



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

A deep study of LSTM networks and its application in financial markets

By

Vikram Kalta

Supervisor: Gordon Dickers

Second Market: Terry Walcotts

Project submitted as part of the requirements for the award of
MSc Data and Artificial Intelligence

September 2023

DECLARATION

I, Vikram Kalta declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed: Vikram Kalta

Date: 14th Sept, 2023

FORM OF CONSENT

I Vikram Kalta hereby consent that my Project, submitted in candidature for the M.Sc. Data and Artificial Intelligence, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations

Signed: Vikram Kalta

Date: 14th Sept, 2023

1 ABSTRACT

This dissertation explores the intersection of algorithmic trading and financial technology, with a specific emphasis on the NASDAQ100 index. It employs Long Short-Term Memory (LSTM) neural networks as predictive tools, examining the NASDAQ100, LSTM network development and training, and reflecting on the findings and their implications.

The research initiates by dissecting the financial literacy, importance of financial investments and then dives into the realm of financial markets. The stock markets are considered a reliable and an important area to get profits on investments if proper analysis is done in the background. This analysis serves as the groundwork for subsequent algorithmic trading strategies.

The primary contribution of this dissertation lies in developing and training LSTM networks tailored for algorithmic trading. The application of deep learning techniques to historical NASDAQ100 data demonstrates the potential of LSTM networks in capturing complex temporal dependencies within financial time series. The study emphasizes that when it comes to stock market predictability the accuracy of a model is not the right performance metric. Rather cumulative returns on the investment is considered to be the right metric for performance.

In this research the author performed various experiments on NASDAQ100 constituent stocks. The experiments had a mixture of features. One of the setup contained a single feature setup i.e the intraday return for each stock and another one contained 2-feature setup i.e the intraday return for each stock along with intraday return for NASDAQ100. The best overall returns attained was 0.03% on total investments. Which implies that if a investor invests \$10000 in a year, the trader can get an additional \$300 returns. Not the best but due to the hardware constraints the author believes that the similar setup could be utilized to achieve best returns on the investment. But the findings from the whole study opens an avenue for further improvement in this study.

Looking ahead, this research suggests avenues for further exploration, including enhancing model robustness through additional features and deploying algo-trading bots in real market conditions.

In addition to its contributions to algorithmic trading, this dissertation signifies personal and academic growth. It combines quantitative analysis, machine learning expertise, and financial understanding, imparting valuable lessons in perseverance and adaptability. The hope is that the insights gained will advance algorithmic trading strategies and inspire further research at the intersection of finance and technology.

Contents

1	ABSTRACT	3
2	INTRODUCTION	6
2.1	Financial literacy	6
2.2	Financial investment	6
2.3	History of trading	7
2.4	Role of emotions in trading	7
2.5	Supply and demand	8
2.6	Price charts and trends	8
2.7	Support and Resistance	11
2.8	Types of analysis	12
2.9	Types of trading	13
2.10	Online trading platforms	14
2.11	Risk management	14
2.12	Long and Short	14
2.13	Establishing trading principles	15
2.14	Automation of manual efforts of traders	17
2.15	What is machine learning	18
2.16	Machine learning workflow	18
2.17	Introduction to artificial neural network	19
2.17.1	Steps involved in training neural network	19
2.17.2	Loss calculation and weight updates	19
2.18	Report Structure	20
3	LITERATURE REVIEW	20
3.1	Determining key performance metrics in algo trading and simplis- tic models exploration	20
3.1.1	Results with single training and retraining	22
3.2	Employing CEFLANN	22
3.3	LSTMs in algo trading	23
4	RESEARCH METHODOLOGY	25
4.1	Research approach	25
4.2	Data collection	26
4.2.1	Dataset description	26
4.3	Algorithm design	26
4.3.1	Features selection	27
4.3.2	Target selection	28
4.4	LSTM networks	28
4.4.1	Step by step LSTM walk through	28
4.5	Software and hardware specifications	30
4.5.1	Hardware specifications	30
4.5.2	Software environment	30
4.5.3	Libraries and frameworks	31

4.5.4	Version Control	31
4.6	Timelines	31
4.6.1	Project Inception (Months 1-2)	31
4.6.2	Data Collection and Preprocessing (Month 0.5)	31
4.6.3	Model Development (Month 1)	32
4.6.4	Training and Evaluation (Month 0.5)	32
4.6.5	Conclusion and Dissertation Writing (Month 1)	32
4.7	Testing and Real-time Simulation	32
5	DESIGN & IMPLEMENTATIONS	32
5.1	Training the model	33
5.2	Testing the model	34
6	RESULTS & DISCUSSIONS	35
6.1	Experiment 1 with k=10 used for only long positions	35
6.2	Experiment 2 with k=10 used for long and short positions	37
6.3	Experiment 3 with k=5 used for long positions	38
6.4	Validation/training loss charts for best performing experiment	40
7	CONCLUSIONS & RECOMMENDATIONS	41
7.1	Algorithmic development	41
7.2	Machine learning model	42
7.3	Performance evaluation	42
7.4	Future works	42
8	REFLECTIONS	43
8.1	Research Process and NASDAQ100 Analysis	43
8.2	Development and Training of LSTM Networks	44
8.3	Personal and Academic Growth	44

2 INTRODUCTION

2.1 Financial literacy

Financial literacy in the today's environment is considered an important skill. It is the understanding of various financial concepts and skills that make informed and effective decisions about personal finances. There is a rapid growth in financial market with a wide range of financial products that people have to choose from such as student loans, credit cards, mortgages, etc. Student loan is one of the most used financial product. The study [1] suggests that in England 94% of the undergraduates preferred taking student loans that were backed by the government. The decision which an individual has to make to choose between these financial products has an indirect impact on individual well being. The kind of financial decisions that people make is directly related to an individual's financial literacy. Author in [2] advocates that there are certain fundamentals of financial literacy that affects the financial decision making the most. Three main concepts are as follows:

- Numeracy: It deals with calculating interest rates and understanding of how compounding in finance works.
- Understanding of rate of inflation.
- Understanding of the risk diversification.

An understanding of these concepts lays a foundation in managing personal finances for an individual. Personal finances mainly includes budgeting, saving and investing. The focus of this study is primarily on investing and understanding the importance of it. Further in this study the focus will be on to explore ways to invest in the market using various strategies. But before that is done, it is important to understand the concept of financial investment.

2.2 Financial investment

Financial investment refers to the allocation of money or resources with the expectation of generating future financial returns. It involves committing funds to various assets or financial instruments, such as stocks, bonds, real estate, mutual funds etc. in the hopes of growing wealth over time through capital appreciation, interest, dividends or other forms of income. Financial literacy provides the knowledge and skills required to make informed investment decisions. The understanding of concepts like risk, return, diversification and compounding enables individuals to assess different investment options and choose those that to align with their financial goals and risk tolerance. Evaluating potential risks and rewards of various investments minimizes the likelihood of significant losses. Allocating investments across different asset classes such as stocks, bonds, etc. and having a diversified portfolio helps manage risk and optimize returns. This involves having knowledge of how various sectors that are currently impacting the economy globally. This paper will try to shed some lights on investing in

stock markets which can be done in form of trading. The rewards it can fetch along with the risks associated with it and how various factors affect the decisions. Later the author will introduce concepts which focuses on eliminating the risk factors and maximizing the benefits. This study can help individuals who seek to understand the stock markets and investments. Whether they are at beginner, intermediate or advanced level in their trading journey, this study will highlight topics which can be of good use to any of them.

2.3 History of trading

The first stock exchange was started in New York City. The first Congress meeting was held in Federal Hall in the year 1789. This meeting was held to ascertain the financial responsibilities of the new colonies and the government. The government made available about \$80 millions in government notes, which created a new opportunity in securities. These securities in tandem with stocks, commodities etc. were then put up for sale for general public. Traders would then gather on a daily basis to sell and buy these assets. The investors would help fund American companies by buying shares of ownership. The investor's equity could be proved by the certificate of the stock issued by the company, in exchange for capital invested by the investor. This is why shares of stock alternatively is called a stock. The formal stock trading began in 1792. New York exchange was established in 1827 in Wall and Hanover street. Although New York City converted Manhattan's farmland to well built area the mentions of the animals were sustained on the Wall street. 2 animals which mostly described the traders were bulls and bears. The reason behind it was that the bulls fight with striking their horns upwards and bears attack with their claws swinging them downwards. This is resembled in trader's community by bull liking to believe that the market would go up and makes profits through that and the bears makes profits from the declining market. The bulls are hence also called the buyers and the bears are described as sellers.

2.4 Role of emotions in trading

Greed and fear play a big role in trading. Greed and fear motivate nearly all the traders. For example, imagine a stock of XYZ company is bought at \$100, after a few hours the price goes to \$200. The trader thinks the price will go higher and decides to put all his/her money in the trading account in this share. This is called greed. But the price now falls \$50. In another instance, imagine a trader buys stocks of XYZ company worth \$1000 and the next day the price falls and stock is now worth \$800, the trader panics, runs to the door and sells the same stocks. This is called fear. Next day the stock prices went higher at \$1200. But if the trader would have held a day longer, he/she would have ended in a win. Another dominating factor that is important to understand in this space is the concept of supply and demand.

2.5 Supply and demand

Demand and supply are fundamental concepts in trading that play a crucial role in determining the price of a financial asset, such as stocks, bonds, commodities, or currencies. These concepts are often used to analyze and predict price movements in financial markets. Demand refers to the desire of traders and investors to buy a particular asset at a specific price and quantity. If a stock is in demand the prices will naturally go higher. This happens because the stock holders refuse to sell the stock at a given time when the demand is high because they believe that price will rise in the future and this compels buyers to pay a higher amount for it. This normally happens when there is some new launch of a very successful product by the company or a great quarterly earnings data released. Supply represents the willingness of traders and investors to sell a specific asset at a particular price and quantity. In some scenarios, if a company is going through bad quarter resulting investors or traders selling their stocks. This causes the stock price to fall as now in this case no one is ready to buy.

2.6 Price charts and trends

The stock analysts uses price charts to determine market sentiment and direction of prices of the stock at any given time. Stock price charts are read on a computer screen which is built using advanced software. The chart will display high, low, opening and closing price of the stock on a selected time-frame which could be hourly, daily, weekly, monthly etc. The market trend is a series of peaks and troughs. An ascending peaks and troughs is seen as an uptrend and a declining seen as a downtrend [3].

