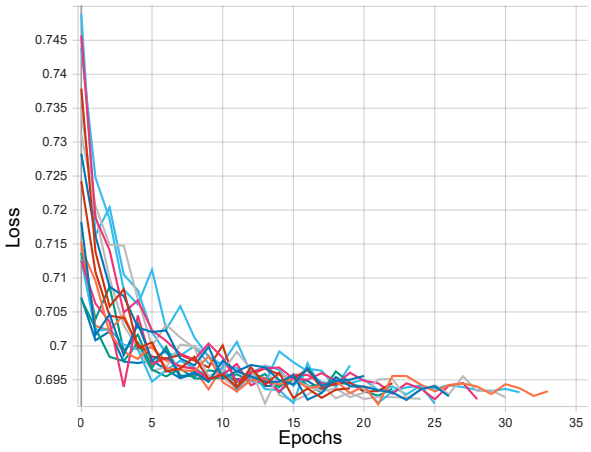

Figure 6.30: Figure of all training losses with two weeks of historic data in Iteration 5

Validation accuracies and losses

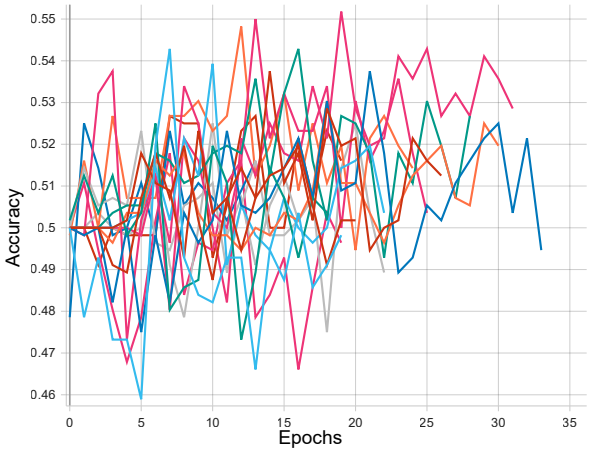

Figure 6.31: Figure of all validation accuracies with two weeks of historic data in Iteration 5

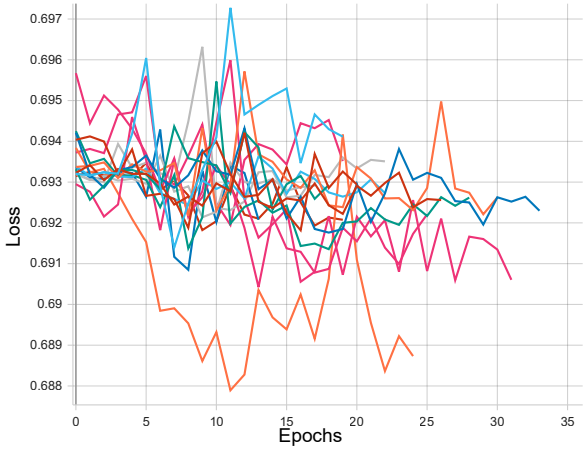

Figure 6.32: Figure of all validation losses with two weeks of historic data in Iteration 5

The greatest validation accuracy for the model with a sequence length of two weeks was 55.18% with the dataset using price and repurchase agreements (repo) data. While the chart above in Figure 6.32 may look erratic, the changes to the loss are minimal due to the scale as it only varies between 0.688 and 0.697. This suggests there is little overfitting as it does not increase

significantly.

One month of historic data accuracies and losses

Training accuracies and losses

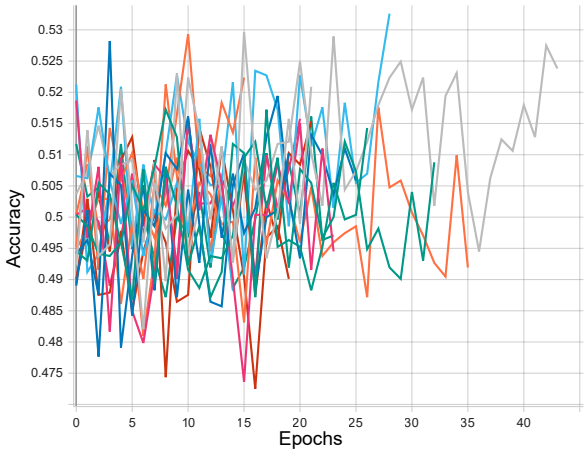

Figure 6.33: Figure of all training accuracies with one month of historic data in Iteration 5

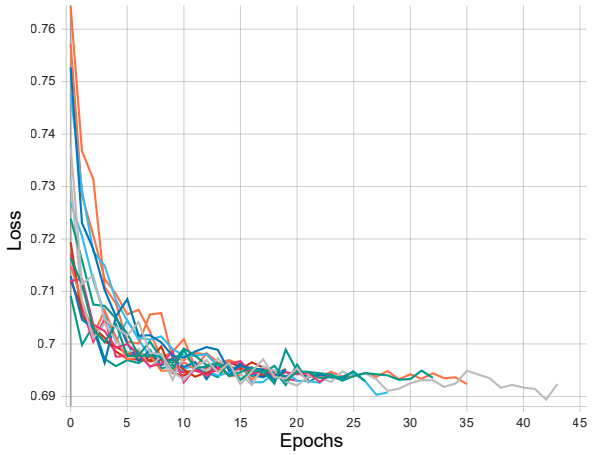

Figure 6.34: Figure of all training losses with one month of historic data in Iteration 5

