LSTM Stock Market Prediction on Volatile Time Frames

Mustafa Alkatib
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, United Kingdom
ec211156@qmul.ac.uk

***Abstract*—Long Short Term Memory (LSTM) are a subset of neural networks used to recognise patterns in data and are extremely valuable for stock market forecasting. With recent economic events, the stock market has experienced extremely volatile price movements. This paper uses the recent volatile stock market data to predict volatile time frames for The Boeing Company (BA) and Zoom Video Communications, Inc. (ZM). A LSTM model was built and assessed by predicting two specific time periods of volatility caused by economic events for the mentioned stocks.**

***Index Terms*—Stock Market Prediction, LSTM, RNN, Volatility**

I. Introduction

The stock market is a volatile environment and, in this case, when an asset is volatile it allows the possibility of profits to be made. With the development of computer programmes and algorithms, hedge funds and corporations have used this to their advantage to make profits. Large amounts of data are available, and this can be used to build, test and validate a model. With algorithms becoming more powerful than ever, the stock market has been dominated by A.I with over 70% of trades being placed by algorithms. However, with all the funding and research in this field, there is still a strong case that the stock market cannot be predicted. With global algorithmic trading expected to reach $18.16 trillion by 2025 there is a strong argument for humans to remain in control in the market and not let algorithms trade 99% of the market. This is because, although humans cannot predict a black swan event, algorithms are more likely to cause one. In fact, this happened in 2010 where there was a flash crash lasting 36 minutes which was caused by Navinder Singh Sarao, who used spoofing algorithms to "rapidly place and cancel orders automatically". In recent times with the evolution of technology, investors and traders have gone towards using data to aid them with making predictions in the market. Techniques such as Machine Learning and Deep Learning have been incorporated into the financial industry. Using historical price data as a training dataset, algorithms have been built and trained to make accurate predictions in the stock market. There are many different techniques that can be used to predict the market such as machine learning (ML), deep learning, Support Vector Machines (SVM), recurrent neural networks (RNN) and artificial neural networks. With machine learning and deep learning models, there needs to be an extremely large amount of data that must be inputted to the model to have a chance in predicting the price of a stock at a given date or time frame. Algorithms are yet to be fully competent when dealing with volatility. January 5th, 2015, when the Swiss National Bank had revoked its €1.20 currency cap there was a short period of volatility where traders saw a move in the market of 20%. Technical analysts have the ability to assess the market and extract patterns from the visual data to take action on their trades. Using high volumes of data, a model is more likely to perform better and make more accurate predictions. In the last few years there have been periods of extreme volatile price movements giving us a set of data that we have never seen before. With enough volatile data, can it be possible to build a model to predict the future price and movement of a stock? As we progress into the future, there is more uncertainty in the market due to various factors. Every resource that can be used to make profits in the market will be more valuable than ever, and hence the motivation behind this thesis is to see the potential that algorithms have. As we head into further uncertainty due to the current economic conditions, algorithms built to predict stocks that can compete with volatility and predict the behaviour of a stock will become highly demanded. Despite the power these algorithms have, there is still a very strong argument that technical analysts are still needed in this field, and we cannot rely on algorithms. The primary goal of this thesis is to use the most competent model to predict a stock's price in a volatile time period. Various factors can affect the price of a stock, however in this paper the focus is using nothing but the historical price data to make predictions and see how the model can perform. To summarise, the overall objective of this paper can be broken down into the following problem statement. With enough volatile data, can we rely on an algorithm to make predictions in the stock market?

II. Related Works

A. Background Research

When building a model, the first and most important part is to acquire a dataset that includes the features and context of similar attributes to what you are trying to predict. Due to the extreme periods of volatility, we have seen in the past

2 years, it is important to train a model using volatile data to predict the price of a stock during a volatile time frame. In order for a model to make predictions accurately, they learn from the past, similar to how humans learn and use pattern recognition to predict future events. Using data from the past would be classified as training data. Once the model is constructed and trained, the next step is to evaluate how the model performs. This is where we use a testing set of data and compare it to the predictions that the model has made. Algorithmic trading is where a model is built and follows a predefined set of instructions to purchase a stock by the trader, with the intention of profiting. This allows the trader to no longer manually check the markets and live prices and can rely on the algorithm to automatically place trades. Algorithmic trading has countless advantages compared to human trading. These include instant trades that are more likely to execute at the desired price, less risk of human errors, less transactional costs and reduced risk of bad decisions due to emotional and psychological led decisions [19]. Two forms of algorithmic trading involve Automated Trading Systems (ATS) and High Frequency Trading (HFT). ATS can perform large trade orders by employing a pre-established automated trading program that incorporates features like time and price. ATS is used heavily by large financial institutions. HFT erases human decision making by using high computational technology to complete large trades in milliseconds. It can analyse the market and perform trades that suit the investor's desired order. This is where financial institutions with large amounts of capital gain an advantage over the public as they can execute a high volume of trades and make tiny profits for each trade, amassing huge profits. However, with all the benefits of algorithmic trading, it is said that humans and technical analysts are in a more desired path than the current algorithms [20], due to the heuristic approach humans have [21]. Technical analysts are still in demand as they can make predictions by using past price movements and assess the market, extracting patterns from the visual data to take action on their trades. However, technical analysis does not guarantee where a stock could head to, it is just a more educated guess. Machine Learning algorithms have now become advanced to understand what the general public feeling is around a particular stock or about the economy. The use of sentiment analysis and text mining allows the algorithm to extract textual context from different internet sources like news articles and social media trends from twitter to gain an accurate understanding. This is proof of how AI and algorithms have gained an upper hand in the market but there is a secure case for humans to be demanded when placing trades in the stock exchanges [9]. Deep learning is a subcategory of machine learning, but deep learning uses neural networks allowing a model to be educated enough to train itself. Deep learning models require less human input than a machine learning model. The user can depend on a deep learning model to adjust itself and once they build the neural network to make predictions, the model can pick up patterns in the data. There are multiple different reasons why algorithms fail. Firstly, Ellis et al, state that if trades take place five seconds apart compared to five minutes apart, there is a 16% difference in accuracy in the latter [11]. The trade size is also a reason why algorithms fail, as the authors tested using 200 trades versus 10,000 trades, the algorithm performed better using smaller trades.