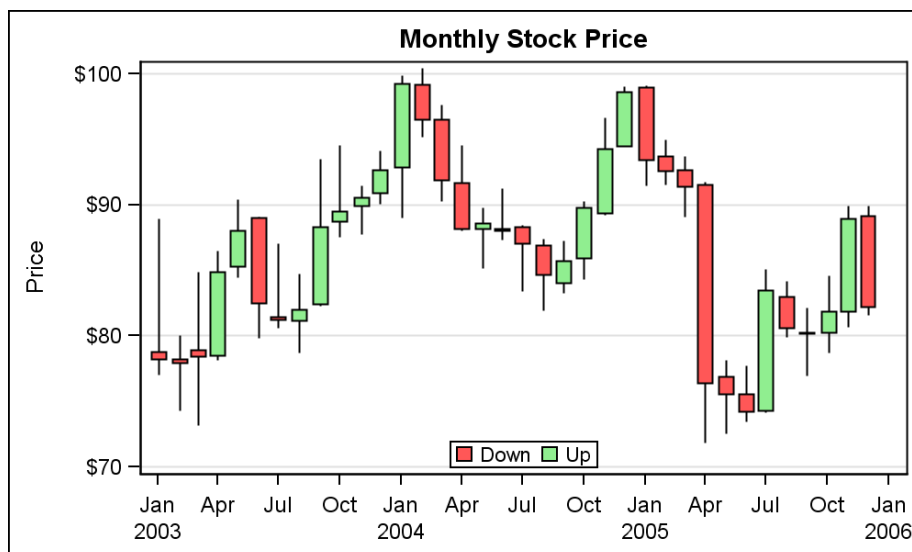


Figure 1: A candlestick chart displaying monthly chart

Figure 1 displays an example of such a price chart. The chart is a candlestick chart which is quite popular amongst the technical analysts to predict the trend and interpret the market sentiment. The candlestick body displays the open and close prices on the given time period and the wicks on top and bottom of the body represents high or low prices. Candlestick charting traces its roots back to Japan in the 18th century, where it was used to analyze the price movements of rice in the futures market. A Japanese rice trader named Munehisa Homma is often credited with developing this early form of candlestick analysis.

There are 3 main types of trends in the stock market:

- Uptrend: This indicates that the overall direction of the market is upward, with buyers/bulls dominating and pushing prices higher. Uptrends are typically associated with the positive market sentiment and economic growth.

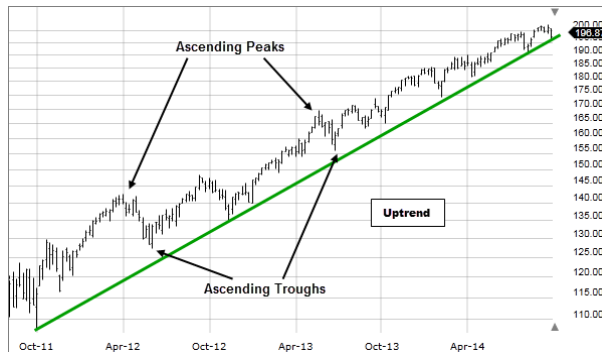


Figure 2: Uptrend chart with ascending peaks and ascending troughs

- Downtrend: This indicates that overall direction of the market is downward, with sellers/bears in control and driving prices lower. Downtrends are often linked to negative market sentiment, economic contraction.

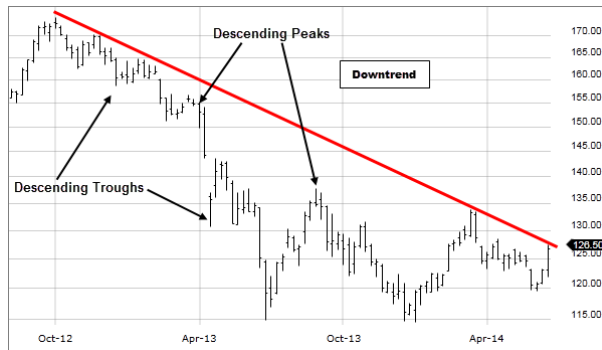


Figure 3: Downtrend chart with descending peaks and descending troughs

- Sideways trend: A sideways trend, also known as a range bound or horizontal trend, occurs when prices move within a relatively narrow range without a clear upward or downward direction.

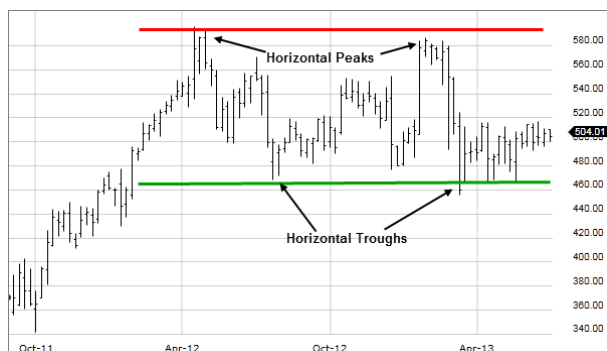


Figure 4: Downtrend chart with horizontal peaks and horizontal troughs

2.7 Support and Resistance

In the realm of financial markets and trading, a fundamental concept that holds significant importance is the comprehension of support and resistance levels. These critical price levels play a pivotal role in technical analysis by influencing trading decisions and market dynamics.

2.7.1 Support Levels

Support levels represent price points in a financial market where downward price movements often encounter obstacles. In simpler terms, these levels are akin to a safety net that prevents the asset's price from declining further. When an asset's price approaches or reaches a support level, it tends to find buying interest, as traders and investors perceive the price as attractive or undervalued. This increased buying pressure can halt or reverse the downtrend.

Support levels are not fixed; they are dynamic and can shift over time based on market sentiment, economic events, and other factors. Traders often identify support levels through historical price analysis, focusing on areas where prices previously found stability or experienced a rebound.

2.7.2 Resistance Levels

On the flip side, resistance levels denote price levels where upward price movements face barriers. These levels act as ceilings that can deter further price appreciation. When an asset's price nears or reaches a resistance level, traders and investors may perceive it as overvalued or facing selling pressure. Consequently, this can lead to increased selling activity, potentially halting or reversing an uptrend.

Like support levels, resistance levels are not static and adapt to changing market conditions. Analysts identify resistance levels by examining historical price

patterns, pinpointing areas where prices previously encountered resistance or stalled.

2.7.3 Significance in Trading

Understanding support and resistance levels is crucial for traders and investors as they provide valuable insights into potential entry and exit points. Traders often use these levels to set stop-loss orders to limit potential losses or take-profit orders to lock in gains.

Moreover, support and resistance levels can serve as indicators of market sentiment and future price movements. Breakouts, where prices move decisively above resistance or below support, can signal significant shifts in supply and demand dynamics, potentially leading to sustained trends.

In conclusion, comprehending support and resistance levels is a cornerstone of technical analysis in trading. These levels reflect the ebb and flow of market sentiment and play a vital role in guiding trading strategies and decision-making. By recognizing these key price levels and their significance, traders can enhance their ability to navigate the complexities of financial markets effectively.

2.8 Types of analysis

The stock price evaluation is majorly done using 2 methods known as fundamental analysis and technical analysis. Fundamental analysis is a method used to evaluate the intrinsic value of a financial asset typically stocks, bonds or commodities. This approach involves analyzing the underlying factors that can influence the value of an asset, such as economic indicators, financial statements, industry trends, management quality, and competitive landscape. The goal of fundamental analysis is to determine whether an asset is overvalued or undervalued in relation to its true worth. Investors who use fundamental analysis believe that by understanding the fundamental factors driving an asset's value, they can make informed investment decisions. Key aspects of fundamental analysis include:

- Financial statements: Analyzing a company's income statement, balance sheet and cash flow statement to assess its financial status and performance.
- Economic indicators: Considering macroeconomic factors such as GDP growth, unemployment rates, inflation and interest rates, which can impact the overall market and individual assets.
- Industry analysis: Evaluating the competitive landscape and trends within a specific industry to gauge the potential growth and profitability of a company.

Technical analysis is a method used to forecast the future price movements of financial assets, such as stocks, forex, commodities, bonds etc based on historical

price and volume data. This approach focuses on studying chart patterns and trends. Technical analysts believe that historical price movements tend to repeat and by identifying patterns and trends, they can predict future price movements.

Since technical analysis is the main focus of this study, it is important to understand how it can be further optimized to make better trading decisions using cutting edge technologies. The author here intends to discuss each of the key aspect in detail which dominates technical analysis. The key aspects of technical analysis includes the following:

- Price charts: Technical analysts uses various types of price charts, such as line charts, bar charts and candlestick charts to visualize past price movements and identify patterns. Out of all the charts candlestick charts are used most by the analysts. As seen in the figure 1, candlestick chart lets a user view open, close, high and low prices in a time period in one picture. Each candlestick is a body consisting of a candle and wicks at the top and bottom. The candle ends represents open and the close price and the wicks represents high and low of the specific time frame the user is looking at. The time period can be hourly, daily or weekly or even monthly and so on, depending upon what kind of trades the user is interested in [4]. If the opening price is higher than the closer price then user would see candlestick in red or black color otherwise in green or white color. There are a number of candlestick patterns which a technical analyst uses as a tool to predict the next movement of the market.
- Indicators: These are mathematical calculations applied to price to derive insights about potential trends and reversals. Examples of indicators include moving averages (simple, exponential, etc.), Relative Strength Index (RSI), Moving Average Convergence Divergence(MACD), etc. These indicators help identify trends, overbought or oversold conditions, and potential entry/exit points.
- Support and resistance: Support and resistance can be used as a technical indicators by some of the technical analysts. The trading signals are generated sometimes on the basis of this concepts. Support is basically an area or a level where buyers interest overtakes the sellers interest [5] and hence causing a reversal of the trend. Resistance is the opposite of the support where selling pressure increases.

2.9 Types of trading

In theory, there are 3 types of traders. They are long-term investors, corporations and high frequency traders [6]. This study emphasises on short-term trading, so it would be nice to look into a few of types of styles of trading within the high frequency trading.

- Scalpers: This type of traders like to trade for short duration, quickly collect profits and close positions.

- **Swingers:** Swing trading refers to opening a position at the start of a trend (uptrend/downtrend) and then closing at the end of the trend. This trend continuation could last for weeks or months.
- **Technical analysis:** In this type of trading a trader utilizes various tools such as technical indicators, momentum oscillators, etc to determine best moment to enter a trade and get the maximum profit with minimum risk.