Validation accuracies and losses

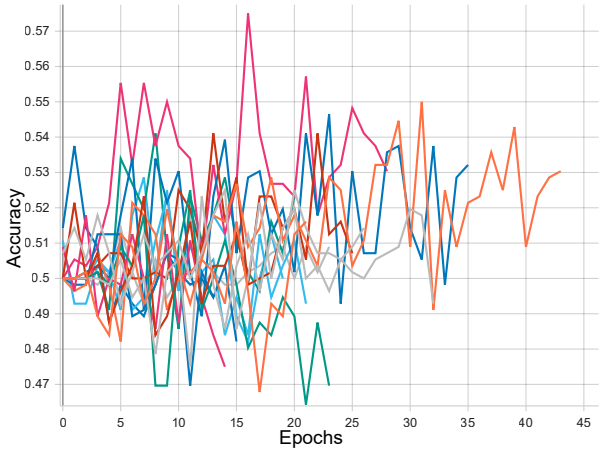

Figure 6.35: Figure of all validation accuracies with one month of historic data in Iteration 5

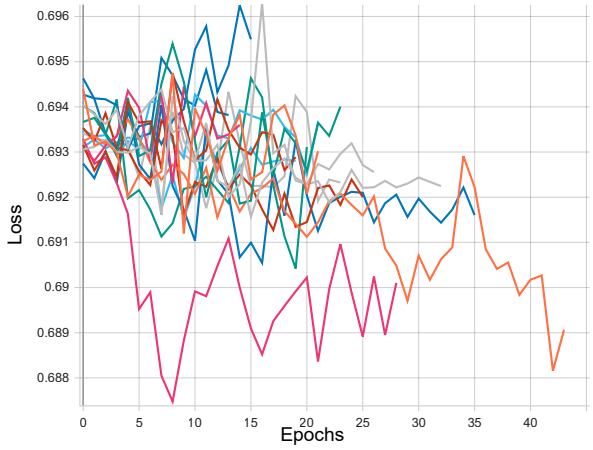

Figure 6.36: Figure of all validation losses with one month of historic data in Iteration 5

The greatest validation accuracy for the model with a sequence length of two weeks was 57.50% with the dataset using price and treasury yields data. While the chart above in Figure 6.36 may look erratic, the changes to the loss are minimal due to the scale as it only varies between 0.687 and 0.696. This suggests there is little overfitting as it does not increase significantly.

Two months of historic data accuracies and losses

Training accuracies and losses

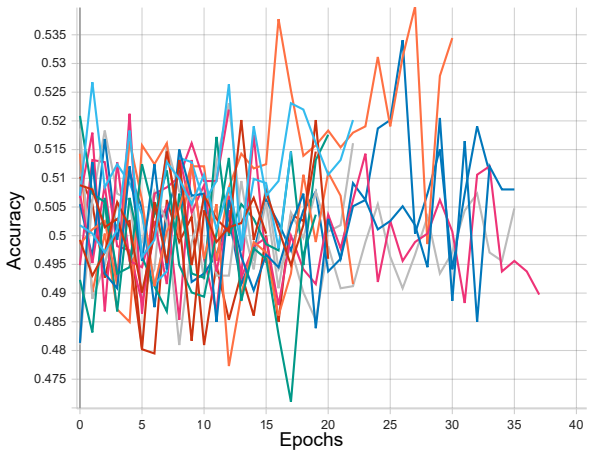

Figure 6.37: Figure of all training accuracies with two months of historic data in Iteration 5

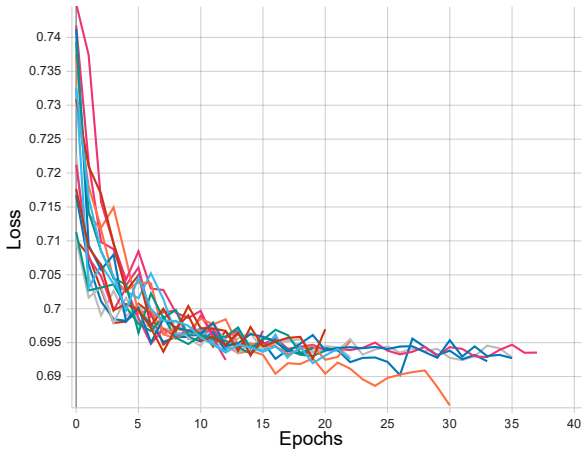

Figure 6.38: Figure of all training losses with two months of historic data in Iteration 5