A company's stock price is determined by various factors, defined as follows: The market share of the company plays a large factor on its share price. If a company monopolises its industry, then it is more likely to have a large share price especially compared to its competitors. Earnings play a huge part in the company's stock price as if a company achieves its target quarterly earnings it can help gain trust from investors, hence causing an increase in price. Balance sheets also give insights into the firm's financial plans which also determine the value of said company. Another key factor is the governing members and public image of the company. A company with competent and trustworthy individuals plays a huge role in the stock price. Elon Musk is an example as his tweets and actions cause great affect to the share price of TSLA. Lastly, still one of many factors and mostly important, is the ongoing economic conditions. During times of economic growth, a stock can see itself also rising, and during a recession, like we are experiencing today, we can see company's stock price falling. Also, various other political factors can affect the stock price of the large companies, such as the War in Ukraine and inflation (which has also been affected by the War).

B. Machine Learning

Although machine learning algorithms won't need as much volume of data like deep learning models would, a competent machine learning model would still need a considerable amount of data to be inputted as training data, which also needs to be structured. Machine learning is broken into two sub-categories, supervised learning and unsupervised learning. Most prediction models for the stock market use a supervised learning approach and the model built was the same [1]. In Supervised Learning, we build a model with the goal of making a prediction on something we don't know, using data that we do already know. In this case the input for a supervised learning model would be the previous historical stock prices, and the output would be the stock price in the future, for instance next week's stock price. Supervised learning is then further divided into two categories: Regression and Classification. Regression is used to predict continuous values such as the future stock price of a company. It is a specific technique that can establish a relationship between one variable and another (or even more). Using an independent variable such as the closing price of a stock today to predict the closing price of the stock tomorrow (dependent variable). Stock market prediction generally falls into the regression category. Classification is where the model produces a label when shown a set of predictors.

C. Neural Networks

A neural network would be able to extract patterns from the input data and conclude an output which would be a

prediction of a new set of data. Introduced in 1943 by Warren McCullough and Water Pitts, neural networks are built with many interconnected layers named neurons that work together to succeed the user's goal. Neural Networks mimic the human brain [22]. A neural network with more than one hidden layer is considered a deep network. The more hidden layers the more accurate the network can be [15]. The perceptron of the neural network has three components. The inputs, which would correlate to a specific feature of the dataset. In the stock market, this could be the close price of a stock. The weights perceptron is where every input would be assigned a degree of value and importance by the neural network. A large weight would result in the input having a bigger responsibility on the output. The output is the result of the weights and the input. This would be our model's final prediction of the future share price of a selected company. A deep neural network model would have the same approach as a machine learning model. The model is built using training data with a number of epochs and then tested with the test data to obtain the model's accuracy. Neural networks (NN) are ideal to use in the stock market due to their ability to extract nonlinear trends and patterns. Neural networks outperform statistical and regression techniques when predicting stock market price movements [11]. What gives neural networks an advantage over, for instance, a machine learning model, is that they can extract unseen patterns from the training dataset that was fed as an input [12]. There are two common types of neural networks: Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). CNNs are mainly executed for image classification problems as they can extract patterns in images allowing them to detect objects or recognise the contents of an image. RNNs are advantageous for sequential or time series data. RNNs have the ability to use an internal state output as an input to the network. An initial input is entered which is passed into the network and computes the RNN internal state computation. Inside the RNN, mathematical operations occur in the hidden layer to generate the output. An input is fed inside the network with a hidden layer which feeds back to the model as another input is placed. Essentially, an input is read by the model which then updates in the hidden state which leads to an output. Recurrent neural networks use the previous components in the network series, whereas regular neural networks would presume that the inputs and outputs of the network are independent. RNNs use the previous input data and current input data to make a prediction. RNNs allow us to have more opportunities to input different types of data such as one-to-one or many-to-one. If a neural network is too complex on the data used to train the model, the NN is more prone to overfitting. This can occur if the model has too many neurons or layers. This means that the model will perform very well on the training data, but with the testing data it will fail. The same can be said if the network is trained too simply on the training data. This will lead to underfitting as for the testing data, the model will fail to perform due to it not being trained adequately in the testing phase. With a balance between the parameters and data, the network can perform well on unseen data, and this is the goal when building a neural network.

One type of RNN is the Long Short-Term Memory (LSTM). I chose to use an LSTM model to make predictions as they are one of the most productive subsets of neural networks. LSTM was first introduced in 1997 by Sepp Hochreiter and Jürgen Schmidhuber and since then have gone on to solve a variety of problems in the modern world [18]. The LSTM has the ability to hold or discard information allowing it to adapt to the context of the data. For periods of volatility, I believe that LSTM models are most suitable as they can remember the price data and make predictions more accurately. LSTM networks are more competent to remember information than RNNs, allowing them to extract patterns in the data dynamically [14]. The common issue with RNNs are that they suffer due to the vanishing gradient problem. LSTMs solve this problem by using forget gates in the hidden layer of the network to adapt to the context of the data [1]. LSTM have layers that are built from sub-layers called units. These units include cells and gates to execute classification tasks and predictions using a set of previous stock market data [2]. LSTM can examine relationships within stock market data due to its memory function. Chen et al proved that LSTM networks performed far better than feed-forward neural networks when predicting returns in China's stock market [14]. LSTMs remember information longer than traditional neural networks and can keep memory of long time lags from the input data [15].