2.10 Online trading platforms

In order to place trades in the real market a trader needs to open a trading account. A trading account can be opened in various trading platforms available online. Some of them are trade212, plus500, E-Trade, etc. Most of the trading platform provides 2 types of account. Paper trading and real account. The paper trading simulates the real world trading environment by placing trades with fake money. And is often used by novice traders to get insights of stock market and its price movements. A novice trader can evaluate the stock price movement with the knowledge of technical indicators and news associated with the company and open a position based on that. This helps a novice trader estimate losses/profits but in a simulated environment. The real trading comes with real money associated with each trades and hence comes with risk of losing money. So its important for a new trader to do a due diligence prior to placing trades.

2.11 Risk management

Risk management in trading is one of the most important aspect. Before a trader enters a trade a trader should estimate a stop loss condition. A stop loss refers to setting a limit to a trade if it starts going in the opposite direction from the estimated one [7]. This will avoid excessive loss conditions. Ideally risk to reward ratio is set to 1:2. For example if reward is \$6 above the entry point stop loss should be at \$3 below the entry point. The amount of money a trader can afford to lose is sometimes considered as a factor to set a stop-loss condition. Later in this paper the author will dive deep into some mathematics based methods which could be used to automate the tasks of placing these trades.

2.12 Long and Short

Another important concept in trading to highlight is the concept of long and short. A trader can both long or short a trade when opening a position. When shorting or selling a stock, a trader sells the stock at a higher price and estimates that the price of the stock will go down and then buy it again at a lower price [8]. This difference in price at the time of buying back is the profit made by the trader. If the price at the time of closing the position is higher than the entry price then the trader would have to bear losses. One of the illustration of such a scenario would be a trader shorting 100 shares of a stock. The broker of the platform being used would then borrow these stocks from another institution

and would sell them in the market. Later when the trader decides to close the position when the stock price drops, the broker would replace it back in the account of the institution from which the stocks were borrowed. If the stock was sold at \$10 each and at the time position was closed the price was \$7 each, the profit made would be the difference multiplied by the number of shares bought. Which in this case is \$3000. If the price rises to \$13 then the trader will have to bear a loss of \$3000. Longing works the opposite way by trader buying a stock at a lower price and selling it at a higher price.

2.13 Establishing trading principles

Till now all the important aspects for opening and closing a position using a trading account has been discussed. Now its time to use this knowledge and use it to pick a stock and try to determine a entry point in the market with help of some illustration. Bear in mind, that before entering a trade, its also important to set a stop-loss and take profit condition. Take profit is a condition in modern trading platforms where a trader could set a limit to stock price in order to close the position after taking some profits.



Figure 5: Tesla stock in uptrend in 5 minutes time period chart

To take a long position, it is generally considered important to make sure the current trend is uptrend within the time-frame of interest. The figure 5 shows Tesla stock (TSLA) in 5 minutes chart. The trend currently is uptrend, so a trader would want to wait for a perfect entry point. To get the perfect entry point the trader generally waits for a pullback. A pullback occurs when a stock price starts declining below a certain resistance zone [9]. This pullback again meets a support zone and this is when a trader would want to enter a trade by opening a long position. If the trader is scalping then the trader would want to close the position when again a pullback condition is met. The pullback illustration is shown in figure 6.



Figure 6: The arrow shows the pullback in an uptrend.

When shorting or selling, trader enters a trade at the time pullback starts and exits when it meets support zone. To confirm this entry and exit points, often traders use charts to draw trendlines and other technical indicators such as simple moving average (SMA), moving average convergence divergence (MACD) lines, etc. A Simple moving average is a commonly used technical analysis tool in trading. It helps traders smooth out price data to identify trends and potential turning points in a stock or other assets. The SMA is calculated by taking the average of a set number of past closing prices over a specified period of time. The MACD uses exponential moving average (EMA) [10]. It composes of 2 EMA lines, 1st line which is also called the MACD line is the difference between 26 and 12 period EMA lines, the 2nd line also called the signal line is 9 period EMA of the MACD line itself. Generally when MACD line crosses over signal line its considered a bullish signal. MACD line illustration is shown in figure 7. The blue line also known as fast line is the MACD line and the red line also known as slow line is the signal line.

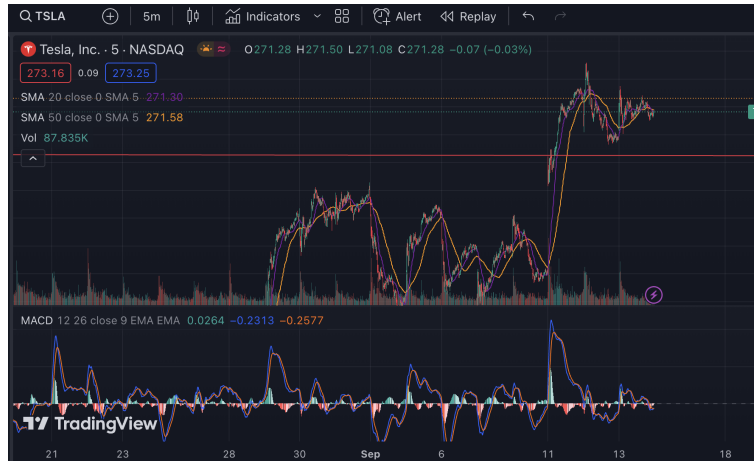


Figure 7: MACD lines. Red line is the slow/signal line and the blue line is the fast/MACD line

Generally it is safe to enter a trade when all the indicators align and share a bullish signal for taking a long position. These are some trading principles that a trader needs to follow to ensure a trading win. In order to short a stock trader generally looks for a bearish signal using same indicators. To enter a trade, a trader has to manually read the charts for each stock, check if all the technical indicators align, check for the overall trend of the market to determine to short or long a stock. As an additional step it is also important for a trader to calculate the stop loss and take profit conditions before a position is opened. There is another element of news based on stocks which has a significant impact of the stock's price movement because it can influence investor's sentiment. One of the most reliable and oldest sources of news for stock markets is cnbc.com. All this manual efforts and knowledge of financial markets are crucial and also at times requires a lot of efforts from the trader.

2.14 Automation of manual efforts of traders

What if the task of reading charts and confirming patterns to enter a trade could be automated. This way a trader could focus on things which could make a trader more productive and can eliminate the role of emotions which comes into play at the same time of placing trades. A trading bot which could mine into the historical datasets and could derive a pattern to predict the next movement of the stock market would be a great use to traders. This study proposes automation of such tasks associated with opening or closing a position in the stock market and investigates certain machine learning based methods to predict the market movement on the basis on the historical data.

2.15 What is machine learning

Machine learning is a branch of artificial intelligence that focuses on the development of algorithms and statistical models that enables computer systems to improve their performance. A machine learning model takes as input some dataset and then tries to find a pattern in it. It is done in such a way that when the model sees a new data, it will from what it has learned previously will try to predict a output for it without being explicitly programmed. Machine learning models are generally of 3 types such as supervised learning, unsupervised learning and reinforcement learning.

- Supervised learning deals with labeled dataset which means every instance of the dataset is associated with a label.
- Unsupervised learning deals with unlabelled data that aims to discover patterns, structures or relationships without explicit guidance.
- Reinforcement learning concerns with an environment to learn a sequence of actions that maximize a cumulative reward signal.

There are a number of machine learning applications and has become an integral part of an average person's daily life. Some common applications of machine learning applications are recommendation systems used in online platforms, spam filtering, fraud detection systems in bank, etc. A few of well known machine learning models are random forest, artificial neural networks, k-nearest neighbours, decision trees etc. It is important to highlight that the use or selection of a machine learning model or a combination of it also known as ensemble learning depends completely on the use case and sometimes the type of data. One of the variant of artificial neural network known as LSTM (long-short term memory) is perfectly suited for time-series data. Since this study works with time-series dataset, this makes LSTM networks a good pick to do the prediction job for the algorithmic trading bot.

2.16 Machine learning workflow

The general adapted workflow of a machine learning includes pre-processing of the data, splitting the dataset into training and testing dataset which could be in a ratio of 80:20 (80% for training and 20% for testing) depending on the use case. In the pre-processing stage the data is cleaned up for the purposes of training the model and achieving the best results. This stage might involve steps like scaling the training feature which are generally numeric types between 0 and 1, this helps in faster convergence of optimal solution. Another steps in this stage also involves handling categorical values and so on. In the training stage which is the most important one, the model is actually given the training dataset and this is where the model mines into the dataset and tries to establish a pattern. In the testing stage the saved model from training stage is used for the prediction task. Later this predicted output values is evaluated against the original output values. It is important to shed some light on artificial neural

networks before the same thoughts are expanded on the use of LSTM networks in the research methodology section.

2.17 Introduction to artificial neural network

Artificial neural networks were inspired from the structure and function of the brain and how it perceives things. In 1943 The first mathematical model of a neuron was proposed by Warren McCulloch and Walter Pitts. Their model was a simplified representation of a biological neuron, which could take binary inputs and produce binary outputs. It laid the foundation for future developments in neural network theory. So how does an artificial neural network work? An artificial neural network consists of interconnected nodes or neurons and is organized into layers. A basic artificial neural network would have 3 types of layers. Layers are discussed below:

- **Input layer:** This is the first layer, where the network receives its input data. Each neuron in the input layer represents a feature or input variable.
- **Hidden layers:** These are one or more intermediate layers between the input and output layers. Neurons in hidden layers process and transform the input data through weighted connections and activation functions. Deep neural networks have multiple hidden layers, which enable them to learn complex patterns within the dataset.
- **Output layer:** The output layer produces the network's final predictions or outputs.

2.17.1 Steps involved in training neural network

The network generally is initialized with random weights and biases for each connection between neurons. These weights and biases are adjusted during training to make the network learn the desired mapping from inputs to outputs. The input values are fed into the input layer neurons. Each input neuron is connected to each neuron in the hidden layer, and each connection has a weight associated with it. The weighted sum of inputs is calculated for each neuron in the hidden layer, and a bias is added. A nonlinear activation function is applied to the weighted sum such as sigmoid or ReLU to introduce non-linearity into the model. This transformed value becomes the output of each hidden layer neuron. The process is repeated for the hidden layer to the output layer connections, using the output of the hidden layer neurons as input to the output layer neurons. Again the weighted sum, bias and activation function are applied to calculate the final output of the neural network. After passing through the hidden layers, the network produces output.

2.17.2 Loss calculation and weight updates

The network's output is compared to the desired or target output to calculate a loss or error. The loss represents the difference between the network's predic-

tions and the actual target values. The network then tries to reduce the loss by going back into the network and updating the weights and biases. The weights and biases are updated in the opposite direction of the gradient to reduce the loss. This step is where the learning occurs.

This understanding of artificial neural network will help uncover the mysteries of how LSTM networks work and how it will be optimized to achieve better results in this study.