Validation accuracies and losses

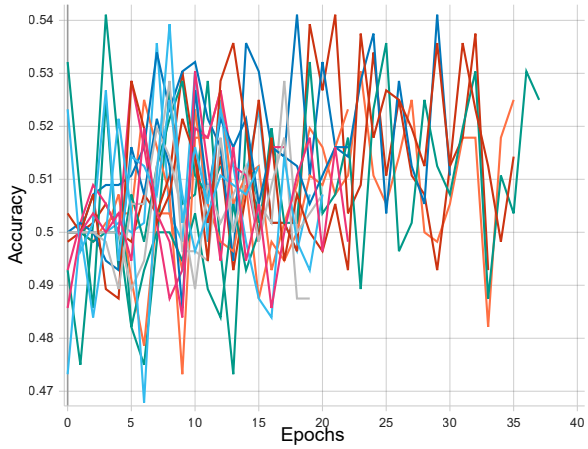

Figure 6.39: Figure of all validation accuracies with two months of historic data in Iteration 5

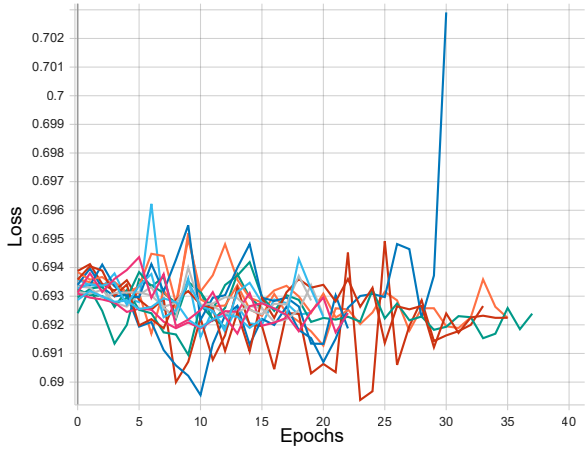

Figure 6.40: Figure of all validation losses with two months of historic data in Iteration 5

The greatest validation accuracy for the model with a sequence length of two weeks was 54.11% with the dataset using the extended price changes dataset, the price and volatility data, or the price and treasury yields data. While the chart above in Figure 6.40 may look erratic, the changes to the loss are minimal due to the scale as it only varies between 0.689 and 0.703. This

suggests there is little overfitting as it does not increase significantly.

One quarter of historic data accuracies and losses

Training accuracies and losses

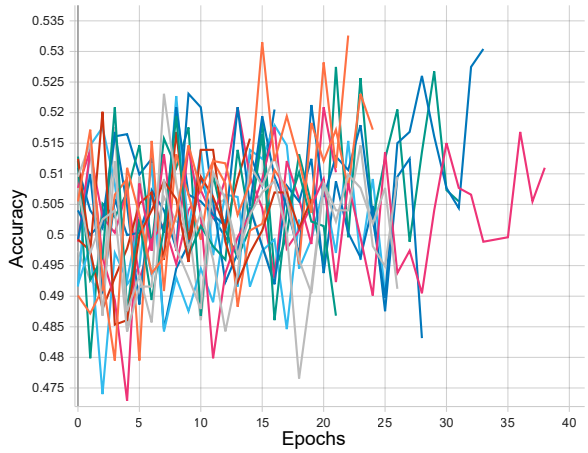

Figure 6.41: Figure of all training accuracies with one quarter of historic data in Iteration 5

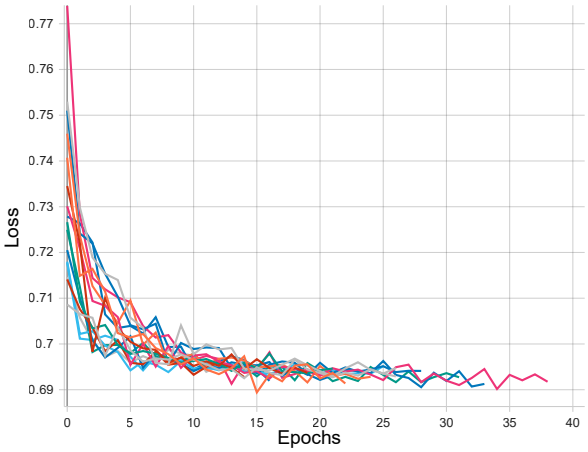

Figure 6.42: Figure of all training losses with one quarter of historic data in Iteration 5

Validation accuracies and losses

Figure 6.43: Figure of all validation accuracies with one quarter of historic data in Iteration 5

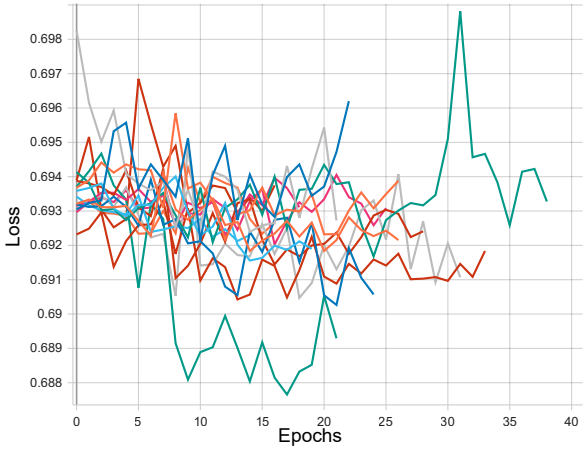

Figure 6.44: Figure of all validation losses with one quarter of historic data in Iteration 5

The greatest validation accuracy for the model with a sequence length of two weeks was 55.18% with the dataset of the price and treasury yields data. While the chart above in Figure 6.44 may look erratic, the changes to the loss are minimal due to the scale as it only varies between 0.687 and 0.699. This suggests there is little overfitting as it does not increase significantly.