D. Deep Learning

Inspired by the human brain, deep learning is a type of machine learning that includes a neural network with three or more layers. Deep Learning and Neural Networks can make use of hidden layers that boost the model's accuracy and improve the likelihood of a prediction being correct. As computers continue to advance so do the algorithms being built including Machine Learning, which is the ability to acquire knowledge, through extracting patterns from a training set of data to make predictions. The best models are those that work well during the deployment stage, when presented with unseen data. It is important for a model to be trained with huge amounts of data to be able to extract patterns and correlations. By using a training set of data, the model can be built and then tested using unseen data. Unfortunately, machine learning cannot process unstructured data as well as deep learning. Deep Learning algorithms can use tools such as feature extraction to clarify the prediction. This discourages the requirement for data to be pre-processed like it would have to in machine learning. A deep learning model starts off with an input layer which includes the data and ends with an output layer which is the prediction being made. However, for a deep learning model to be trained and fully utilised to make accurate predictions, extremely large amounts of data and powerful, optimised computer hardware is needed. This process can be expensive and time consuming. One example of an algorithm used in the stock market is clustering analysis. A form of deep learning which can look at features like volatility and volume

profile to make an educated decision and group certain stocks that match features into groups (clusters).

E. Literature Review

In recent years, with the rise of using algorithms in the financial markets, there have been, many different models built achieving various results. Hiransha et al performs comparisons between linear models like ARIMA and non-linear models such as neural networks [4]. The authors managed to test various models on two stock exchanges, NSE and NYSE. The results proved that Convolutional Neural Networks (CNN) performed best due to their abilities to observe changes in the prices for both stock exchanges. ARIMA failed to classify the non-linearities in the data unlike the neural network models used in this paper. Lu et al used a combination of CNN and LSTM to predict the next day's closing for price for a stock [5]. By using all the stock's price data, the authors managed to conclude that the CNN-LSTM model offered a strong reference for investors to maximise their return on investment (ROI). However, the models developed in this paper failed to incorporate the current news articles and the general feeling of the public. Wu et al also proved this in their paper where the combination of the neural networks was used to make predictions on both Taiwanese and American stocks [6]. Yadav et al, build a LSTM network and tests the network with differnt numbers of hidden layers [15]. The authors discovered that as the higher the amount of hidden layers within the network, the better the results. This causes the standard deviation of the model to decrease.

With more data being available than ever, there is more opportunity to make profits in the stock market. Data can be analysed and discover correlations and patterns between the market and sentiment data. Financial corporations can use their budget to amass big data to gain an understanding of where a stock could head to before the public knows. Today, hedge funds can purchase data from all kinds of sources to predict the direction in a stock's price. Orbital Insight is a company that obtains satellite images across the globe. The company assessed images of oil tanks and estimated how much oil is stored in tanks around the entire globe. With more oil, the lid of the tanks would affect the shadows shown in the satellite images. Orbital Insight would then sell this data to financial companies and energy companies allowing them to be up to date with the current state of the energy and oil industry. This allows these massive corporations to make decisions on whether to keep or sell a stock and as they hold large amounts of shares their decisions can affect the market. Orbital Insight also observes how many flights take off from an airport, giving their clients huge intel on how the travel industry could be impacted. This shows the true power of data and how it can be used to estimate a company's quarterly earnings, sales, production rate and even their public image before the official numbers are released. Although this could be seen as insider information, it is still declared legal allowing hedge funds to gain a huge upper hand compared to the average investor/ trader. It is extremely important for the correct data to be used for algorithms and AI to succeed. With AI dominating the market, there is a question mark over who takes responsibility if the data fed into the model is out of context and leads to losses for traders and hedge funds. As we move forward into the future, AI can adopt more human-like characteristics leading to humans switching to communication operations and AI performing high intense thinking tasks [9]. There are strong unethical concerns over these large financial institutions using big data as they acquire data in unethical and non-consensual methods leading to a loss in privacy for humans. Algorithms can also cause changes in market activity, increasing risk of investments [24]. In the short run using algorithms and AI may lead to more profits in the stock market but in the long run, humans could be the ones who lose out.

III. Methodology

A. Data

The data was extracted from YahooFinance and tested my model to see how it performs against different sets of stocks. The data was extracted for multiple different stocks and the timeframe was the entire set of historical prices for each individual stock. I chose to download the data in a csv format and upload it onto GoogleColab using Python3 to load the data and store it as the variable 'prices'. I felt more comfortable using static data as my model is a tool used to test my hypothesis of how LSTM could perform against volatile data. If I were to use this model every day to make investment decisions, I would have switched to using dynamic data instead which would be the YahooFinance API. Researching neural networks and deep learning I found a key requirement for training a model. It is stated for the model to perform well, a requisite is for high volumes of data to be inputted into the model. When training my model, I started off with using the all-time data of multiple stocks to see how the model would extrapolate the data at hand. The outcomes of this will be discussed in the results section of the paper. It is important to assess the context of the data that is being fed into the model. A model that is designed for unsuitable data won't have a good performance [13]. Theoretically I started with using all of a stocks data, from when it was first initially traded in the market, as the input to the model. Before trialling the model, I assumed that the more data the better the outcome. I started off with finding high volume traded stocks that are considered popular and had been on the market for a significant time frame. This resulted in using a superset of stocks that included the SP 500 ETF (SPY), Advanced Micro Devices (AMD), Tesla (TSLA), Meta Platforms (META), Alphabet (GOOGL) and Amazon.com (AMZN). I initially thought that the model would perform best using these stocks due to the amount of data available and the longevity of them being publicly traded. These selected stocks are also more volatile than the rest and this is important as using volatile data in the model would help improve the performance. Since 2020, there has been extreme uncertainty, with Covid-19 being declared a Pandemic, the US election and the recent Russian invasion into Ukraine. As stated previously, the stability of the economy

