The impact of machine learning in the retail financial market

CM3203 - BSc Project
BSc Computer Science with Security and Forensics
School of Computer Science and Informatics, Cardiff University

Otto Hooper
Student ID: 1949120
Project Supervisor: Dr Oktay Karakus

Abstract

This report investigates the development and implementation of a deep learning algorithm using Long Short-Term Memory (LSTM) networks for predicting the next-day closing price of stocks based on both short-term intraday- and historical closing prices. The study includes a literature review exploring the principles related to stock market behaviour and Machine Learning's (ML) potential impacts on the financial sector, specifically retail and institutional investors. The LSTM model is tested on low-end volatile stocks with high market capitalisation, Apple (NASDAQ: AAPL) for example, and is compared to alternative ML models. Results demonstrate high accuracy rates from the LSTM algorithm, highlighting its promising predictions on short-term closing intraday prices. The S&P Capital IQ Pro platform is also examined from an institutional investor's perspective, with a web dashboard presented to visualise the predictions and historical data. By analysing the potential risks and challenges arising from ML advancements, valuable insights are provided for future research and policy discussions in the financial field alongside risk management.

Acknowledgement

I would like to extend my heartfelt appreciation to my supervisor, Dr Oktay Karakus, and Dr Kathryn Jones for their guidance and assistance in generating ideas for this project. I am also grateful to S&P for granting me complimentary access to S&P Capital IQ Pro, which allowed me to gain insights into the resources available to institutional investors.

I would also like to express my gratitude to Leander Ingham Hjertvik, an undergraduate student from the BI Norwegian Business School, for generously sponsoring the IEXCloud data subscription and peer-reviewing the report.

Furthermore, I wish to thank Dr Anqi Liu and Dr Maggie Chen from the School of Mathematics for their valuable suggestions regarding alternative data providers that could be well-suited for the project.

Table of Contents

List of Tables

List Of Figures

List of Abbreviations

Abbreviation	Full form
AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central processing unit
CSS	Cascading Style Sheets
DOM	Document Object Model
ESM	ECMAScript
Forex	Foreign Exchange Markets
GDP	Gross Domestic Product
GPU	Graphics processing unit
GRU	Gated Recurrent Unit Network
HTTP	Hypertext Transfer Protocol
IDE	Integrated development environment
IPO	Initial Price Offering
LSTM	Long-Short Term Algorithm
M&A	Mergers & Acquisitions
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
ORM	Object-Relational Mapping
R2	R Squared
RCE	Remote Code Execution
REST	Representational state transfer

RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
S&D	Supply and Demand
SEC	Security Exchange Commission
SMTP	Simple Mail Transfer Protocol
SPAC	Special Purpose Acquisition Company
SSE	Server-sent events
URL	Uniform Resource Locator
UTP	Unlisted Trading Privileges

1.Introduction

This report seeks to offer valuable insights regarding the opportunities and challenges with ML for policymakers, industry practitioners, and individual investors, who navigate the evolving landscape of retail finance.

The impacts of ML on the retail finance market will be identified, along with its potential to revolutionise the economy. The current applications of ML in finance will be reviewed while addressing the hypothetical risks associated with its adoption. This paper also focuses on evaluating the accessibility of ML-driven tools and platforms for retail investors, identifying the barriers to their access, and exploring strategies to democratise finance.

1.1.Introduction to Machine Learning

In recent years, ML has seen significant advancements, leading to improved algorithm efficiency and an increasing range of applications. As a result, ML has become more accessible to the general population through open-source libraries and freemium services like OpenAI's popular ChatGPT and now more recently the company released GPT-4, which can beat 90% of people in certain US exams, such as the Uniform Bar Exam (OpenAI 2023). There are community-driven projects like Open Assistant and GPT4All as well, these developments have democratised access to powerful tools and enabled users from various backgrounds to harness the potential of ML technology.

One area where ML has gained prominence is the retail financial market, which includes consumer banking, lending, and investing. Historically, ML has been employed in the finance sector to enhance risk assessment (Bracke et al. 2019), fraud detection (Guida 2018), and investment management, primarily for professional entities such as hedge funds, banks, and institutional investors. The increased accessibility of ML, combined with the rapid growth of the internet, has made it easier for retail investors to access vast amounts of financial data (James et al. 2013) and conduct their own analyses or develop ML models for market forecasting (Krauss et al. 2017).