6.6.2 Comparisons of models in iteration 5

The table below in Figure 6.45 shows the accuracies of each combination of input features and length of historic data.

	2 days	1 week	2 weeks	1 month	2 months	1 quarter	Average
price only	53.57%	53.21%	52.68%	51.79%	53.57%	53.75%	53.10%
price extended	51.07%	51.25%	51.79%	53.75%	54.11%	50.71%	52.11%
price and volume	54.29%	52.50%	52.50%	52.86%	53.04%	52.50%	52.95%
price and volatility	55.54%	52.14%	52.50%	54.11%	54.11%	54.11%	53.75%
price and money supply	54.11%	52.86%	53.93%	55.00%	52.86%	52.50%	53.54%
price and gdp	54.46%	55.00%	52.50%	52.32%	51.96%	52.86%	53.18%
price and treasury yields	53.04%	53.93%	54.82%	57.50%	54.11%	55.18%	54.76%
price and effr	54.82%	52.68%	53.75%	52.50%	52.68%	52.68%	53.19%
price and repo	52.32%	55.54%	55.18%	54.64%	53.57%	54.11%	54.23%
price and reverse repo	53.39%	51.61%	52.32%	52.32%	52.32%	52.32%	52.38%
price and gold	54.82%	52.68%	53.75%	52.50%	53.75%	51.61%	53.19%
price and currency	53.75%	51.96%	54.29%	51.96%	53.04%	53.21%	53.04%
price and options	54.46%	53.75%	54.29%	54.11%	52.86%	52.86%	53.72%
price and inflation	52.14%	52.86%	53.04%	51.79%	52.68%	52.68%	52.53%
price and employment	53.04%	51.96%	51.96%	52.50%	53.93%	51.61%	52.50%
price and other sentiment	56.61%	53.57%	55.00%	51.25%	53.21%	55.00%	54.11%
Average	53.84%	52.97%	53.39%	53.18%	53.24%	52.98%	

Figure 6.45: Figure of all validation accuracies of each combination of input features and sequence length in Iteration 5

The results show the model of the stock market index price combined with treasury yield data and a sequence length of 1 month (21 days) of historic data, has the best accuracy with a value of 57.50%.

However, there is additional data that can be inferred from the results. On average, the model with a sequence length of two days performs the best with a validation accuracy of 53.84%. This suggests that the stock markets may react strongest to short term changes. The models' accuracies slightly decrease for the one-week sequence length, and fluctuate slightly across other sequence lengths but never increase over the model with a sequence length of two days.

On average, the model utilising price and treasury yields data, have the highest average validation accuracy of 54.76%. This shows that treasury yields data can be an important factor to stock market returns.

In order to better visualise how different features affect the model, each combination of input features was compared against the 'price only' dataset for each sequence length. The results of these comparisions can be shown in the table below in Figure 6.46.

	2 days	1 week	2 weeks	1 month	2 months	1 quarter	Average
price only	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
price extended	-4.67%	-3.68%	-1.69%	3.78%	1.01%	-5.66%	-1.82%
price and volume	1.34%	-1.33%	-0.34%	2.07%	-0.99%	-2.33%	-0.26%
price and volatility	3.68%	-2.01%	-0.34%	4.48%	1.01%	0.67%	1.25%
price and money supply	1.01%	-0.66%	2.37%	6.20%	-1.33%	-2.33%	0.88%
price and gdp	1.66%	3.36%	-0.34%	1.02%	-3.01%	-1.66%	0.17%
price and treasury yields	-0.99%	1.35%	4.06%	11.03%	1.01%	2.66%	3.19%
price and effr	2.33%	-1.00%	2.03%	1.37%	-1.66%	-1.99%	0.18%
price and repo	-2.33%	4.38%	4.75%	5.50%	0.00%	0.67%	2.16%
price and reverse repo	-0.34%	-3.01%	-0.68%	1.02%	-2.33%	-2.66%	-1.33%
price and gold	2.33%	-1.00%	2.03%	1.37%	0.34%	-3.98%	0.18%
price and currency	0.34%	-2.35%	3.06%	0.33%	-0.99%	-1.00%	-0.10%
price and options	1.66%	1.01%	3.06%	4.48%	-1.33%	-1.66%	1.21%
price and inflation	-2.67%	-0.66%	0.68%	0.00%	-1.66%	-1.99%	-1.05%
price and employment	-0.99%	-2.35%	-1.37%	1.37%	0.67%	-3.98%	-1.11%
price and other sentiment	5.67%	0.68%	4.40%	-1.04%	-0.67%	2.33%	1.89%