and world events plays a huge role in the market. As investors and traders speculate future events, the market will constantly be changing and during this time we saw the March 2020 stock market crash due to the pandemic. Since then, the stocks mentioned above saw all-time highs and large amounts of volatility. It is believed that algorithms struggle to compete against the volatility of the market and the results describe how the model performs [23]. However, with enough data to train the model, an LSTM has the advantage to learn the patterns within the stock chosen and make predictions. I then proceeded to specifically predict two different stocks: Zoom Video Communications (ZM) and The Boeing Company (BA). The reasons for choosing these stocks were due to how they reacted during specific time frames. Before the pandemic, ZM had very slow market activity after initially trading publicly in April 2019. Trading at $107.47 just before the World Health Organization (WHO) declared COVID-19 a pandemic in March 2020, there was a huge spike in the share price. ZM saw a 425% rise in its stock price reaching an all-time high of $559 in October 2020. However, during the period where the Pfizer-BioNTech COVID-19 Vaccine was announced in November 2020, ZM saw a 20% drop in its stock price. During this period of volatility, I felt that using ZM was an applicable stock to test how my model would perform. I then also chose to assess BA. BA and Airbus have a duopoly in the aircraft industry. It almost seemed certain that once travel restrictions had come into effect during the pandemic, the public would panic and sell off BA shares. BA had started to see a sharp fall in its price in February 2020 just before the pandemic was announced. BA saw a 70% fall in its price from February 2020 to the historical stock market crash on March 16th, 2020. However, over the past few years, since 2017, the company has been trading fairly volatile. I used BA's stock price from 2017 to predict this huge stock market crash. I expect the model to underperform since it is running on nothing but the 'Close' price data to make predictions. The LSTM model wouldn't know of a pandemic with its univariate design, and theoretically the model should underperform when predicting BA's stock.

B. Libraries and Python

Python was the language used in my model as it contained the essential libraries needed to perform stock market prediction using LSTM. In addition, Python has become one of the fastest growing languages and this project was an opportunity to build my skills and become more proficient in the financial industry. The goal of my model was to build the most competent model for stock market prediction. From my research I had decided that an LSTM model was the most optimal to achieve an accurate prediction for volatile data. In order to accomplish the task, I imported the following essential libraries: Pandas was used to import the csv data file for a stock and store it in the variable 'prices'. Matplotlib was used to plot the stock graphs and create visualisations for the training, validation and test data. This was key to creating graphs that show how my model performed. Numpy was the next library used with the intention of converting the data into a matrix vector that was then used as the input for the model. Datetime was needed in the pre-processing stage of the dataset. Tensorflow and Keras are the most influential libraries as they gave me the opportunity to build my LSTM model in python and further details will be explained later in the paper.

C. Data Management

When importing the data obtained from YahooFinance, there were many variables included in the data frame such as 'Date', 'Open', 'High', 'Low', 'Close', 'Adj Close' and 'Volume'. Since the model is predicting the closing price of the stock most of these variables can be discarded. Only the 'Date' and 'Close' variables will be used as input data in the LSTM model. Since the goal of the model was to use time series data (in this case the previous stock prices) to predict the price of a stock, I found that using the 'Close' price which is the final price recorded on a specific date, to be satisfactory enough when testing the model. The objective was to identify if an LSTM model is competent enough to make accurate predictions on where a stock would head to. For an investor this model would give them an insight of where a stock could be in the next couple weeks or months. Despite using one variable after building this model there are many other resources one could use to make even more accurate predictions about the stock market such as sentiment data, but the motivation behind this model was to see how it would comprehend using time series data only. In order to move on to the prediction stage, the data had to be pre-processed. Since the 'Date' data was a string, not a date I had to convert this data type into a date using the python library Datetime. Using a function this achieved the goal of converting the data type into a date. The next stage was to create a function that creates a new data frame which holds the target date and matches with the target price. Within the data frame contains three columns which include the price of the stock one day before the target price, two days before the target price and three days before the target price. These are the input data, and the output would be the target price. This is a basic supervised learning problem by setting the target date with the target price and gathering the last 3 days as input to predict the target price (output). After creating this supervised learning data frame, the next stage of the model was to convert this data into NumPy arrays which would be fed into the tensorflow model. A function would be created with the goal of generating a list of dates, a three-dimensional input matrix containing the previous price data and an output vector which would be the target price we are trying to predict. Since this model is only using the 'Close' price as an input variable this would be considered univariate stock market forecasting. The next stage was to split the data into training, validation and test data. The training data would train the model, the validation data would assist in training the model and the testing data is used to evaluate the true performance of the model. The training data will be split into the first 80% of the input data. The validation data is 80% - 90% of the input data. Lastly, the test data would be the final

10% of the input data, which previously mentioned would assess the model's performance [1].

D. LSTM Network Architecture

LSTM, unlike an RNNs, can decide which information is deemed as important and should be kept, and which information should be discarded. In RNNs, the current input and previous output which is known as the hidden state are joined together in a vector form which proceeds to go through an activation function forming a new output. This output becomes the new updated hidden state. As data is imputed, the network operates to process the data sequentially through forward propagation. Within the cells of LSTMs, the network is intelligent enough to determine which information to discard or retain within the model [18]. Inside the LSTM, the network is broken into cell states and gates that have different roles. Firstly, the cell state acts as a path which smoothly transfers information throughout the network. The advantage LSTM has over RNNs is here in the cell state, as the network has the ability to convey information from previous points in time throughout the entire network. This is why LSTM models are extremely valuable for time series data. In the stock market during periods of volatility the network can pick up patterns and store this information when trying to make predictions on the future stock price. Where most algorithms fail to predict the stock price, LSTMs shine as they are able to take the context of the data into consideration. This internally is what allows LSTM networks to minimise the consequences of short term memory. An issue RNNs face. Moving on to the responsibilities of the gates, these are neural networks that dictate which information should proceed in the cell state. During training, the gates are intelligent to control which information is relevant enough to keep or to discard. Inside the gates are activation functions and there are three different gates that synchronise how the information flows [15]. Firstly, we have the forget gate which dictates which information should be kept or discarded. The previous hidden state $h_{(t-1)}$ and information from the current input is processed into a sigmoid function which outputs values between zero and one [6]. In this computation the closer to zero means to keep the information and one means to forget. After this, the sigmoid layer then outputs numbers that fall between zero, which tells the network to let everything through, and one which does not let anything through to the cell state. The next gate is the input gate which uses the previous hidden state and the current input and proceeds to the sigmoid function. This computation determines which part of the information is deemed important and not important. The closer the value is to 1 the more importance it has. Further computations are performed, the hidden state and current input is used to help regulate the network. This is called the tanh function where the values are essentially compressed between -1 and 1. This output is then multiplied with the sigmoid output. After this process the network has enough information to proceed to the next stage. The cell state is then multiplied by the forget vector and the model takes the output from the input gate and completes a polarise addition, updating the cell state to the new values. Finally, we have the output gate which controls what the next hidden state should be. The output is the new hidden state which is passed again throughout the network for the process to be repeated again.