2.18 Report Structure

In the next section the author will look at some of the work done in the past in the field of algorithmic trading using machine learning. In the research methodology section the author will outline the methodology employed, feature and target selection procedures and the adapted design of the experiments conducted for training and testing the machine learning model. In the design and implementation the author details the LSTM architecture with a focus on achieving best results by tuning the hyperparameters. In the results and discussions the author have highlighted some key findings of the experiments and then finally concluding the report and highlights space for future work to further improve the performance of the algorithmic trading bot in the conclusions section.

3 LITERATURE REVIEW

This section will explore the work that has been already done in the area of financial forecasting. This will serve as understanding the gaps, and contributions that this research seeks to address. This section presents comprehensive overview of existing research and scholarly discussions related to financial forecasting.

3.1 Determining key performance metrics in algo trading and simplistic models exploration

In [11], the authors have evaluated a couple of performance metrics for the used machine learning methods on financial forecasting. One is average accuracy and the other one is the cumulative returns on the investment. They use 6 machine learning models (OneR, JRip, Logistic model tree, C4.5, KStar and naive bayes). The initial experiments were conducted for USDJPY pair for 6 hour time frame and data is collected in the range of 6 years. The models are built for binary classification for classes being up (when the price goes up in the next trading period) and down (when the price goes down in the next trading period) used for prediction for the next trading period. Later on, similar experiments were run for EURGBP and EURUSD currency pairs. The key take away from this study and the fact that the authors have stressed about is that the rules which are used for accurate predictions are regularly evolving which makes it harder for a prediction model to work consistently. This has been

the main reason for the authors to retrain the models periodically to achieve better accuracy. Further the author have used a combinations of parameters used in model training such as technical indicators, seasonality features, price based features etc. that helped in enhancing the prediction capabilities of the model to achieve better profitability. The authors also advocates the use of simple machine learning models instead of complex models like neural networks and support vector machine which have been used heavily in the literature in this space. Although the simplicity of the models are apparent still they can do a better job at quantitive analysis in financial area. This also simplifies the usage of computational resources and avoids ordering and indexing issues in the data. Hence authors aims to introduce low complex models to generate profitable trades with similar consistency as found in the complex ones. The author debates that while the main metric to measure a machine learning model's performance is the accuracy, this might not work in the financial trading area. The most important metric considered by the authors for measuring a model's performance in financial trading is measuring the cumulative returns. The authors have compared these metrics for 6 different models. The authors found accuracy score of approximately 50%. As far as the experimental setup is concerned the authors have introduced a concept of a multi-agent system to carry out the experiments for a 2 year period. The multi-agent system is built using java based framework called BESA [12]. The machine learning models used in the experiments is integrated into the system using WEKA tools. The open source nature allows easy integration of WEKA tools. The multi-agent system consists of 2 agents, the market agent and the trader agent. The market agent gathers all the financial information such as open, close, low and high prices in every 6 hour time frame. The market agent delegates this information to trader agent and maintains an order book for such transactions to track profits and orders. The trader agent then based on the input outputs open and closed trades at every 6 hours period. The binary classifiers inside trader agent predicts whether the price will go up or down in the next trading period. The result from the classifier is used to take long or short position for the instrument. The training set has 5191 instances starting from 2nd Jan, 2002 till 29th Dec 2006. The test set has 2510 instances starting from 18th Jan, 2007 till 22nd June 2009. The 9 attributes selected are closing price, hour, day of the week, percentage of change in price, lagged percentage of change in price, lagged percentage of change in moving average (10 periods), relative strength index, Williams%R, class. The authors have argued that even though a model presents a better accuracy rate does not imply it will fetch profits in the long run. The trading strategy needs to ensure the profits needs to overcome the additional charges associated with the trading such as slippage, commissions etc. Even though the model's accuracy looks good but ends up in losses is a major setback for this performance metric. Hence the cumulative returns over a period of time proves to be a better performance metric in this study. Cumulative returns takes into consideration the accuracy achieved by the model in the prediction task as well as a weighted average of successfully executed trades.

The results of the experiments are determined using the following metrics:

- Accuracy: Accuracy rate.
- UP Accuracy: Next trading period's UP predictions rate.
- DOWN ACCURACY: Next trading period's DOWN predictions rate.
- Cumulative return: Accumulated returns percentage at the close of each trading cycle.
- Maximum drawdown: This is the maximum incurred loss by the trader agent in a trading period.
- Average return per trade: Average of all the returns attained at closure of each trade.
- Long accuracy: Total profits gained when long position was taken.
- Short accuracy: Total profits gained when short position was taken.

The test was conducted with a total of 2510 trades. The results with single training were poor, when both cumulative returns and accuracy was taken into consideration. The retraining of the model resulted in profitable trading. To validate these results, similar tests were performed for GPBUSD and EURUSD pairs.

3.1.1 Results with single training and retraining

First set of experiments is conducted with 5191 instances. An accuracy of 51.5% was maximum and cumulative returns achieved were 31.96%. All the models have reported negative cumulative returns except for OneR which is trained with Williams%R oscillator. OneR was extremely profitable in downwards periods for this experiment with 0.2691% average return per trades for short trades and 0.06357% average returns for long trades.

The hypothesis of periodic retraining might increase the average accuracy was key motivation for the next set of experiments. Periodic retraining was conducted for every 50 periods. Cumulative returns for C4.5 was highest at 87.5%.

3.2 Employing CEFLANN

In [13], author uses computational efficient functional link artificial neural network (CEFLANN) along with a set of rules to generate trading signals. The trading prediction problem here is divided into 3 classes i.e buy, sell or hold. The CEFLANN generates continuous trading signals within the range of 0-1 by analyzing various technical indicators and establishing a non-linear relationships between them. The model performance is also compared with other machine learning models such as decision trees, k-nearest neighbours, support vector machine and naive bayesian. The authors have proposed extreme learning machine (ELM) technique instead of back propagation method within the CEFLANN.

As an input 6 commonly used technical indicators have been used. CEFLANN tries to find the non-linear relationships between the technical indicators. During training instead of using discrete class values, values between 0 and 1 are fed to the network. The final output by the model is done in buy, sell or hold classes. The model's performance metric is measured on the basis of the average of the profits generated during the test period. CEFLANN is a single layer artificial neural network which has 2 main components i.e Learning component and Functional expansion block (FEB) component. A set of nonlinear trigonometric functions are used in FEB to capture the nonlinear relationships in the input set. Due to the single FEB it results in high rate of convergence and in less computational cost. The 6 technical indicators used as input are Simple Moving Average (MA), Moving Average Convergence and Divergence (MACD), Stochastic K, Stochastic D, Relative Strength Index (RSI), Larry William's R%.

The data used for training is of 5 years of historical data of BSE SENSEX and S&P 500 from Jan 2010 to Dec 2014. The datasets are divided into training and test sets i.e 1000 patterns for training and 208 for test in BSE and 1000 for training and 221 for test in S&P 500. Authors have derived a few specific rules for uptrend or downtrend identification. If closing price leads MA to rise for 5 days then it confirms a uptrend and if it falls for 5 days then it is a downtrend. After obtaining the trend confirmation following rules are used to determine the result:

- If the next day trend is Uptrend then decision is BUY.
- If the BUY decision already exists then HOLD.
- If the next day trend is equal Downtrend then decision is SELL.

If the SELL decision already exists then HOLD. As a result CEFLANN generated 47.2007% profit which was highest comparing to other models. Second highest was Naive Bayesian's at 42.3267%.

3.3 LSTMs in algo trading

In [14], authors have used long short-term memory (LSTM) technique for the trading prediction task. LSTM is considered as state of the art technique when it comes to sequential learning. The experiments have been conducted on S&P 500 stocks from 1992 till 2015. The daily returns confirmed by the authors is 0.46% which is before the transaction costs. The authors compared it with memory-free classification models and found to be outperforming them. The other models used for comparison were random forest, logistic regression and deep neural network. The authors defines the study period as training-trading set. The training set consists of 750 days data which is from 3 years and trading set consists of data from 250 days which is close to 1 year. The entire data from 1990 till 2015 is split into 23 of these study periods. The data is collected for approximately 500 stocks from S&P 500 index. The authors have calculated returns for each day and stacked them in a vector which has dimensions total

stocks (500) and number of days in study periods (1000). As such each study period consists of 380000 samples in which 255000 is for training set and rest 125000 is for trading set. As for binary classification is concerned the authors have created 2 classes 0 and 1. Class 0 is assigned to a stock in a time period if the return value of stock is less than the cross sectional median of all stocks returns in the next time period and if its equal or greater than class 1 is assigned. Further authors highlighted the some of the hyperparameters for the training of the model. RMSProp is used as optimizer, 1000 epochs and early stopping at 10. The overall topology is given below:

- Single featured input layer with 240 time-steps.
- LSTM layer with $h=25$ hidden neurons.
- Dropout value at 0.1.
- Output layer with 2 neurons and activation function is chosen as softmax.

The daily returns registered by LSTM was at 0.46% and other models used for benchmarking such as random forest, Deep neural network and logistic regression were at 0.43%, 0.32% and 0.26% respectively. Making LSTM the outperforming model.

In [15] The authors have employed random forest and LSTM networks methods to analyze the average returns from the investments. For benchmarking the authors have used the work in [14] and instead of using a single feature have taken an approach of using a multi-feature to improve the performance. The multi feature consists of the returns as described in [14] in addition to closing prices and opening prices for each trading period. The results were promising as 0.64% returns were registered for LSTM bettering the previous work of 0.46%. In case of random forest the figures were again improved by registering 0.54% returns when compared to previous work of 0.43%. The data is collected from SP index from Jan 1990 till Dec 2018 using bloomberg. The dataset is split into 26 study periods each consisting of 756 days of training period and 252 days of trading period. For comparison sake the authors have kept most of the things similar to the work in [14]. The hyperparameters for LSTM though would be interesting to highlight here. They are as follows:

- Categorical cross-entropy is chosen for loss function.
- RMSProp is chosen as optimizer.
- 512 is the chosen batch-size.
- Early stopping set at 10 epochs.
- Validation split set at 0.2.

4 RESEARCH METHODOLOGY

Algorithmic trading, often referred to as algotrading, has revolutionized financial markets in recent years. The sophisticated algorithms and advanced computing technologies has enabled traders to execute complex strategies with unparalleled precision and speed. As financial markets become increasingly digital and interconnected, the role of algorithmic trading in shaping market dynamics and influencing asset prices has grown substantially. Consequently, understanding the intricacies of algorithmic trading strategies and their impact on financial markets has become a focal point of contemporary financial research. This study delves into the realm of algorithmic trading with a specific focus on the design, development, and evaluation of an algorithmic trading bot's performance. The objective of this research is to explore the effectiveness of algorithmic trading strategies in the dynamic and volatile landscape of modern financial markets. By developing an algorithmic trading bot, this study seek to gain insights into the potential advantages and limitations of algorithmic trading in achieving investment objectives.