Figure 6.46: Figure of all validation accuracies of each combination of input features and sequence length in Iteration 5

The model with a sequence length of two days does not see many significant improvements with most combinations of input features. For sequence lengths, many factors can have a positive impact to the models, the results above show that volume, volatility, M1 money supply, GDP, Effective Federal Funds Rate (EFFR), gold prices, currency exchange rates, put-to-call ratio (options), and other sentiment data. Whereas for sequence lengths of one month, there is some overlap of important factors that affect the accuracy of the models. The factors that impact one month sequence lengths are the

extended price dataset, as well as volume, volatility, M1 money supply, GDP, treasury yields, EFFR, Repurchase Agreements (Repo) utilisation and rates, Reverse Repo utilisation and rates, gold prices, currency exchange rates, and put-to-call ratio (options) data.

6.6.3 Evaluation of iteration 5

The results from iteration 5 provided valuable data in order to be able to answer the research questions. With the best model having an accuracy of 57.50%, the results suggest that stock market movements are not always truly random and there are factors that affect stock prices. This final iteration has provided great insight into which input features affect accuracy of stock market predictions the most, as well as what sequence lengths of historic data as input. This iteration provides a model with greater accuracies than any model prior to it. The models in this iteration also show very little overfitting as seen in only very minute fluctuations to the validation losses that do not increase significantly.

Due to time constraints and computational complexity, it was not possible to test various different models of different layers and layer sizes for each combination of input features which may impact the results.

6.7 Benchmark application results

The benchmark application uses an existing case study of input features and applies it to the existing research method of the CNN-LSTM sequential model as described in Iteration 3.