As ML becomes increasingly integrated into the investment landscape, understanding these implications will be crucial for anticipating how the market may evolve and ensuring that retail investors can effectively leverage ML technology to achieve their financial goals.

1.2.Research Objectives

This paper aims to explore the potential short-term and long-term impacts of ML adoption on retail investors in the financial market. To delve into the effects of incorporating ML models within the retail finance market, the following three distinct objectives are developed:

The first objective focuses on examining the practicality of enabling an average individual or retail investor to forecast financial markets using a custom Python ML model. This model will concentrate on short-term markets, as they are less prone to significant fluctuations caused by real-world events and have more identifiable patterns and techniques employed by professional traders.

The second objective involves contrasting my findings with those from existing research (Fletcher 2012; Henrique et al. 2019; Vats and Samdani 2019) to evaluate my hypothesis about the possible influence of ML on financial markets. This objective adopts a more scientific approach, as it entails analysing the potential impact and theorising about methods that could be used to mitigate it or how the market may operate in the future.

Lastly, the third objective is to gauge the readiness of everyday individuals to invest in financial markets if they have access to ML tools that can help predict market trends to some degree and verify their investment ideas. This should, in theory, instil greater confidence in their decision-making. However, due to receiving the ethical approval relatively last minute, I did not have enough time to conduct the study. As a result, this section remains inconclusive, and further research is required to explore the potential impact of ML tools on the investment behaviour of everyday individuals.

1.3.Structure of the Report

This report has eight chapters, providing an understanding of the application of ML in retail finance and its impacts.

Chapter 1 serves as an introduction to ML outlining the basics of ML and the identified research objectives; followed by Chapter 2, which focuses on explaining the relationship between ML and retail investors, and the source of financial data used in this project.

Chapter 3 presents a literature review discussing the principles of how the stock market functions and the existing applications of ML for forecasting the finance market. This chapter also proposes hypotheses on the impacts of ML in the finance market, which could be researched further in the future.

Chapter 4 details the research approaches that have been used for this project, including the LSTM algorithm, other ML algorithms, the React web dashboard, NodeJS HAPI API, (Application Programming Interface) Docker and covering the S&P Capital IQ Pro platform.

Chapter 5 presents and discusses the forecasting results obtained from the application of my algorithm on the stock market in two sections; the first section is dedicated to forecasting the closing price utilising historical closing prices, while the second section is for intraday forecasting consisting of 15-minute intervals of the closing price.

Chapter 6 concludes the key findings, which offer insight to retail investors, future researchers and potential policymakers.

Chapter 7 details the future work that could be done to the algorithm itself to produce more accurate results and to make it more capable of long-term predictions. Also, specific research could be carried out in relation to the impact of ML itself and the sentiment of retail investors if they had access to tools like these. Lastly, Chapter 8 provides an overall reflection of this entire project, discussing the challenges faced and the lessons learnt from carrying out this project.

2. Background

2.1.Machine Learning and Retail Investors

This chapter aims at introducing some background information that is of relevance to this project, particularly it might assist audiences who may lack some specific domain knowledge to better understand this topic.

ML is a branch of artificial intelligence that makes it possible for a program to learn from input data and improve its performance over time without any explicit instructions given to it. In the context of finance, ML algorithms are generally used to analyse financial data of any kind to identify patterns and make predictions on the direction of the market to make more informed investment decisions. In a similar manner, it could be used for more security purposes, fraud detection for instance, which conducts risk assessments to analyse data like financial histories automatically in order to determine if it is safe to lend the person money.

2.2.Financial Data

A number of sources have been considered to obtain the financial data needed for this project, including databases from Cardiff University, Yahoo Finance, Xignite and IEXCloud. Among these options, only IEXCloud was chosen to be the data provider with the others deemed to be either inaccessible or unsuitable due to the following reasons:

The access to the financial data services available at Cardiff University is restricted to the Business school only, therefore, it is inaccessible for students from the School of Computer Science and Informatics.