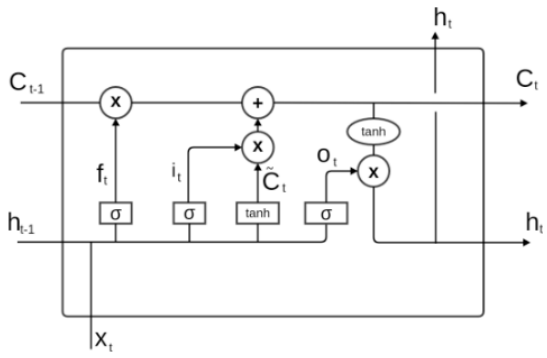

Fig. 1. LSTM Cell

E. LSTM Model

Moving on to the next stage, this is where the LSTM model is built and tested. As mentioned previously TensorFlow is imported as well as Keras to build a sequential model, optimise the model and use many layers to achieve accuracy within the model's output. When building the model, I used 64 as the parameter for my LSTM model. This would be the number of neurons that the network holds. After trial and error this number was best suited in terms of achieving accuracy within all three components of the split data. Increasing the number of neurons would increase the complexity of the LSTM model, therefore leading to a higher chance of overfitting. While it's nice to see the model be incredibly accurate for the training data this would negatively impact the results of the test data and is then counterintuitive to the goal of the model. Using less neurons would lead to the model underfitting which then negatively impacts the output. Using more hidden layers leads to better results for the LSTM network [15]. The activation function used in this LSTM model is linear. The next step of the model would be to minimise the loss function and, in this case, I have chosen to minimise the Mean Squared Error (MSE). Then using 100 epochs, the model is fitted using the training data and validation data. Using more epochs allows for better accuracy when trying to predict the price of a stock [8]. Moghar and Hamiche show this when building their model, using 100 epochs performed far better than 50, 25 and 12 and their results show that their precision increases. When using large datasets this process was time inefficient but would cause the loss function to decrease. With more epochs the mean absolute error decreases. I chose to then assess the mean absolute error for the validation data and can see that as the model is run through the 100 epochs it significantly falls until it hits a stale point.

IV. Results and Discussion

Using the knowledge obtained from learning about neural networks and deep learning models, I assumed that the more data the model has, the better it will perform when making predictions. I observed how the model would perform using the all-time data for many stocks, and the LSTM model failed drastically. For instance, with Tesla (TSLA), using data from 2012 to predict the current prices, despite good performance in the training stage the model failed to predict recent prices. This happened on all stocks, not just TSLA, where the entire stock dataset was used. This is because in the training stage the model had not seen the extremely high stock prices. In the last 2 years it has been an incredible period for an investor in the stock market. When deploying the model on TSLA, the training data was only up to the first half of 2020, and the all-time high during this period was roughly $300. However, in the testing portion of the dataset, the company reached an all-time high of $1222.09. After seeing these results, I noticed that the model was not suited to using all of the stocks data. The goal of this thesis was to analyse how a model would perform with volatile data. I felt that using data from 01-01-2020 would be most suitable to predict today's prices and the model performed well across many different stocks. I used this period as a general test to see how the model would perform. I was interested to see if the model would also fail due to the extreme volatility, we have experienced in the last 2 years. Despite this the model performed extremely well, predicting trends in the market and had been fairly accurate. This gave me an insight that the model was competent to compete with volatility and made me move on to the next stage of testing to see if the model is able to predict a specific volatile period. After testing many different stocks, some more volatile than others, I had decided to use two stocks to perform a test on my model. As stated in the previous section, the two stocks were Zoom (ZM) and Boeing (BA). The metric used to value how the model performed was the RMSE. I chose to run the LSTM model multiple times with the same parameters to see how it would perform.

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

Fig. 2. RMSE Equation

TABLE I
Result of LSTM Model

LSTM Model Results	
BA RMSE	ZM RMSE
32.93383	20.685482
17.62617	22.56781
17.56224	26.539343
44.126698	22.772053
15.386299	24.993017
34.435787	19.722157