In this research methodology section, the author outlines the approach and methodology employed to conduct this research. This section details the data collection methods, algorithm design, and testing procedures used to develop and assess the trading bot's performance.

This section serves as a roadmap for the subsequent chapters, where the author present the empirical results, analysis, and implications of this research. By the end of this study, the author aims to contribute to the broader discourse on algorithmic trading while providing valuable insights for practitioners and investors navigating the complexities of algorithmic trading strategies.

This section is further divided into few subsections. They are as:

- Research approach.
- Data collection.
- Algorithm design.
- LSTM networks.
- Software and hardware specifications.
- Timelines
- Testing and Real-time Simulation

4.1 Research approach

This study employs a quantitative research approach to investigate the development and performance of an algorithmic trading bot in the context of the NASDAQ100 stock market index. The NASDAQ100 is a stock market index that represents a diverse group of 100 of the largest non financial companies

listed on NASDAQ stock exchange. It is one of the most widely followed equity index and is often used as a benchmark for the technology and growth sectors of the U.S stock market. The quantitative approach is well-suited for this research due to its ability to analyze large datasets, derive statistical insights, and develop predictive models, aligning with the nature of algorithmic trading.

4.2 Data collection

The dataset used for this study was sourced from Kaggle and covers historical data of the NASDAQ100 index from year 1962 till date [17]. Although for simplicity of this study the author have considered training and testing models on data from 2012 to 2020. Another addition to dataset was added which composes prices of NASDAQ100 index itself for same dates [18]. This was required to test a hypothesis that there exists a correlation among the constituent stocks prices and the index's prices itself. This index price data could also help in predicting the price movement of a stock. The dataset includes information on stock prices such as open, close, low, and high price of a stock on a daily basis.

4.2.1 Dataset description

- Open - Opening price of a stock on a given day.
- Close - Closing price of a stock on a given day.
- Low - Lowest price of stock on a given day.
- High - Highest price of a stock on a given day.
- Nasdaq_open - Opening price of NASDAQ100 on a given day.
- Nasdaq_close - Closing price of NASDAQ100 on a given day.

The nasdaq_open and nasdaq_close columns are added separately by a merging procedure using python programming language.

4.3 Algorithm design

To assess the effectiveness of the algorithmic trading bot, the dataset was divided into 2 distinct periods: a training period comprising of approximately 750 days (3 years) for training and a testing period comprising of approximately 250 days (1 Year). This makes it a total of 6 study periods with non-overlapping testing periods considering data from 2012 till 2020. The sequence length for training is selected as 240 and hence each window of this sequence length consists of 240 consecutive days of information. The information is nothing but the intraday return of each stock for each day. Another feature which is added in this window is NASDAQ100 intraday return corresponding to same date. The information in this window is then used to predict the next day's intraday return. The dates having intraday returns for all the past 240 consecutive days are selected for training, meaning initial 240 days in training period is discarded since they do

not have all previous days data. This makes it a total of approximately 500 days of data in each training period. After the first sequence of 240 days is processed, rolling window is then shifted by 1 time period which is 1 trading day and same process is repeated. This way each study period has a total of 500 such rolling windows for training approximately. The last 250 days in a study period is used for testing and predicting the intraday return for the next day. For the next study period the entire process above is repeated for a year ahead from the previous study period's start year. The data is sorted by dates and stocks in the data preparation stage. Only those stocks which has data for all days present in a study period are filtered out. Each study period has a total of approximately 50 stocks which fits the criteria. This makes a total of 25000 sequences for training and 12500 sequences for testing in each study period.

Feature vector

Date	1	2	3	4	...	237	238	239	240	241	242	243	...
$\bar{R}_t^i$	0.057	-0.451	-1.336	0.095	...	0.687	-0.300	-0.415	-0.515	-0.438	-0.173	2.455	...

Stock s_1	1	2	3	4	...	237	238	239	240
	0.057	-0.451	-1.336	0.095	...	0.687	-0.300	-0.415	-0.515

Sequence 1

Stock s_2	1	2	3	4	...
	0.418	-2.335	-2.161	1.246	...

Stock s_1	2	3	4	...	237	238	239	240	241
	-0.451	-1.336	0.095	...	0.687	-0.300	-0.415	-0.515	-0.438

Sequence 2

Stock s_2	2	3	4	...
	-2.335	-2.161	1.246	...

Figure 8: Each sequence construction

As the figure 8 displays, each sequence has past 240 days of intraday return information. This sequence is present for each of the stocks in the study period for all the dates.

The training and test data contains 2 features. This dataset preparation for the training is pretty much similar and inspired from the works of the authors in [14] and [15]. In [15] author has added 3 features and achieves better results than the result in [14]. In this paper in addition to the intraday returns in [14] the author further added a new feature for the training i.e. the NASDAQ100 intraday returns. With this addition the author wants to test the hypothesis that this correlation could help achieve better returns.

4.3.1 Features selection

The following is the list of the features selected for this study and its description:

- Intraday returns: The price change with respect to the current day's closing price.
- NASDAQ100 intraday returns: The price change of NASDAQ100 with respect to the NASDAQ100 current day's closing price.

4.3.2 Target selection

Each stock at a same date is divided into 2 classes. First the cross-sectional median is calculated for all stocks at each date. The stock with intraday return less than the median is classed as 0 and the stock with intraday return equal or higher than cross-sectional median is classed as 1.

4.4 LSTM networks

LSTM also known as long-short term memory networks is a variant of RNN (recurrent neural network). While feed-forward neural networks are excellent for tasks with fixed-size input and output, they fall short when dealing with sequential data, such as time-series dataset used in this study. RNN were introduced to address this limitation. RNNs have loops within their architecture [19], allowing them to maintain a hidden state that captures information from previous time steps. This recurrent connection enables RNNs to model sequences and dependencies over time. While RNNs are proven effective for sequential tasks, they suffer from vanishing gradient problem. When training RNNs on long sequences, gradients can become too small during backpropagation, making it challenging to learn long-term dependencies. LSTM were introduced as a solution to the problems associated with RNNs. LSTMs are a specific type of RNN architecture designed to capture long-range dependencies and handle sequences of varying lengths more effectively.

LSTMs include several key components. They are explained below:

- Cell State (Ct): LSTMs introduce the concept of a cell state, which is a memory that can be explicitly written to, read from and modified. The cell state allows LSTMs to capture long-range dependencies by selectively preserving or discarding information.
- Hidden State (ht): This represents the hidden state at a particular time step, just like in a standard RNN. However the LSTM's hidden state is more complex because it includes both the hidden state and the cell state.
- Gates: LSTMs use 3 gates to control the flow of information:
 - Forget gate (ft): Determines which information from the previous cell state should be forgotten or retained.
 - Input gate (it): Controls what new information should be added to the cell state.
 - Output gate (ot): Decides what part of the cell state should be revealed as the output.

4.4.1 Step by step LSTM walk through

The first step in LSTM is where it decides what information to preserve and what to throw away from cell state. A sigmoid layer does this job known as

forget gate layer. The output is calculated and scaled between 0 and 1 for each number of the cell state C_{t-1} . 1 implies preserve the information and 0 implies removal of the information. This is shown in figure 9

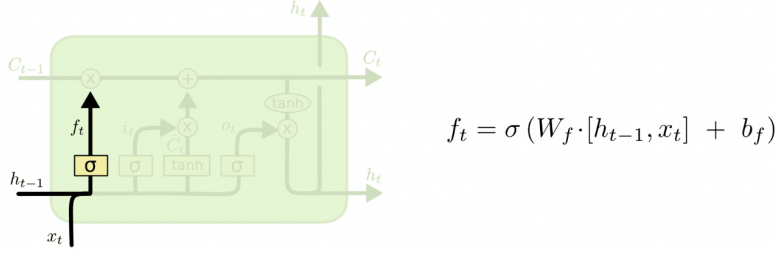


Figure 9: Forget gate

The next step is a combination of a sigmoid layer also known as "input gate layer" and a tanh layer that generates new candidate values C_t as shown in figure 10.

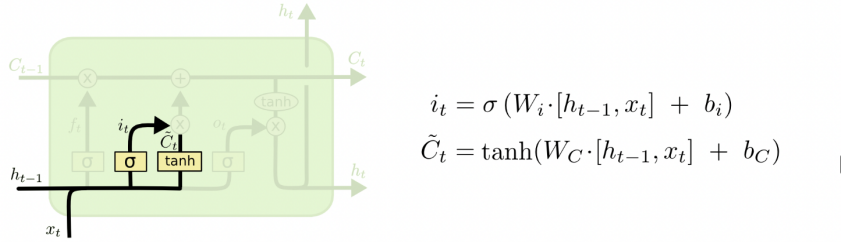


Figure 10: Input gate

Next the old cell state i.e. C_{t-1} is updated by doing a point-wise multiplication of the old state with f_t , which decides which information to be forgotten and then add it * C_t as shown in figure 11.

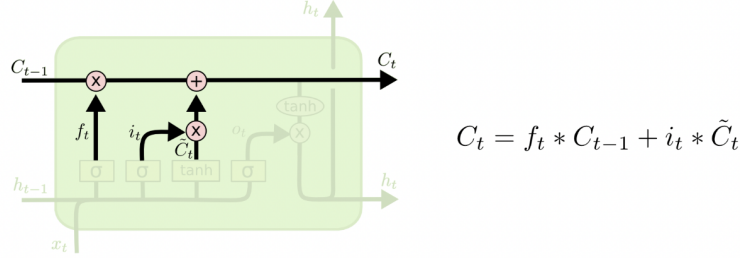


Figure 11: Cell state

Finally the output uses a sigmoid layer to decide what part of the cell state would be filtered, then a tanh is applied to cell state to scale the values between -1 and 1 as shown in figure 12.