Although Yahoo Finance provides historical and real-time data through an unofficial API, it was considered to be unsuitable due to ethical concerns associated with using this unofficial solution, as well as rate limitations. Whilst Xignite and IEXCloud are both commercial providers, it is of relatively high cost, requiring more than $15,000 USD to obtain the financial data needed from Xignite, compared to the cost of $49 USD a month with IEXCloud. It is worth noting that institutional investors typically have access to these more expensive commercial data providers and more advanced investment tools like Bloomberg Terminal, which already gives a more competitive edge. I did manage to get access to a commercial institutional data platform called S&P Capital IQ Pro by contacting S&P Capital directly for research, which delivers real-time data of companies and relevant news with alerts. However, it is important to note again that these types of tools normally would cost over $20,000 USD per year to have access to (Martel 2023).

The following Figure 13. Xignite Email shows the price provided by Xignite and that they do not provide any academic access. I had to anonymise it for the privacy of the employee.

IEXCloud, as a commercial data provider, offers a platform for subscribers to access a range of financial data, including historical, financial and real-time stock prices, which encompasses various stock market points such as open, high, low, close, and volume for the stocks being analysed. Data obtained is then utilised for training the LSTM model and making predictions on future stock prices.

IEXCloud changed its pricing model sometime around March 2023 based on the Internet Wayback Machine (Wayback Machine 2023). It used to be that you could get access to the basic financial fundamental data for $49 USD a month, but it has now recently changed that you have to pay for different forms of data separately per month, such as fundamental data being $300 USD a month and historical data being $200 USD a month.

This implies that IEXCloud may no longer be an appropriate choice for retail investors seeking to develop ML models due to the elevated costs associated with acquiring the necessary data. While it is possible to recoup these expenses through investments, this approach typically necessitates more starting capital and possibly riskier trading strategies, engaging in e.g. options contracts, futures, forex, et cetera. Trading Options contracts primarily involve puts and calls, in which investors speculate whether the value of equity will rise or fall to a specific level within a specific time interval, allowing them to buy or sell at a predetermined price and earn a premium (Chen 2023). However, such trading can be fraught with risk, as evidenced by the collapse or significant losses experienced by certain hedge funds during the GameStop phenomenon (Hart 2021).

3.Literature Review

3.1.Principles behind the operation of a Stock Market

There is no single formula that can fully describe and explain how a stock market functions in general, as it is a complex system that involves many different factors and variables (Malkiel 2008; Guida 2018). However, there are several key concepts and principles that are crucial in explaining the overall operation of a stock market, including general economic trends, the principle of supply and demand (S&D), the factor of market efficiency, and market psychology.

The overall performance of the stock market is closely tied to broader economic trends in relation to the growth in Gross Domestic Product (GDP), inflation, interest rates, regulations and geopolitical developments. Changes in these factors can impact the demand for stocks, which directly affects the stock prices, the principle behind such impact will be explained later on. A clear correlation can be found between certain stocks, commodities and geopolitical events. For example, the recent geopolitical situation in Ukraine has been affecting the stocks of military companies. For example, the stock price of Lockheed Martin Corp. has increased by an estimated 22% since the 24th of February 2022. A similar positive correlation can also be seen with other stocks that develop armaments, ammunition, and weapons technology (Klebnikov 2022).

The principle of supply and demand (S&D), which was referred to in the paragraph above, is identified as “the relationship between the quantity of a commodity that producers wish to sell at various prices and the quantity that consumers wish to buy” (Britannica 2023). Having applied this principle to the stock market, when there are more investors wanting to buy a particular stock than the number of shares available in the market, the price of that stock tends to rise. On the other hand, when the number of investors who want to sell a particular stock is more than the number of those who want to buy it, the price tends to fall. The principles of S&D in microeconomics are therefore of relevance to a stock (Cvitanić and Zapatero 2004).

Further, the factor of market efficiency comes into play. The stock market is considered to be an efficient market because stock prices generally reflect all available information about the company, such as financial performance, prospects and values to a certain degree (Malkiel 2008; Investopedia 2022). This means that it is difficult to consistently beat the market by using a single formula or strategy. It is also noteworthy that institutional investors have more information available as referenced earlier in this paper. These company-specific factors, such as earnings reports, product launches, accounting statements, management changes, and regulatory developments, indeed play a key role in stock price evaluation, having a significant influence on the price performance of individual stocks.

Other than being influenced by company-specific factors, the stock market can also be influenced by investor sentiments and market psychology, such as fear, greed, and optimism

(Shefrin 2007). Market psychology has become an increasingly important factor, especially within risk management. These emotions can drive fluctuations in stock prices that are not always rational or predictable, such as the GameStop Corp. phenomenon in January 2021 (Hasso et al. 2022).