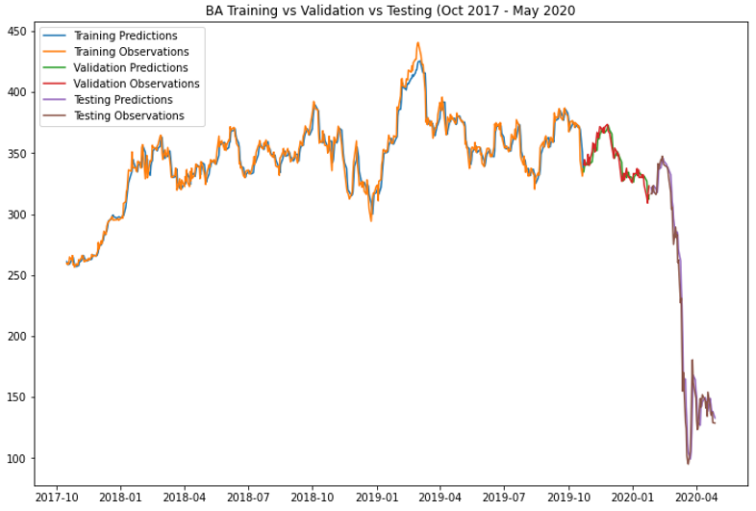

Fig. 3. BA Stock

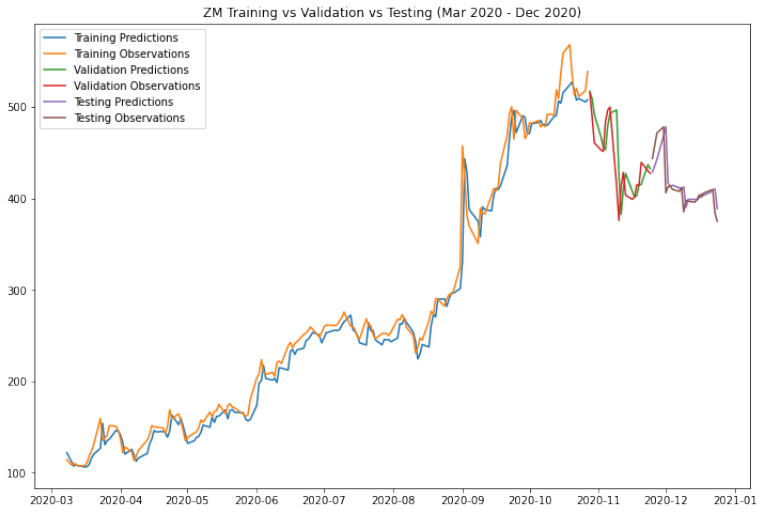

Fig. 4. ZM Stock

RMSE and MSE were used to observe how the model performed. Running the model multiple times with the same parameters obtained different results because the LSTM neural network goes through different pathways. The more times you run the model the more sets of RMSE and MSE values are obtained. My model performed better on BA, and this is indicated in the results table where the lowest RMSE was 15.386 for BA, and the lowest for ZM was 19.722. This is because BA had more volatile data, which proves that the more data a model has as input, the better it performs. The graphs below show how the model predicted the price movements of both stocks, and it is clear to see that BA performed better despite both stocks achieving accurate predictions of the direction and price of the stock. With BA having more volatile data fed into the LSTM network, the model managed to achieve better performance despite predicting bigger market activity.

Figure 3 represents BA's stock and figure 4 represents ZM's stock. Starting off with BA, the goal of the model was to predict the huge drop in March 2020, the testing data clearly shows the model succeeded in predicting this. By using the previous three days to predict the target date, the LSTM model is able to store this valuable information and prove its competence with a visual proof. By using nothing other than the previous closing price of the stock, the model managed to predict the huge crash in the aircraft manufacturer's stock. This shows that with enough suitable data the model is proficient enough to make an accurate prediction on the stock's price movement. With no sentiment data applied to this model I believe the results are impressive and show that LSTM networks are able to compete with volatile datasets. However, the correct data must be included in the network to perform. Using BA's all time data the model extremely underperformed and using data from only 2020 failed to predict the March 2020 crash. There must be an educated approach when inputting data into the network. One must analyse the appropriate timeframe to use and find data with as much volatility to be able to predict a volatile timeframe. For BA I looked at the data in YahooFinance and using the trendline tool, I was able to find many periods of volatility. Hence using data from 2017 to predict the March 2020 crash, this set of data worked very well as shown in figure 3. In figure 4 Zoom had seen a huge spike in its stock price during March 2020 due to the pandemic. This period of extreme volatility was another great opportunity to test how the model would perform. As shown in the graph the model was able to predict the continuation in the fall in value of ZM's share price. This crash was due to many factors; however, the announcement of the vaccine caused investors to lose confidence with ZM, due to the belief of interactions becoming normal again after being taken away from us. However, the model wasn't as accurate as BA's predictions due to the model having less data to use. BA had over 2 years prior to ZM's dataset to use and in turn helped make the model more accurate when making a prediction. Overall, the model performed well and gave insights on how powerful LSTM is for time series data. It is still impossible to predict what will happen with a stock, but the core basics of this model can give an investor a subtle prediction of where a stock could head to. One must consider which data to use to make a prediction. The appendix shows my model tested on various different stocks and it is clear that using data from 2020 to make predictions in recent times, the model performs accurately and can compete with volatile data. We have seen the highest stock prices for hundreds of stocks and even more uncertainty is yet to come with inflation in the economy being extremely high, we could see some more volatility in the market, and this could be a great opportunity to see how the model performs.

V. Conclusion

Based on the results achieved, I can state that with enough volatile data an individual can have some idea in the direction of a stock. Of course, the stock market is impossible to predict as there may be black swan events like the Wirecard scandal where the company was involved in corruption, and this caused the stock to plummet. However, as we enter the next few years, I believe that there will be even more volatility due to many economic factors. With the US Federal Reserve printing more money than ever, this will cause even more uncertainty and therefore we will potentially have even more data to observe and test this LSTM model.

References

[1] Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. Expert Systems with Applications. Volume 184.

[2] Shen, J, Shafiq M. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. . J Big Data 7. 66.

[3] Shah, A., Gor, M., Sagar, M. et al. (2022). A stock market trading framework based on deep learning architectures. Multimed Tools Appl 81, 14153–14171.

[4] Hiransha, M., Gopalakrishnan, E.A., et al. (2018). NSE Stock Market Prediction Using Deep-Learning Models. Procedia Computer Science, Volume 132.

[5] Lu, W., Li, J., Li, Y., Sun, A. and Wang, J., (2020). A CNN-LSTM-based model to forecast stock prices. Complexity 2020.

[6] Wu, J.M.T., Li, Z., Herencsar, N., Vo, B. and Lin, J.C.W., (2021). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. Multimedia Systems, pp.1-20.

[7] Roondiwala, M., Patel, H. and Varma, S., (2017). Predicting stock prices using LSTM. International Journal of Science and Research (IJSR), 6(4), pp.1754-1756.

[8] Moghar, A. and Hamiche, M., (2020). Stock market prediction using LSTM recurrent neural network. Procedia Computer Science, 170, pp.1168-1173.