6.7.1 Accuracies of all tests

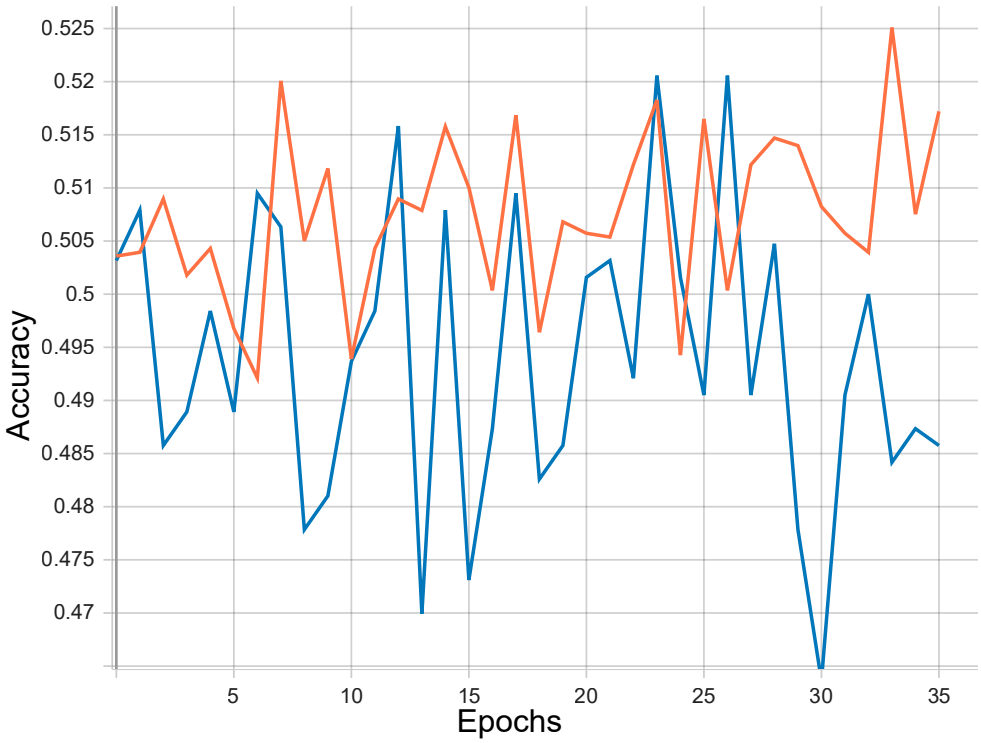

Figure 6.47: Figure of training and validation accuracies in benchmark model

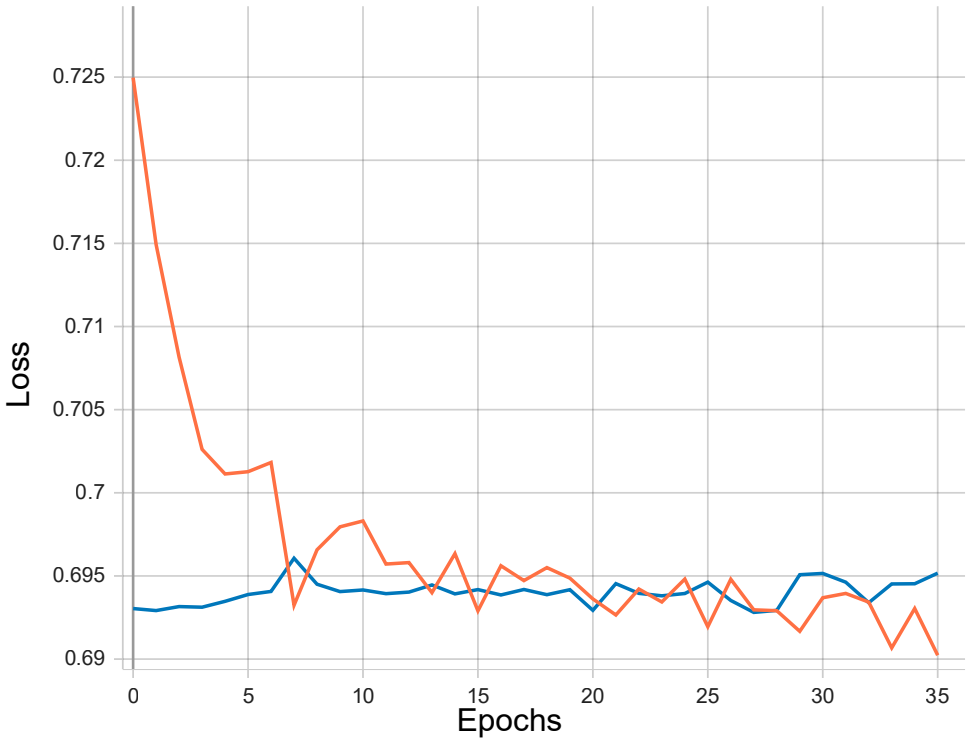

Figure 6.48: Figure of training and validation losses in benchmark model

The orange line represents the training set and the blue line represents the validation set. The validation loss shows little overfitting as can be seen in Figure 6.20 as the loss does not significantly increase. At 10 epochs, the validation accuracy reaches its highest value of 54.55% whilst the training accuracy reached 50.18%. The validation accuracy being somewhat greater than the training accuracy can be explained by the dropout layers within the model that can affect the results of the training set but does not affect the validation set.

6.7.2 Accuracies of the best result

6.7.3 Evaluation of benchmark model

With a validation accuracy of 54.55%, there is an improvement above a random guess of the direction of the next trading day. This model was included to compare the results of the models of the artefact to understand how well it performs to existing models. This result suggests that the models of all five iterations of the artefact outperform the existing case study as described in subsection 5.13.3.

6.8 Artefact evaluation

The artefact in this project is shown to be novel based on the unique sets of input features as it is the first time application of a novel case study to an existing research method. Generally, the results from the models shown have been able to support in answering the research questions.

6.8.1 Research questions evaluation

Predicting a stock market index's direction

Question Is it possible to accurately predict a stock market index's (S&P 500) direction for the following day with an artificial intelligence model?

Sub Questions

- If so, what is the accuracy of the model?
- If not, what limitations affected the model to cause it to be inaccurate?
- Can the findings of this question be used to prove/disprove the efficient market hypothesis?

The results from the artefact suggest that it is possible to predict the stock market index's direction with a reasonable degree of accuracy. The best model of Iteration 5 had a validation accuracy of 57.5% which suggests there is often an ability to predict the daily direction of a stock market index. This may also suggest there are often inefficiencies in the market and may offer some evidence to disprove the efficient market hypothesis. While this model has been quite accurate, there are some limitations involved that may affect the accuracy. The model uses data from November 2006 to March 2022; this involves a relatively large time horizon and as a result, some ideas learned by the model may be outdated as the markets evolve over time; for example settlement periods adjusting from T+3 days to T+2 days.

Optimal amount of historical data as input

Question How do different lengths of time as history for the input in the training data affect the model?

Sub Questions

- What is the ideal length of time to be used in the training data?
- What are potential reasons for this length of time being the most useful?

The results from the artefact generally show that one month of historical data can provide the best models as seen in Figure 6.46; though for the input features tested, two days has the best results when averaged as the results show in Figure 6.45. Generally, a short sequence length of historical

data provides the best result for predicting the daily direction of the stock market index with few input features affecting the accuracy of the model. However, given an optimal set of input features, the model may be able to recognise patterns in longer sequence lengths.

The two day sequence length may be more accurate due to the fact markets are always adjusting and reacting to changes; and as a result there may be short term inefficiencies that can be exploited. However, the one month sequence length accuracies may be explained due to the fact many economical factors are reported by institutions / governments on a monthly basis. Furthermore, the one month sequence length may be the best length for the LSTMs in the neural network model to identify and remember patterns that are beneficial to the model and disregard any patterns that provide less value.

Most important input features

Question What are the most important input features that affect the model?
Sub Questions

- How much do each of these input features contribute to the model?
- Do any of the input features identified have negligible impact to the model?
- What are the potential reasons these features do / do not impact the model?