It is safe to say that the almost endless amount of variables and other factors that play into the pricing of a financial asset makes predictive analysis an increasingly difficult process.

3.2.Application of Machine Learning in the Stock Market

It is evident that the stock market is a complex and dynamic system involving a wide range of factors that influence its behaviour. In recent years, there is a growing trend in relation to the application of ML being seen in the financial sector, and the retail investor market is no exception. In particular, ML algorithms have been used to develop predictive models for stock prices and identify profitable investment opportunities.

Several academic publications have explored the use of ML in predicting stock prices. For example, Jin et al. (2020) used deep neural networks to predict stock prices based on news articles and social media data combined with the LSTM algorithm. They found that their model outperformed traditional models, indicating the potential of ML in predicting stock prices. Similarly, Dang et al. (2021) used a hybrid model combining artificial neural networks and support vector regression to predict stock prices, achieving higher accuracy compared to traditional models. Combining multiple factors is wise, but when using sentiment as a factor, you have to remember that the accuracy can be affected by factors like sarcasm or people falsely advertising something due to cost fallacy for instance.

ML has also been used to develop investment and trading strategies. For instance, Krauss et al. (2017) compared the performance of deep neural networks, gradient-boosted trees, and random forests in predicting the S&P 500 index's daily return, concluding that these ML techniques achieved statistically significant positive returns and outperformed traditional linear models. Additionally, a paper by Bao et al. (2017) proposed a deep learning framework for predicting financial time series data, combining stacked autoencoders and LSTM networks. When applying to stock indices from various countries, their model demonstrated superior prediction accuracy compared to other benchmark models. These studies highlight the potential of ML in developing effective investment and trading strategies that can achieve higher returns than traditional methods.

3.3.Summary

In conclusion, the stock market is a system that is influenced by numerous factors, including general economic trends, S&D, market efficiency and market psychology. Due to the complexity, there is no single formula or strategy capable of perfectly predicting the market.

However, recent advancements with ML have shown promising results in outperforming traditional investment methods. By leveraging the existing ML techniques, like the LSTM algorithm and hybrid models, investors can better understand the stock market's behaviour and capitalise on it to make profitable investment decisions. Despite these promising results, it is crucial to remain vigilant when investing due to the limitations associated with ML, particularly market psychology which can have a significant impact on the market affecting the accuracy of predictions and the length of time needed for these ML models to adapt. Ultimately, it is believed that ML can positively impact market analysis. More radical hypothetical impacts that ML can introduce to the market will be discussed in the next chapter.

3.4.Hypothesis

Expanding upon the literature review on existing academic studies, we can delve into several radical hypothetical outcomes that might stem from advancements in ML and their potential impact on the relationship between retail and institutional investors. While these hypothetical scenarios may seem unlikely, examining their implications on the stock market and macroeconomic outcomes provides valuable insights.

A potential development in the realm of artificial intelligence (AI) and finance could be the creation of a comprehensive AI-driven predictive model that accurately forecasts stock prices by considering an extensive range of variables. This model would account for factors such as geopolitical events, corporate statements, and various aspects of human behaviour, with some limitations related to nuances like sarcasm. The establishment of such an advanced predictive tool could have significant implications for the financial markets and the balance between retail and institutional investors. In this scenario, the playing field for all market participants would be levelled, granting retail investors access to the same insights and predictive power as institutional investors. Consequently, the traditional advantage held by institutional investors could diminish, leading to a more democratised and competitive market. However, this might also result in an over-reliance on AI, which could increase market volatility and make the market more susceptible to manipulation or system failures. As a result, there would likely be more market regulation regarding market manipulation by having excessive order blocks, so it requires large purchases to move the price by a certain percentage at all.

Another unlikely outcome involves ML algorithms becoming so advanced that now both types of investors are on equal ground being rewarded solely based on company stock growth that relies on company fundamentals and other geopolitical events. In turn, this could lead to an increased focus on transparency, ethical investing, and the democratisation of financial services.