[9] Ashta, A. and Herrmann, H., (2021). Artificial intelligence and fintech: An overview of opportunities and risks for banking, investments, and microfinance. Strategic Change, 30(3), pp.211-222.

[10] Ellis, K., Michaely, R. and O'Hara, M., (2000). The accuracy of trade classification rules: Evidence from Nasdaq. Journal of financial and Quantitative Analysis, 35(4), pp.529-551.

[11] Lawrence, R., (1997). Using neural networks to forecast stock market prices. University of Manitoba, 333, pp.2006-2013.

[12] Naeini, M.P., Taremian, H. and Hashemi, H.B., (2010), October. Stock market value prediction using neural networks. In 2010 international conference on computer information systems and industrial management applications (CISIM) (pp. 132-136). IEEE.

[13] Pahwa, K. and Agarwal, N., (2019), February. Stock market analysis using supervised machine learning. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) (pp. 197-200). IEEE.

[14] Chen, K., Zhou, Y. and Dai, F., (2015), October. A LSTM-based method for stock returns prediction: A case study of China stock market. In 2015 IEEE international conference on big data (big data) (pp. 2823-2824). IEEE.

[15] Yadav, A., Jha, C.K. and Sharan, A., (2020). Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science, 167, pp.2091-2100.

[16] Shah, D., Campbell, W. and Zulkernine, F.H., (2018), December. A comparative study of LSTM and DNN for stock market forecasting. In 2018 IEEE international conference on big data (big data) (pp. 4148-4155). IEEE.

[17] Grudnitski, G. and Osburn, L., (1993). Forecasting SP and gold futures prices: An application of neural networks. Journal of Futures Markets, 13(6), pp.631-643.

[18] Hochreiter, S. and Schmidhuber, J., (1997). Long short-term memory. Neural computation, 9(8), pp.1735-1780.

[19] Lo, A.W., Repin, D.V. and Steenbarger, B.N., (2005). Fear and greed in financial markets: A clinical study of day-traders. American Economic Review, 95(2), pp.352-359.

[20] Buczynski, W., Cuzzolin, F. and Sahakian, B., (2021). A review of machine learning experiments in equity investment decision-making: Why most published research findings do not live up to their promise in real life. International Journal of Data Science and Analytics, 11(3), pp.221-242.

[21] Raab, M. and Gigerenzer, G., (2015). The power of simplicity: a fast-and-frugal heuristics approach to performance science. Frontiers in psychology, 6, p.1672.
[22] Wang, S.C., (2003). Artificial neural network. In Interdisciplinary computing in java programming (pp. 81-100). Springer, Boston, MA.
[23] Kim, H.Y. and Won, C.H., (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. Expert Systems with Applications, 103, pp.25-37.
[24] Svetlova, E., (2022). AI ethics and systemic risks in finance. AI and Ethics, pp.1-13.

MSc Project - Reflective Essay

Project Title:	LSTM Stock Market Prediction on Volatile Market Pairs
Student Name:	Mustafa Alkatib
Student Number:	180452927
Supervisor Name:	Marcus Pearce
Programme of Study:	MSc Big Data Science

My project describes how algorithms are used in the stock market to make predictions and analyse the different techniques used in modern finance and computer science. The hypothesis was then stated, which was to analyse how a model would perform if it was fed with enough volatile historical stock data. The model was built using a LSTM neural network to predict periods of volatility for two separate stocks: The Boeing Company (BA) and Zoom Video Communications, Inc. (ZM). The model's results were then compared to the two stocks to see the outcome and why the model performed better for BA over ZM. This essay describes the strengths and weaknesses of my project, future works, the engineering behind the model if I had more time, the legal, and ethical issues for my work.

Approach

My motivation behind my problem statement was from trying to understand how data can benefit me as a personal investor in the stock market. I started off with learning about how data science is used in the financial markets and discovered new terminology that was unseen to me. As I have strong beliefs that the current economic state will worsen, I wanted to see if it was possible to gain further knowledge about my investments in the stock market. Learning technical analysis as a hobby over the past few years has allowed me to make high profits in the stock market in the last two years and I learnt that during periods of volatility this was the best opportunity for myself to amass profits. When initially researching topics for my project, I started learning more about how algorithms are used in the stock market. I realised that algorithms struggle during periods of volatility and with the uncertainty we are facing today I was intrigued to find out how using data can allow me to have an insight of where the market will head in the next few months. I started discovering neural networks and deep learning, a concept I never understood in detail and learned how important it is for models to be trained with high volumes of data. I believed that the reason algorithms fail to predict volatile data is that there isn't enough volatile data to train a model with. However, after assessing the movements in various stocks in my portfolio, I was encouraged to build a model using the data in the last few years to see how it could perform. The last few years have been historical with many assets in the financial markets seeing huge gains and huge losses. There has been more volatile data available than ever, and I built my model and trained it with as much volatile data as possible to gain the best predictions. I must say I am impressed with how my model performed and was worried if it would fail. I tested the model to see how it would predict two stock crashes for two different stocks in different industries and different time frames. I was most impressed with my results for

BA as it managed to predict a 70% fall in the stock's price which happened from February 2020 to March 2020. The fact the model was able to predict the crash without seeing a fall like this in the training data was very impressive. Using the past three days to predict the target price seemed to work very well and by testing other methods it seemed to work best.

Strength and Weaknesses

The strength of my project was that I managed to achieve fairly accurate results and my model performed well on the testing dataset. Overall, the model was tested on many stocks and saw very similar outcomes. I can say that the research question was answered to an acceptable standard. With enough data I strongly believe that algorithms will perform better and make more accurate predictions in the stock market. Before 2020, there wasn't as much volatility in the markets as we see today. With more data available today and with more market activity, algorithms are only going to get better and may be able to make accurate predictions that investors can rely on to make profits. As more volatile assets like Bitcoin (BTC) appear, this gives us more opportunity to test models and improve their performance. The increase in technology has allowed us to make better predictions in the market. With more data available than ever, the sky's the limit for machine learning and deep neural network models.