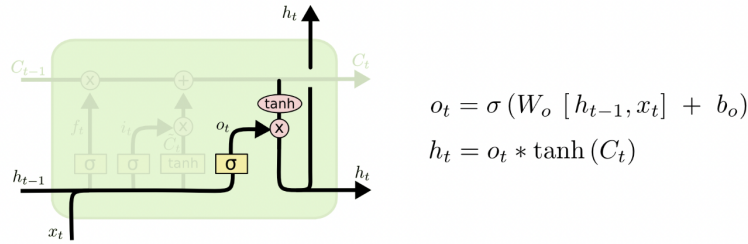


Figure 12: Output gate

4.5 Software and hardware specifications

4.5.1 Hardware specifications

The algorithmic trading bot development were conducted on an Apple MacBook Pro (2022 model). The hardware specifications of the system used for this research are as follows:

- Processor: Apple M2 chip
- RAM: 8 GB
- Operating System: macOS Ventura (13.4.1)

4.5.2 Software environment

- Programming language: Primary programming language: Python (Version 3.8)

- Development environment: Integrated development environment (IDE): Anaconda Navigator; Anaconda Version: 5.3.1; Anaconda Navigator Version: 2.10.4;
- Operating System: macOS Ventura (13.4.1)

4.5.3 Libraries and frameworks

The following python libraries and frameworks were instrumental in the development of the algorithmic trading bot:

- Tensorflow: An open-source machine learning framework used for building and training deep learning models.
- Pandas: A data manipulation and analysis library for handling and processing structured data.
- NumPy: A fundamental package for scientific computing with Python, providing support for numerical operations and arrays.
- Jupyter notebooks: An interactive computing environment for creating and sharing documents that contain live code, equations, visualizations and narrative text.
-

4.5.4 Version Control

Git (Version 2.30.1) The Git version control system was used to manage the codebase, track changes and facilitate collaboration.

4.6 Timelines

The successful execution of this research hinged upon a well structured timeline that ensured efficient progress and adherence to deadlines. The research timeline encompassed the following key phases:

4.6.1 Project Inception (Months 1-2)

During the initial phase, the research objectives and scope were defined. A detailed review of existing literature on algorithmic trading, financial markets, and LSTM networks was conducted to establish a solid foundation for the study.

4.6.2 Data Collection and Preprocessing (Month 0.5)

This phase involved sourcing historical NASDAQ100 data, which included stock prices, trading volumes, and relevant macroeconomic indicators. Data preprocessing techniques, such as cleaning, normalization, and feature engineering, were applied to ensure data quality and suitability for model development.

4.6.3 Model Development (Month 1)

The core of this research involved the development and fine-tuning of LSTM networks for algorithmic trading. This phase included architecture design, hyperparameter tuning, and iterative model refinement to optimize predictive accuracy.

4.6.4 Training and Evaluation (Month 0.5)

LSTM models were trained using historical data, and their performance was rigorously evaluated through simulating against historical stock price movements. This phase also included model performance comparison against traditional trading strategies.

4.6.5 Conclusion and Dissertation Writing (Month 1)

The final phase involved summarizing the research findings, drawing conclusions, and synthesizing the insights gained throughout the study. The dissertation was written, reviewed, and refined to ensure clarity and coherence.

4.7 Testing and Real-time Simulation

To assess the historical performance of the LSTM-based trading models, testing was conducted using the testing dataset. This involved simulating trading decisions based on historical data and evaluating the outcomes according to the established performance metrics.

5 DESIGN & IMPLEMENTATIONS

In this study, the time-series dataset used is from the year 2012 till 2020 from NASDAQ100 index. The author has employed LSTM networks to train and test this time-series dataset. In this section, the author elucidates the intricacies of the implementation phase. Jupyter notebook was used as a IDE to conduct all the experiments. The data for each of the constituent stock of NASDAQ100 is stored in a folder named `nasdaq100_stock_data` at the root directory level. As a initial step it was important that the data is structured well before it is being fed to the LSTM networks to train and test. The original dataset comprises data from 1962 till current date. So a filtering method is written to filter out data for years between 2012 and 2020. This data is then stored as a dataframe after loading from the csv files. As a next step the NASDAQ100 stock prices is merged by dates along with the filtered stock data. This is required to test one of the hypothesis that the index price data has a correlation on its constituent stocks and might help achieve better returns if used for training the model. The new columns added after the merge are `"nasdaq_open"` and `"nasdaq_close"`. Next is the task of generating labels. At first, for each date all the stocks intraday return is calculated. Then the cross-sectional median is calculated. Then in

the end the labels are assigned to each stock instance for each date. 1 if the intraday return is greater or equal to median and 0 if the intraday return is less than median.

Next step is the most important step where the features are generated. The intraday returns of the stock and the nasdaq index intraday returns are calculated. Since the sequence length is set to 240, 240 columns for each date is generated for each feature making it a total of 480 columns in addition to existing columns. Each date now has 240 days of each stocks intraday return data. For example if total dates are 1011 in a study period and approximately 50 stocks data is available, then total rows would be 50550 approximately containing more than 480 columns. Each row has past 240 days of intraday return data. Thus the initial 240 days are discarded from the dataframe. Later the data is split into training and testing data by keeping approximately 500 days of data for training for each stock and approximately 250 for testing for each stock. An important feature is also added here which is the next day's intraday return which is used at the time evaluating the performance of the model. This feature is named "intraday_return_future". As a next step the data is scaled using "MinMaxScaler" function. "MinMaxScaler" is a data preprocessing technique to scale numerical features within a specified range, typically between 0 and 1. The range is the difference between the maximum and minimum in a column/feature.

5.1 Training the model

In this part of the study deals with the most important step which is training of the LSTM networks. Although the final product would only shows the code which achieved the best return. The code is tweaked many times to reach till that stage. The parameters used in the experiment is discussed below:

- Loss function: Categorical cross-entropy. A loss function also known as a cost function is a fundamental component of many machine learning algorithms. Its primary purpose is to quantify how well a model's predictions match the actual target values. The choice of a loss function depends on the specific task and the type of problem at hand. Categorical cross-entropy loss function is used in multi-class classification problems. It quantifies the difference between the predicted class probabilities and the true class labels.
- Optimizer: RMSProp with learning rate of 0.001. An optimizer is an algorithm used to adjust the parameters of a model during training in order to minimize the value of a loss function. The primary goal of an optimizer is to find a optimal set of model parameters that result in the best possible performance in minimizing prediction error. RMSProp also known as "Root Mean Square Propagation" is an optimization algorithm that adapts the learning rates for each parameter based on the magnitude of their gradients.

- **Batch size: 512** The batch size parameter is a hyperparameter that specifies the number of data samples to be used in each iteration of training when it is being fit into the neural network model. It controls how many data points are processed in parallel before the model's weights are updated. It also affect factors like convergence speed and memory usage.
- **Epochs: 10** An epoch is another hyperparameter that specifies the number of times the entire training dataset is passed forward and backward through the neural network. Each complete pass through the entire dataset is referred to as one epoch.
- **Validation split: 0.2** This parameter determines the partitioning ratio of the training and validation set during training. This technique is used to assess model's performance and prevent overfitting during the training process.
- **Activation function: softmax** An activation function is a mathematical function that introduces non-linearity into the model. Activation functions are crucial because they determine the output of a neuron based on its weighted sum of inputs. The non-linearity allows neural networks to approximate complex, non-linear relationships between input and output data. Softmax function is typically used in the output layer of a neural network, it is used to convert the output scores of a neural network into probability distributions over multiple classes.

5.2 Testing the model

After the model is trained, the model predicts the price for the next day using the testing data for each study period. This predicted probability for the class 1 for each of the stock for each date is then stored in a python object. The probability data is made available as part of the model's prediction method. For this study in the binary classification case, the object returned from the model's prediction method carries in its first index the probability for 0 and the second index carries the probability of class being 1.

The next step is where the predicted data is evaluated against the original intraday return data. The "simulate" method loops through all the dates, in each iteration it first filters out all stock's data for that particular date and gets the next day's intraday returns data which is stored in "intraday_return_future" column created in feature generation stage. Next all the stocks are sorted in descending order based on the probabilities of them being of class 1. In this case the top k stocks, (k being 10 and 5 in different sets of experiment) are selected for longing i.e. for buying the stocks on that particular day and flop k stocks are selected for selling. This is followed by calculating the mean of all intraday returns for top k stocks. This process is repeated for all the study periods between 2012 and 2020. In the last part of the code a chart is prepared displaying the mean of intraday returns for each year.

6 RESULTS & DISCUSSIONS

This section discusses the different experimental setups used in this study in detail and also tries to reflect on the performance of the experiments conducted in this study. The purpose of this trading bot was to study the patterns in the NASDAQ100 constituent stocks and pick the top k best performing stock to open a long position and in some experiments the flop k stocks were picked to open a short position.

In the previous works such as in [14] and [15] the authors conducted experiments for a longer duration i.e 23 study period equalling 26 years. The top 10 stocks were used for opening a long position and flop 10 stocks were used for shorting. This resulted in average returns of 0.46%. Although with the limitations with hardware specifications in this study restricted the author to filter out stocks in shorter duration in comparison to the previous works resulted in relatively lower dataset size used for training and evaluating performance. The epoch used in previous studies was set to 1000 but for this study it was restricted to 10 due to the memory limitations. Each round of experiment consumed around 30 minutes. So these hardware based limitations justifies the lower performance achieved compared to previous studies.

6.1 Experiment 1 with k=10 used for only long positions

The initial experiments were carried out with 1 feature i.e only consisting of the intraday returns. The top 10 stocks were selected for opening a long position and sell the next day. The overall returns for all the years were 0.02% for this experiment.

In the second experiment which takes 2 features as input achieves 0.03% average returns for the years 2012 to 2020. Which was slightly better than the previous experiment with 1 feature.

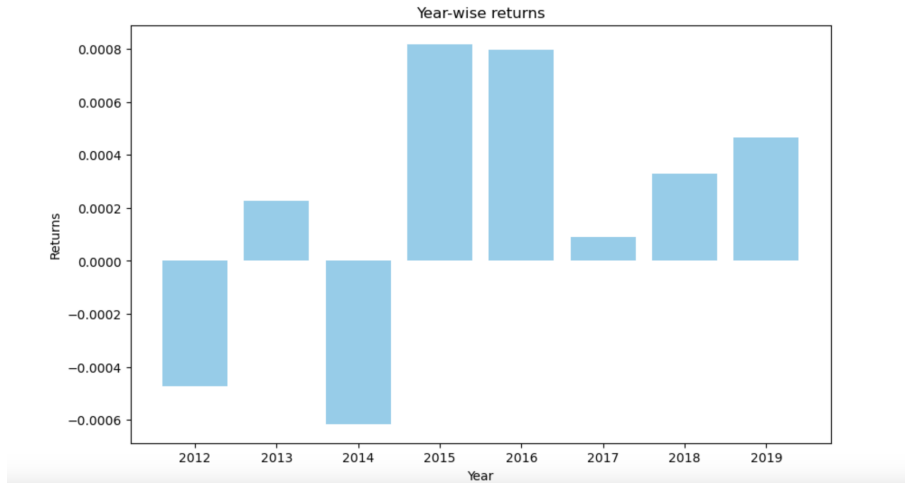


Figure 13: 1-Feature - long positions - $k=10$

Figure 13 shows the results year wise. First experiment with 1 feature 2014 was the year with the lowest average returns of -0.06%. Whereas 2015 was best at 0.08% average returns. The year 2012 as well has given negative returns just like year 2014. Rest all the years stayed positive.