As ML algorithms continue to improve, the stock market may become increasingly efficient, creating a market efficiency paradox. In this hypothetical scenario, it could become nearly impossible for any investor to outperform the market consistently. This might lead to a decline in the role of active fund management, as investors shift towards passive investment strategies. Consequently, an over-concentration of investments in a small number of market indices or assets could emerge, ultimately increasing systemic risk and market volatility.

The widespread adoption of AI tools among retail and institutional investors could also lead to a self-reinforcing feedback loop that amplifies market trends. This might result in a series of AI-driven market bubbles and crashes, as investors follow algorithmic recommendations to buy or sell stocks without considering the broader economic context or fundamentals. These events could potentially destabilise the financial markets and thus the global economy; in turn, cause economic decline i.e., recession. Building upon such a theory would necessitate stronger regulatory oversight of AI applications in the financial sector, for instance, EU Parliament has already started regulating the use of AI (European Parliament 2021) and the US government has a department that does the same. (National Artificial Intelligence Initiative Office 2020)

Finally, institutional investors might engage in an AI arms race in a bid to maintain their competitive edge, continuously seeking to develop more advanced algorithms that outperform the competition. This could create a growing divide between institutional investors with access to cutting-edge technology and retail investors, further exacerbating existing inequalities in the financial sector.

By exploring these radical hypothetical scenarios, we can gain a deeper understanding of the potential risks and challenges that may arise and inform future research and policy discussions in the field.

In conclusion, the stock market is highly complex and involves numerous factors that influence the prices of equities sold on the market. We looked at the principles of S&D, market efficiency, economic trends, company-specific factors and the sentiment of the investor, which all can influence the price of a stock. ML continues to advance over time and its application in the financial sector could potentially revolutionise how we predict and trade these equities, levelling the playing field between retail and institutional investors. However, these advancements in ML come with their own potential risks such as increased market volatility and relying too much on predictions. As we examine the hypothetical scenarios, we can better understand the potential implications as ML advances, informing future research and regulation to address these types of risks, and making a more transparent market for everybody.

4.Approach

4.1.Introduction

In this project for my main objective, I developed a custom LSTM algorithm using Python and NumPy to predict future stock prices based on historical data. The model was trained on historical time series data primarily but is adapted to work with real-time data to forecast intraday prices. To assess its practicality for retail investors, the model focused on short-term predictions less affected by real-world events that could skew accuracy. A web dashboard was created using React to visualise the financial data and the predictions produced by the LSTM algorithm. I created a NodeJS API with the HAPI framework to send the data between the algorithm and the dashboard. Also, user management was handled by the same API, employing bcrypt hashing for password security and Prisma for database ORM, (Object-Relational Mapping) ensuring compatibility with various databases.

My literature review in this project provides a compartmentalised version of past academic literature with my own findings. This has been critical in developing the hypothesis seen in this project on how ML could impact the retail finance market as it gets adopted more over time due to technology advancing and becoming more sustainable for the everyday person.

4.2.LSTM Algorithm

4.2.1. Introduction to LSTM and its applications

After reading through various literature to see what other people have used to predict future values, I noticed that the LSTM algorithm was brought up multiple times, such as papers that were looking into how to predict household electric power consumption (Cascone et al. 2023) or predict CO2 concentration in classrooms longer-term (Yang et al. 2022).

4.2.2. Advantages of the LSTM model for financial forecasting

I discovered that a particular model is well-documented and advantageous for my research. This model does not necessitate manual data labelling, enabling the use of larger datasets, such as time series with years of historical data. Unlike traditional Recurrent Neural Networks (RNNs), it circumvents the gradient vanishing problem (Pascanu et al. 2013), allowing for more accurate predictions. The model can effectively capture the impact of past events, including social sentiment, interest rates, inflation, product performance, and other factors, on future stock prices. It also accounts for recurring events, such as quarterly earnings, by recognising patterns and extrapolating potential future outcomes. By employing this model, I can conduct a comprehensive analysis of the various factors influencing stock prices, both positively and negatively. This approach ensures that the algorithm can discern the relationships between past and future events, as well as identify underlying patterns and trends. Consequently, this model allows for more informed decision-making and enhances the accuracy of financial forecasts.

Considering the benefits offered by this model, I am confident that its adoption will be instrumental in advancing my research objectives. Furthermore, the ease of handling larger datasets will enable a more extensive exploration of time series data, thereby providing a deeper understanding of the intricate dynamics at play within the financial markets. Ultimately, this model serves as a valuable tool in the pursuit of quantitative finance and risk management, promoting more effective investment strategies and market insights.