The results are significantly more nuanced and cannot be placed into neat categories as described in section 5.2. On average, including treasury yields data to the model improved accuracies by 3.17% compared to price alone as seen in Figure 6.46. The artefact suggests that information regarding the treasury yields have the most impact to the neural network models across all sequence lengths apart from two days. This is likely due to the fact there generally aren't significant changes to treasury yields in a two day period;

but they can grow to be more significant over a long time horizon. The increased accuracy is likely due to the changes in yields may affect institutional decisions depending on the risk management. Firms may reallocate funds between the stock market and treasury products; treasury products are generally less risky but offer lesser returns than the stock market. Repurchase agreements (repos) show trends similar to the treasury yields. There generally aren't significant changes in 2 days, but there may be patterns over longer periods of times. This may be due to the fact changes to repo markets can affect liquidity in other markets.

There are several factors that do not impact the model greatly, such as reverse repurchase agreements (reverse repos). This is likely due to the fact reverse repos were generally underutilised, especially in the training period and as a result did not affect the model significantly. Exchange volume, gold, currency exchange rates, inflation rates, and employment rates generally did not affect the accuracy significantly. This is likely due to those factors remaining relatively stable most of the time.

6.8.2 Requirements evaluation

Functional requirements

Necessary functional requirements

Requirement	Met?
Predict daily price direction with 60%+ accuracy	No
Identify which input features are most important	Yes
Identify what sequence length (number of days of history of each input feature) is optimal	Yes
Present charts of accuracies and losses of each model	Yes

Table 6.2: Table showing whether necessary functional requirements were met

Unfortunately, the models were not able to provide an accuracy greater than

60%, but this did not affect the models' ability to identify optimal features in stock market forecasting. Charts were also able to produced using the Tensorboard callback.

Optional functional requirements

Requirement	Met?
Predict daily price return with a mean absolute deviation of 2.5% or lower	No

Table 6.3: Table showing whether optional functional requirements were met

This requirement was not met due to time constraints and decisions to focus on other models predicting direction only. This decision was made due to the belief that understanding the direction is more important than specific prices to generate a positive return on investment.

Non-functional requirements

Necessary non-functional requirements

Requirement	Met?
The artefact should take less than one minute to run per model chosen (on modern Nvidia GPUs)	Yes*

Table 6.4: Table showing whether necessary non-functional requirements were met

*All models apart from concatenated parallel hybrid models were able to train the models in under a minute. The models in Iteration 4 were not chosen to be used in the final iteration due to poor accuracies.

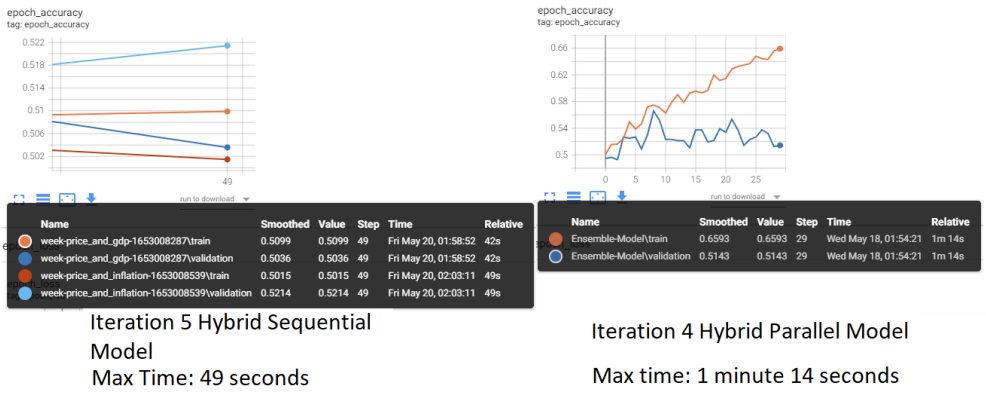

Figure 6.49: Figure comparing the time taken to train models in Iteration 5 compared to Iteration 4

Optional non-functional requirements

Requirement	Met?
Front-end application that allows users to input features and sequence lengths to train different models	No
User-facing documentation for artefact	No

Table 6.5: Table showing whether optional non-functional requirements were met

Due to time constraints these requirements were not completed. Due to this being primarily a research project rather than an engineering project, it was deemed unnecessary to have a front-end application. It is relatively easy to adjust different models with the python code provided. The code also contains comments throughout and uses relevant variable names to ensure readability and understanding. Furthermore, Tensorboard provides a good front-end for visualising performance metrics of the models.

Chapter 7

Project Management

Chapter 8

Conclusion

8.1 Overview

This project highlights the debate concerning the efficient market hypothesis and the feasability of predicting stock markets, particularly the US stock market. This debate is likely to continue for many years to come, but the project shows some evidence of market inefficiencies in short term horizons that neural networks can detect from both short-term and mid-term sequence lengths. This project looks at the performance of a few neural network models including Convolutional Neural Networks (CNNs), Long-Short Term Memory networks (LSTMs) and hybrid approachs (CNN-LSTMs).

The artefact generated in this project has a meaningful accuracy in stock market forecasting, and the results from final iteration have been able to aid in optimal feature selection in terms of input features and sequence lengths. The results of the models can be used to aid investors' decisions in what factors they focus on when making investment decisions as well as to allow investors to potentially take advantage of market inefficiencies.