Figure 14: 2-Feature - long positions - $k=10$

Now let's look at the figure 14 which shows results for 2-feature setup that is with the addition of NASDAQ100 intraday returns. This experiment has shown slight improvement hence proving the hypothesis that there is a correlation between the NASDAQ100 index price movements and the constituent stocks

price movements. This is important observation because the previous studies miss out on utilising such relationships for better predictions.

The year 2012 and 2014 again has shown negative average returns. Year 2012 is approximately 0.04% which is the lowest and 2014 with 0.03% approximately. The year 2016 has shown better average returns at 0.1%. As evident the bot's performance on NASDAQ100 stocks in the years 2012 and 2014 were relatively poor, resulting in negative average returns. To utilize on negative returns, one possible solution is to open short positions such stocks, which leads this study to the next set of experiments where top 10 stocks would be used for taking long positions and flop 10 would be used for taking short positions. In short position as explained in the introduction part the trader borrows stocks from a broker which in return borrows from a institution and then buys back when the price is lower and returns it to the broker. The difference in the sell and buy price results in gain or loss in the trade.

6.2 Experiment 2 with $k=10$ used for long and short positions

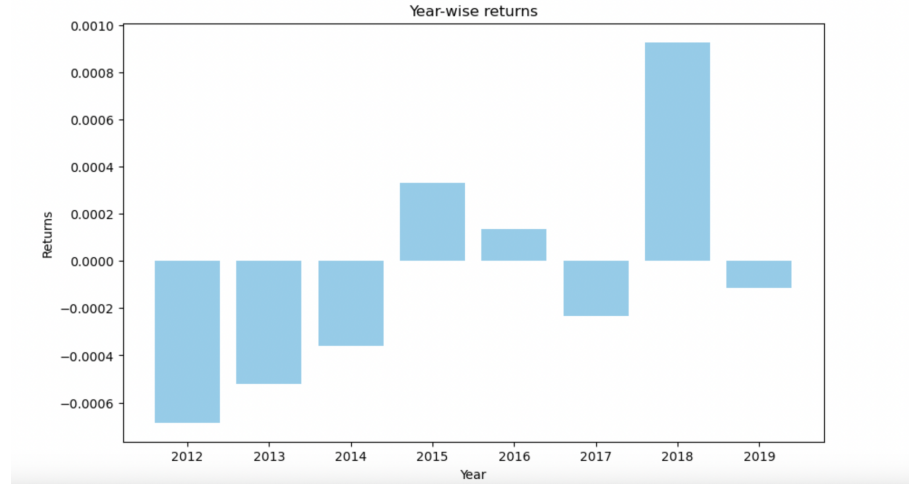


Figure 15: 1-Feature - long/short positions - $k=10$

The figure 15 shows the result for experiment with 1 feature which resulted in negative returns of -0.006% which is close to 0 but the year 2018 has shown pretty good returns around 0.1%.

As part of next experiment, it would be a good option to conduct the same experiments for 2 feature setup.

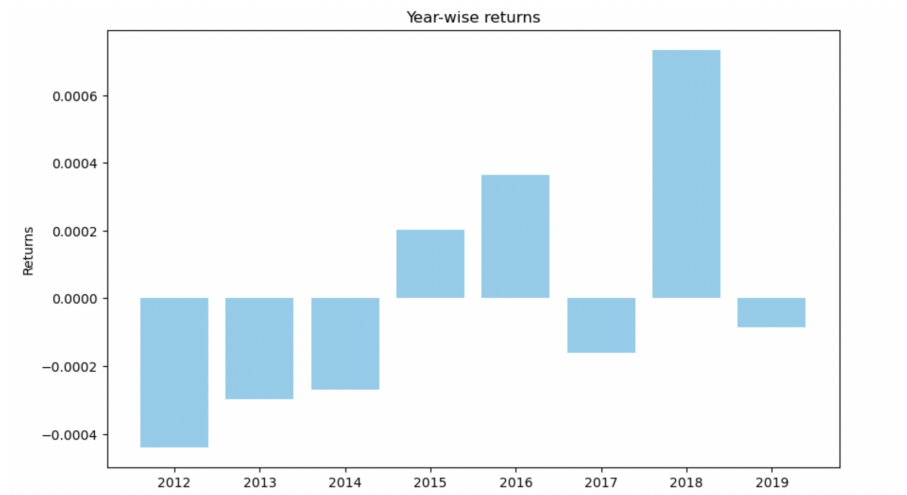


Figure 16: 2-Feature - long/short positions - $k=10$

As evident in the figure 16, the results again show slight improvement with the addition of the feature. The overall returns registered is 0.0005%. To receive improved returns looking at the landscape of the experiment a better idea would be to reduce the k value to 5. This might eliminate the poor performing stocks and keep the best returns, which might show improvement in the overall performance.

6.3 Experiment 3 with $k=5$ used for long positions

So, let's first start with 1-feature setup and try only longing the stocks.

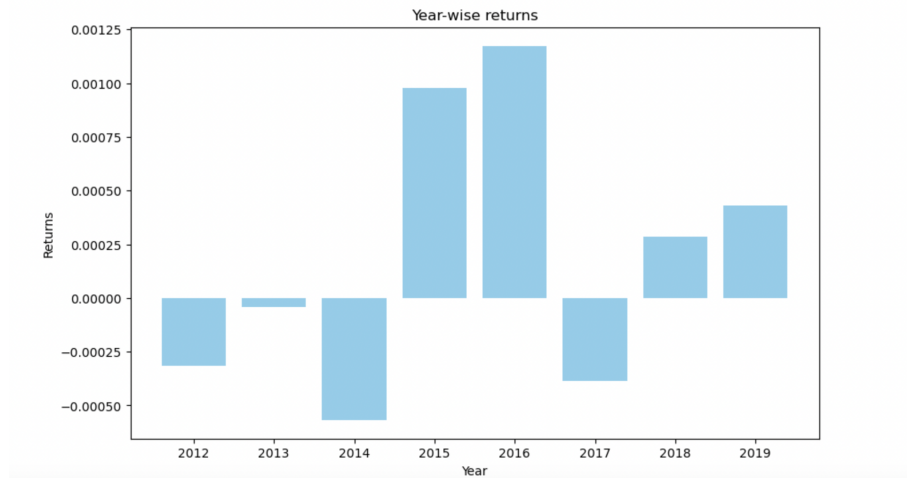


Figure 17: 1-Feature - long positions - $k=5$

As the figure 17 shows the 1-feature setup with k set to 5 receives approximately 0.02% returns which is same as previous result when k was 5.

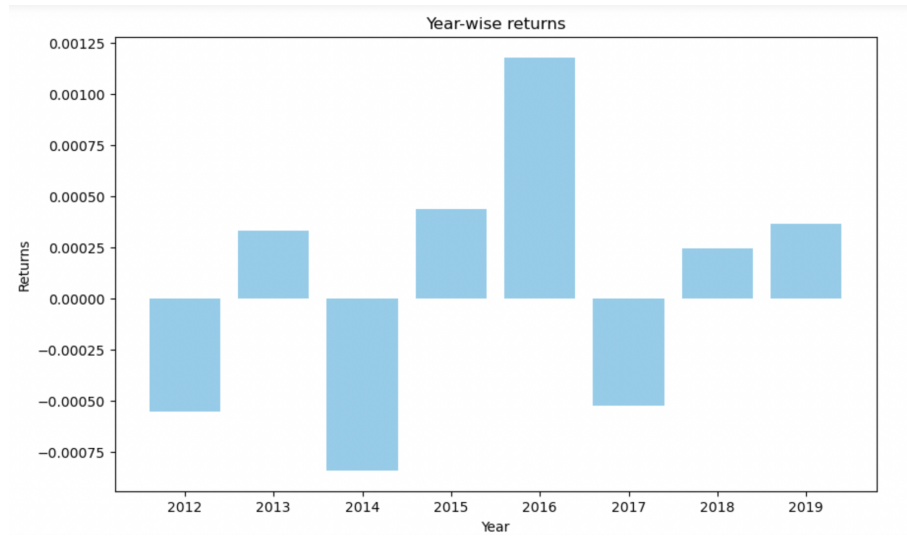


Figure 18: 2-Feature - long positions - $k=5$

6.4 Validation/training loss charts for best performing experiment

It is concluded that the first set of experiments conducted with only taking the long positions with k equals 10 fetched the best results overall for both 1-feature and 2-feature setup. So it would be a good idea to also have a look into the best performing model's accuracy and loss graphs for these experiments to get a clue that where the improvement could be made. For this the author has utilized "matplotlib" library to visualise the validation and loss for each epoch.

A training loss trendline represents the loss on the training dataset, which measures how well the model is fitting the training data. Similarly validation loss trendline represents the loss on a validation dataset that model has never seen before or during training. This helps to assess that how well the model has generalized to new, unseen data. In an ideal scenario, both training and validation loss should decrease together. This indicates that the model is learning and generalizing well. If the training loss continues to decrease while validation loss remains stable or increases, this is clear indication of overfitting. Overfitting means that the model has learned to fit the training data well and is not generalizing to the new unseen data. If both the training and validation loss are high and not decreasing significantly, it indicates underfitting. Underfitting means the model is too simple to capture the underlying patterns in the data. As a fix to the overfitting issues, early stopping can be applied to the training. As soon as the validation loss starts increasing after a certain number of epochs, training could be stopped to prevent overfitting and save the model. In this experiment the epoch is already set at 10 which is quite a low number generally, so the need to add early stopping in this study is ruled out. Now that an understanding of validation and training loss is established, lets jump into the graphs of top and worst performing years and try to interpret the performance at each epoch.

Now lets look at the top and worst performing years for the 2-feature setup when only long positions were taken since this setup was proved out to be outperforming the rest.