4.2.3. Uniqueness of my LSTM algorithm

The following explains how the modifications to the LSTM algorithm allow the implementation designed as part of this project to become different from the existing ones:

- **Optimisation**: Unlike conventional LSTM that relies typically on the Adam optimiser or other form of optimisers to improve gradient descent, my algorithm instead employs Xavier's Initialization to optimise gradient flow.
- **Compactness**: My LSTM model focuses purely on necessary features for financial market prediction, eliminating the bloat that other ML libraries would have. In the end, if you only have a single motive, all the other unnecessary algorithms or code can be removed as they would instead slow down the process.
- **Timestep Solution**: My algorithm has a timestep solution that allows predicting several days ahead, which is something that other algorithms may lack. Also, this works as a backup solution if there are more significant recent price changes that could affect the accuracy of the algorithm due to taking time to adapt. Instead, you can compare the results of single-time steps and multi-time steps.
- **Performance Metrics**: My model calculates performance metrics at the end of the prediction, which focuses more on the overall accuracy, rather than per-epoch.
- **Visual Accuracy**: As I utilise the matplotlib library, my algorithm can provide a visual representation of prediction accuracy, which is absent in many other algorithms. This allows me to fine-tune the model to achieve better accuracy by reading the graph, rather than comparing raw numbers all the time.

4.2.4. The configuration of the LSTM algorithm

The input size was configured to be 2, the output size was 1, and the learning rate was established at 0.4. These parameters yielded the most plausible outcomes during the experimentation phase. It is crucial to carefully calibrate the learning rate, as excessively high or low values can dramatically impact the results by influencing the weight values, thereby causing an increase in the loss function instead of a reduction (Brownlee 2019; Google 2022).

Parameter	Value
Input size	2

Output size	1
Learning rate	0.4

Table 1. LSTM Class Parameters

Additional parameters for the LSTM are shown below to configure the stock selection, epochs, overriding training percentage and other needs:

Parameters	Value
stock_symbol	AAPL
epochs	1000
training_percentage	0.8
plot_days	600
save_model	False
live_data	False
send_data_to_api	False

Table 2. Additional LSTM Parameters

The *'training_percentage'* parameter facilitates a more precise division of data between training and testing subsets. It is customary to allocate a larger proportion of data to training, typically around 80%, while the remaining 20% is reserved for testing in order to evaluate the efficacy of the algorithm.

The weights of the algorithm can be saved by toggling the save_model parameter to *True*. It creates a NumPy .npz file that contains the weights using the NumPy "savez" method.

The Python library "requests" is utilised for making POST requests and transmitting the prediction data to the API. Subsequently, Prisma is employed to store the data in a designated table. To activate this functionality, the parameter *'send_data_to_api'* must be set to *'True'*.

4.2.5. GPU Support and its limitations

By adding GPU (Graphics Processing Unit) support to the algorithm using the cuda and cupy library; this could not be tested properly at the time of implementation, as there was no support for Python 3.11. One of the drawbacks of cuda is that it only supports GPUs that have CUDA cores, which is limited currently to the Nvidia GPU line-up (Nvidia 2020).

4.2.6. Evaluation metrics for loss

In order to assess the algorithm's loss, I employed a variety of mathematical techniques, including MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and R2 (R Squared). While MSE is typically utilised for this type of algorithm, I have provided alternative metrics for a more comprehensive evaluation. A picture of the implementation of how I implemented the methods to calculate the loss can be seen in Figure 8. Methods to calculate machine learning loss. The following Figure 17. My LSTM evaluation metrics show us the overall metrics produced for the final prediction. Traditionally, you would calculate the loss for each epoch, but I decided to instead calculate it for the overall result.

4.2.7. Visualising predictions and accuracy

The matplotlib library was used to plot the predictions of the historic data. This was used to see if the algorithm was working as hypothesised and to measure the accuracy when fine-tuning the parameters. See Figure 5. AAPL Matplotlib LSTM Prediction for an illustration of the graph generated following a prediction. The displayed accuracy percentage pertains to the test data set.