8.2 Future Work

For the future, additional input features or different combinations of input features can be tested on the same research method to identify if there are opportunities for further optimisations. Furthermore, the same input features can be tested on different research models to verify the results of the optimal feature selection identified in this project. With an increased amount of time, or with more powerful hardware, additional neural network parameters such as the layer sizes and amounts can be tested to identify if there are better performing models.

While this project was not able to identify specific future prices of the stock market, that is an area that can be looked into further utilising the factors identified in this project. This could be helpful specifically to the options market where specific prices are important to participants.

Additionally, as this project identifies some input features are more or less important depending on the sequence length of historical data; an approach to exploit the advantages of different sequence lengths could prove to produce a model with greater accuracy.

References

Bachelor, L. (1900). Speculation theory. *3rd series, 17*, 21–86. https://doi.org/10.24033/asens.476

Eapen, J., Bein, D., & Verma, A. (2019). Novel deep learning model with cnn and bi-directional lstm for improved stock market index prediction. *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 0264–0270. https://doi.org/10.1109/CCWC.2019.8666592

Fitriyaningsih, I., Tampubolon, A. R., Lumbanraja, H. L., Pasaribu, G. E., & Sitorus, P. S. (2019). Implementation of artificial neural network to predict s&p 500 stock closing price. *Journal of Physics: Conference Series*, *1175*(1), 012107.

Goval, A., & Welch, I. (2004). A comprehensive look at the empirical performance of equity premium prediction. *NBER Working Paper Series*, *10483*. https://doi.org/10.3386/w10483

Greenspan, A. (1996). The challenge of central banking in a democratic society [Remarks by Chairman Alan Greenspan at the Annual Dinner and Francis Boyer Lecture of The American Enterprise Institute for Public Policy Research, Washington, D.C. [Accessed: 2021 10 29]]. http://www.federalreserve.gov/boarddocs/speeches/1996/19961205.htm

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, *38*(8), 10389–10397. https://doi.org/https://doi.org/10.1016/j.eswa.2011.02.068

Hao, Y., & Gao, Q. (2020). Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning. *Applied Sciences*, *10*(11), 3961. https://doi.org/10.3390/app10113961

Hu, H., Tang, L., Zhang, S., & Wang, H. (2018). Predicting the direction of stock markets using optimized neural networks with google trends. *Neurocomputing*, *285*, 188–195. https://doi.org/10.1016/j.neucom.2018.01.038

Kinsley, H. (2018). Python programming tutorials. https://pythonprogramming.net/balancing-rnn-data-deep-learning-python-tensorflow-keras/

Kumar, M., & Thenmozhi, M. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest. *Indian institute of capital markets 9th capital markets conference paper*.

Lee, C.-C., Lee, J.-D., & Lee, C.-C. (2010). Stock prices and the efficient market hypothesis: Evidence from a panel stationary test with structural breaks. *Japan and the World Economy*, *22*(1), 49–58. https://doi.org/https://doi.org/10.1016/j.japwor.2009.04.002

Maier, A., Syben, C., Lasser, T., & Riess, C. (2019). A gentle introduction to deep learning in medical image processing. *Zeitschrift für Medizinische Physik*, *29*(2), 86–101. https://doi.org/10.1016/j.zemedi.2018.12.003

Malkiel, B. G. (2005). Reflections on the efficient market hypothesis: 30 years later. *Financial Review*, *40*(1), 1–9. https://doi.org/https://doi.org/10.1111/j.0732-8516.2005.00090.x

Malkiel, B. G. (1973). *A random walk down wall street : The time-tested strategy for successful investing*. W.W. Norton; Company.

Narayan, P. K. (2006). The behaviour of us stock prices: Evidence from a threshold autoregressive model. *Mathematics and computers in simulation*, *71*(2), 103–108.

NYSE. (2020). *Equity index and fx products*. https://www.nyse.com/publicdocs/Trading_Days.pdf

Sewell, M. V. (2012). The efficient market hypothesis: Empirical evidence. *International Journal of Statistics and Probability*, *1*(2). https://doi.org/10.5539/ijsp.v1n2p164

Shen, S., Jiang, H., & Zhang, T. (2012). Stock market forecasting using machine learning algorithms. Retrieved December 8, 2021, from http://cs229.stanford.edu/proj2012/ShenJiangZhang-StockMarketForecastingusingMachineLearn pdf

Sheta, A., Ahmed, S., & Faris, H. (2015). A comparison between regression, artificial neural networks and support vector machines for predicting stock market index. *International Journal of Advanced Research in Artificial Intelligence*, *4*, 55–63. https://doi.org/10.14569/IJARAI.2015.040710

Thakkar, A., & Chaudhari, K. (2020). Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks. *Applied Soft Computing*, *96*, 106684. https://doi.org/https://doi.org/10.1016/j.asoc.2020.106684

Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, *5*. https://doi.org/10.1186/s40854-019-0138-0

Appendix A

First Appendix