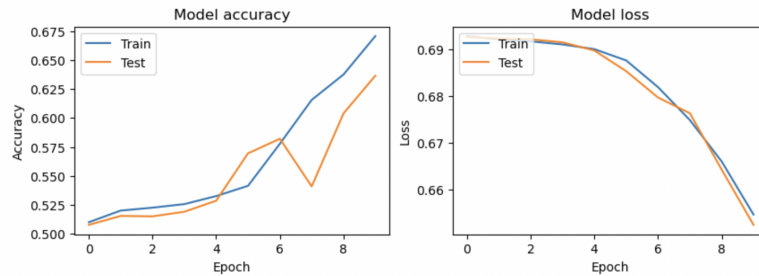


Figure 19: Validation/training loss - 2-feature - 2012 (long positions)

The figure 19 shows the validation/training loss chart for the year 2012 which was the worst performing in all of the years. Both the trendlines are moving along well in tandem showing no signs of overfitting or underfitting.

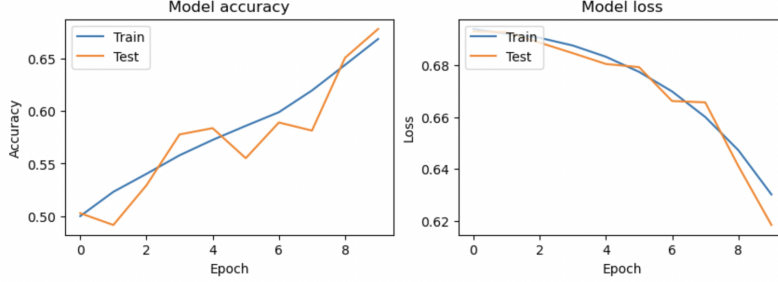


Figure 20: Validation/training loss - 2-feature - 2016 (long positions)

The figure 20 shows the training/validation loss trendlines for year 2016 which resulted in the 0.1% average returns being the best in all the years. As evident there are again no signs of overfitting or underfitting.

7 CONCLUSIONS & RECOMMENDATIONS

In this study, the author embarked on a comprehensive exploration of algorithmic trading, aiming to develop and evaluate an algorithmic trading bot tailored for NASDAQ100 index. Through an extensive literature review and the design and implementation of the trading bot, this research sought to contribute to the evolving field of financial technology. The key findings and contributions of this study are discussed below.

7.1 Algorithmic development

The dataset selected in this study was from NASDAQ100 index which is quite popular for its technology sector based constituent stocks. The dataset used for training and testing was belonging to years between 2012 and 2020. Each study period in each of the experiments conducted had 3 years of the data for training and next 1 year data was used for testing making it a total of 6 study period. The experiments were conducted with 2 different setups. 1 had only intraday returns and the other one had the intraday returns for each stock at each day and the corresponding intraday return for the NASDAQ100 index for same date. The data was labelled with 2 classes i.e. 0 if the intraday return for all stocks for each day is less than the cross-sectional median and 1 if it was equal of higher. Each sequence of data was of 240 time periods close to a year time. The trained model was tested on testing instance to predict the next day's intraday return. Overall mean of average was used to evaluate the performance.

7.2 Machine learning model

The author employed LSTM networks to do the job of the prediction, since it is a well-known machine learning model for time-series based datasets and is a variant of RNN where it uses a few gates to keep/discard some information from the past. Since RNN faces the issue of vanishing/exploding gradients, LSTM networks are known to address this issue. A very simple architecture of LSTM networks is used in this study mostly inspired from the previous works in [14] and [15].

7.3 Performance evaluation

As established in the [11], the authors have demonstrated well that the average accuracy of the trained model cannot be taken into consideration when evaluating performance of a trading bot. The cumulative returns on the investment is the best way to evaluate the performance in this space and thus in this study the average returns have been the key performance metric. There were a number of experiments conducted with various combinations of features and longing and shorting of the stocks. The best setup in this study was proved to be of 0.03% returns on the investment. Not the best but given the limitations of the hardware specifications such as RAM of 8GB the epoch was chosen to set at 10, which does not give enough room for the networks to learn but can be used to approximate the returns and also highlights the need of future work with the usage of the right hardware specifications to achieve better results. Another reason for this lower performance is the number of stocks for NASDAQ100 for each year were close to 50, whereas in the previous works [14] and [15] the authors worked with approximately 500 stocks dataset for a period of 26 years.

7.4 Future works

The author believes although the results achieved would have been not satisfactory in this study but this definitely opens avenues for future research and development in algorithmic trading. The learning gained from these experiments could be extended to create a well established production ready algorithmic trading bot. First and foremost is the use of GPUs (Graphics Processing Units) to enhance training performance. GPUs offer significant advantages over traditional CPUs due to their parallel processing capabilities. GPUs are designed with thousands of small cores that can perform multiple calculations simultaneously. This parallel architecture can speed up the training times significantly. Another way to improve performance is addition of the features that could prove to improve performance significantly. Features such as macroeconomic features. Macroeconomic features refer to key economic indicators and variables that describe the overall performance of an economy. These features provides insights to the health and growth prospects of a nation's economy. It plays a vital role in stock prediction and are often incorporated into the predictive models to provide economic context for stock market movements. A few of

the macroeconomic features such as commodities such as oil, gold, agricultural products, etc. might have direct impact on some sectors. As such these commodity prices could be integrated with the existing features to prove that the correlation exists and achieve better returns. It is important to be aware that the sentiment analysis can as well have a significant impact of stock prices and financial markets. It is a natural language processing (NLP) technique used to gauge public sentiment, emotions and opinions expressed in the text data. The news plays a crucial role in stock price movements. Positive news can drive the prices up whereas negative news can lead them to fall. This can be used as an additional feature to predict the stock price movements.

In conclusion this study demonstrates the potential of algorithmic trading in the financial markets. Although it is nice to have a algorithmic trading bot do all sorts of trades for an individual or an organization it is equally important to acknowledge the inherent risks associated with algorithmic trading and the need for continuous refinement and adaptation.

8 REFLECTIONS

Throughout the course of this dissertation, the author embarked on a journey to explore the intricacies of financial markets, with a particular focus on the NASDAQ100 index. This endeavor aimed to bridge the gap between theory and practice, as the author harnessed the knowledge to develop and train Long Short-Term Memory (LSTM) networks for algorithmic trading. In this section, the author reflects on the key aspects of this journey, the lessons learned, and the implications of the findings.

8.1 Research Process and NASDAQ100 Analysis

The research process began with an in-depth study of the financial markets, stock market and the factors affecting its price movements. Then the author analysed the NASDAQ100, a dynamic and influential index comprising top technology and internet-based companies. The author delved into historical data, examining price movements, volatility, and underlying factors that shape the market. This phase provided valuable insights into the market's behavior and helped the author identify potential patterns that could be exploited through algorithmic trading.

One of the fundamental lessons from this analysis was the importance of understanding market sentiment in shaping price movements. While technical analysis played a crucial role in identifying trends, it became evident that a holistic approach, encompassing both quantitative and qualitative factors, was necessary to make informed trading decisions.

8.2 Development and Training of LSTM Networks

The core of this dissertation revolved around the development and training of LSTM networks for predictive modeling in algorithmic trading. This endeavor involved not only mastering the intricacies of deep learning but also integrating financial data into the model effectively.

During the development phase, the author encountered various challenges, such as data preprocessing, hyperparameter tuning and the hardware limitations. These challenges underscored the importance of model robustness and a suitable environment to run the experiments on a large dataset. Iterative refinement was necessary to achieve meaningful results.

As the author trained LSTM networks on historical NASDAQ100 data, it was observed the power of neural networks in capturing complex temporal dependencies within the financial time series. However, through past work it was also realized the need for continuous model monitoring and adaptation to changing market conditions.

8.3 Personal and Academic Growth

This dissertation has been a transformative experience in terms of personal and academic growth. It has not only deepened the author's understanding of financial markets but also honed the skills in data analysis, machine learning, and algorithmic trading. The journey has taught the author the value of perseverance, adaptability, and continuous learning in the rapidly changing field of finance and technology.

In conclusion, this dissertation represents a significant milestone in the author's academic journey, where the author applied theoretical knowledge to real-world financial data. The blend of quantitative analysis, machine learning, and financial expertise has been both challenging and rewarding. It is the author's hope that the insights gained from this research will contribute to the advancement of algorithmic trading strategies and pave the way for further exploration in this exciting intersection of finance and technology.

References

- [1] Callender, C., Davis, S. Graduates' responses to student loan debt in England: "sort of like an acceptance, but with anxiety attached". High Educ (2023). <https://doi.org/10.1007/s10734-023-01045-5>
- [2] Annamaria Lusardi, Financial literacy and the need for financial education: evidence and implications, 2019
- [3] John Murphy, Chapter 3, Charting made easy, 2013

- [4] Angel Ann Varghese, Dr.Krishnadas J, Dr. R. Satheesh Kumar, Candlestick Chart Based Stock Analysis System using Ensemble Learning, 2023
- [5] Carol Osler, Support for Resistance: Technical Analysis and Intraday Exchange Rates, 2000
- [6] Younes Chihab , Zineb Bousbaa, Marouane Chihab, Omar Bencharef, and Soumia Ziti, “Algo-Trading Strategy for Intraday Foreign Exchange Speculation Based on Random Forest and Probit Regression“, 2019
- [7] Cristian Păuna, CAPITAL AND RISK MANAGEMENT FOR AUTOMATED TRADING SYSTEMS, 2018
- [8] Wiley-Blackwell, Chapter 5, Risk Management and Financial Institutions, 2012
- [9] Roland John Wiley, YChapter 5, CMT Level I 2017: An Introduction to Technical Analysis, 2017
- [10] Rashesh Vaidya, Moving Average Convergence-Divergence (MACD) Trading Rule: An Application in Nepalese Stock Market(NEPSE), 2020
- [11] Eduardo A. Gerleina,c,, Martin McGinnitya,b, Ammar Belatrechea, Sonya Colemana, Evaluating machine learning classification for financial trading: An empirical approach, 2016
- [12] González, Enrique, Jamir Avila, and César Bustacara. ”BESA: Behavior-Oriented, Event-Driven, Social-Based Agent Framework.” PDPTA. Vol. 3. 2003.
- [13] Rajashree Dash, Pradipta Kishore Dash, A hybrid stock trading framework integrating technical analysis with machine learning techniques, 2016
- [14] Thomas Fischer, Christopher Krauss, Deep learning with long short-term memory networks for financial market predictions, 2017
- [15] Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari Sahoo, Forecasting directional movements of stock prices for intraday trading using LSTM and random forests, 2021
- [16] Aurélien Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow by Aurélien Geron (O’Reilly), 2017
- [17] <https://www.kaggle.com/datasets/svaningelgem/nasdaq-100-daily-stock-prices>
- [18] <https://www.kaggle.com/datasets/guillemservera/sp500-nasdaq-spy-qqq-ohlc-data>
- [19] Ralf C. Staudemeyer, Eric Rothstein Morris, Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks, 2019
- [20] <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>