I would say that there were many weaknesses with my project. Despite achieving good results, I felt that I could have used more metrics to analyse how my model performed. In addition, I would have to wanted to build another model such as a SVM model, CNN or a CNN-LSTM model and compare the differences in results. This is something that would have been more possible with time on my side. Like all models the data used to train the model must be selected very specifically. It takes someone with more technical skills to understand periods of volatility in the market. If the model is fed with out of context data, for instance using the all-time data to make a prediction for today's stock price following the recent volatility in the market, the model will fail completely. This means that an individual must be able to understand stock price graphs and use tools such as trend lines to extract as many periods of volatility as possible to feed into the model.

Future Work

As time went by when completing my project and testing my model, all I wanted to do was try different things to improve the model's accuracy and see how the model could perform under different sets of parameters and stocks. After researching many papers and seeing how different models can be paired together with an LSTM model to acquire better results, with more time I felt that I could try to build several models such as a statistical model (ARIMA), a CNN-LSTM model, an artificial neural network model (ANN) as well as attempt to use support vector machines (SVM). In the financial markets there are many different combinations one can use, and the rewards seem extremely high as they can help amass profits to investors. The use of sentiment analysis is something I find very intriguing as a model could factor in how investors and the public feel about a company or the current economic conditions. This can massively

improve the model and perform better when predicting volatile market activity as the model can pick up how investors would react to a company's quarterly earnings or any rumours circulating the company's market activity. I believe that the use of sentiment analysis is extremely valuable in today's age, as the rise in social media has given incentive to share how individuals feel and a LSTM model with sentiment data can create more accurate predictions on the movement of a stock. With more time, I believe I could have run even more tests on my model, using various different stocks and changing many different parameters and testing how the model would perform on different time periods of volatility. I believe that my model is still very naive and has its limitations, but with more time and acquiring more data the model has the potential to succeed.

Legal, Ethics Issues and Sustainability

This project was built on using previous stock market data meaning that confidential data obtained from individuals was not needed. However, there is still a risk of using my model when making predictions in the stock market. Since the motive behind purchasing stocks is to make profits and avoid losses, there is a huge risk to use this model when dealing with large funds. At the top of the financial industry where corporations are using billions of dollars to place trades, relying on algorithms and models places huge responsibility on the developers of the model. This means that although my model performed well on ZM and BA there is still not enough evidence to indicate the model is reliable. In times of black swan events there is absolutely no way the LSTM model can rely on previous closing prices to predict this event. Perhaps pairing it with sentiment data can help. The GameStop (GME) saga is a great example where redditors caused the price of the stock to shoot up after hedge funds started shorting the stock. Relying on a univariate LSTM model would be irresponsible. It is important to warn investors or traders that using this model is only to gain an insight on the stock market and is not a financial tool to use and trust. Therefore, this means that myself or any developer of a stock prediction model shouldn't be liable to any legal issues. By stating that this model should not be relied upon for financial investments, this should remove most ethical issues associated with the model. However, the model could be used with many alternative variables such as the volume of the stock, sentiment data or even other techniques like CNNs with LSTM to make a more accurate prediction. As mentioned previously, we are living in uncertain times and although the model performed well on predicting the movement of ZM and BA we could see even bigger market crashes in the next few years. As much as my model can help gain an insight on the direction of the market, it can also lead to a loss on an investment. AI and algorithm can cause reactions in the market creating a dangerous chance for investors to lose money due to spoofing algorithms, for example [4]. After learning the value of data and researching how many companies sell data to financial corporations, it can lead to very unethical ways of acquiring data. Orbital Insight is a great example as they sell data using satellite images giving these large institutions a huge upper hand against the public. This means that these companies can have a very educated prediction of where a stock could be and with all the research a regular investor could do, they would never know what these large financial companies know. With profits being the end goal

for placing trades in the stock market, this could lead to even more unethical ways of data being acquired. One big issue I see happening with the stock market and big data in the future.

There has been a rise in awareness for the environment with carbon emissions worsening, deep learning models are being used every day. Deep learning and machine learning models depend on very heavy computational resources, that negatively impact the environment [2]. Using GoogleColab allowed me to keep my model sustainable due to how models are run on the Google Cloud. The cloud warrants to help reduce carbon emissions released from the GPUs by using more sustainable technologies such as soler panels. This was announced in the latest Google Carbon Offset programme. This allows my model to be more efficient and reduce the effects caused to the environment. My model's total power consumption can be calculated using the following formula:

Power consumption x Time x Carbon Produced Based on the Local Power Grid =

300W x 12.5h = **3.75 kWh** x 0.62 kg eq. CO_2/kWh = **2.33 kg eq. CO_2**

2.33 kg of CO_2eq was produced for my model which would be equivalent to 9.42 kilometres driven by an average ICE vehicle.

Personal Development

I challenged myself in this project by firstly building a model to predict volatile periods after initially believing the model will fail. I had to find the best approach to building model and neural networks were the key to building my model. With not studying neural networks or deep learning at an academic level, I had to self-teach the core concepts of a very technical subject within data science. I must say I am very happy I chose this project as it made even more educated and opened my eyes to how powerful data is in the modern world. I am grateful to my supervisor Marcus Pearce, for guiding me in this project and assisting me when I presented issues with my thesis and model. I look forward to testing my model even more on more stocks, more volatile time periods and maybe one day purchase or sell a stock using insight from a LSTM model.

References:

[1] Ashta, A. and Herrmann, H., (2021). Artificial intelligence and fintech: An overview of opportunities and risks for banking, investments, and microfinance. Strategic Change, 30(3), pp.211-222.

[2] Strubell, E., Ganesh, A. and McCallum, A., (2019). Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*.

[3] Static.googleusercontent.com. (2022). [online] <https://static.googleusercontent.com/media/www.google.com/en//green/pdfs/google-carbon-offsets.pdf

[4] Svetlova, E., (2022). AI ethics and systemic risks in finance. *AI and Ethics*, pp.1-13.