4.2.8. The implementation of the algorithm and modifications

My implementation follows the standard LSTM algorithm formula but uses Xavier's Initialization to set the initial values for the weight matrices for the input, forget and output gates, candidate values and the output layer, this is done to help maintain a balanced flow of gradients during the training process if working with larger datasets. I created a diagram for the LSTM algorithm to better visualise how it works, which can be seen in the following figure: Figure 1. LSTM Diagram. Additionally, this is how I implemented Xavier's Initialization method as seen in Figure 2. Xavier Initialization. Now let me explain how the algorithm works:

Input gate:

We compute the input modulation gate, which can be denoted as i_t.
$i\_t = SIGMOID(W\_i * [XT, HT - 1] + b\_i)$, where W_i and b_i are the weight matrix and bias for the input gate.

Candidate value:

We compute the candidate value, which can be denoted as Ĉ.
$\hat{C} = TANH(W\_c * [XT, HT - 1] + b\_c)$, where W_c and b_c are the weight matrix and bias for the candidate value.

Forget gate:

We compute the forget gate, which can be denoted as f_t.

$f\_t = SIGMOID(W\_f * [XT, HT - 1] + b\_f)$, where W_f and b_f are the weight matrix and bias for the forget gate.

Output gate:

We compute the output gate, which can be denoted as o_t.
$o\_t = SIGMOID(W\_o * [XT, HT - 1] + b\_o)$, where W_o and b_o are the weight matrix and bias for the output gate.

Cell state updates:

We now update the cell state, which can be denoted as c_t.
$c\_t = f\_t * c\_t - 1 + i\_t * \hat{C}$, where f_t is the forget gate output, i_t is the input gate output and Ĉ is the candidate value.

We can now compute the output values by doing $HT = o\_t * TANH(c\_t)$, where o_t is the output of the output gate and c_t is the updated cell state.

4.3.Other Algorithms

4.3.1. Comparing machine learning algorithms and libraries

I employed ML libraries in Python, such as Keras, and Brain.JS in Node.JS to compare their performance with my solution. Additionally, I sought to examine the performance of other algorithms, like GRU (Gated Recurrent Unit Network) and the original RNN, in relation to LSTM.

My choice of Keras for Python implementation was influenced by its widespread popularity among prominent companies, such as Google, which is evident from testimonials on their homepage. Furthermore, Keras utilises TensorFlow, a library developed by the Google Brain Team, as its underlying framework.

4.3.2. Performance considerations between Node.JS and Python

I came across Brain.JS through YouTube and, upon experimentation, found it to be highly flexible in terms of data inputs, capable of accommodating multiple raw numerical data points simultaneously. Moreover, it prioritises GPU usage, with the ability to fall back on CPU (Central Processing Unit) support if a compatible GPU is unavailable. This, in turn, accelerates the training process on machines with compatible GPUs. TensorFlow also offers GPU support, and Node.JS generally exhibits superior performance compared to Python. (Wong 2019; Dawid and Joanna 2020)

4.3.3. Ensuring fair evaluation across algorithms

I maintained an equal number of iterations or epochs for my custom algorithm and the comparative algorithms to ensure a fair evaluation. Additionally, I aimed to keep a consistent learning rate, specifically between 0.4 and 0.5, across all algorithms, with 1000 epochs. This approach was adopted to achieve results that were as realistic and reliable as possible. Although increasing the number of epochs further could improve the results, the gains at this stage would be marginal. Utilising fewer epochs is more suitable for computers with limited hardware capabilities. For context, the equipment used for training the algorithm was equipped with a Ryzen 7 4700u processor, which can be considered a high-end laptop, however not all retail investors would have access to this and thus exceeding 1000 epochs would likely yield unrealistic expectations.

4.4.React Web Dashboard

4.4.1. Choosing the React Framework

I went with the React web framework due to my own past positive experiences with it, both personal and professional experience. It is the second most-used web framework on the Internet in 2022 (Statista 2022). This has facilitated access to information and provided robust development support, as the technology is developed by Meta, formerly known as Facebook. However, I opted for "React Vite" instead of the standard version of React, as it offers faster building, custom plugin support, and focuses on native ESM (ECMAScript) support.

4.4.2. React's virtual dom and state management

React uses virtual DOMs (Document Object Models) to handle state management, as in updating website components if there is new data for instance or a button is pressed that would trigger an event. The useEffect is an inbuilt method that is used to update components in real-time based on a condition or when a page is opened the first time.

For instance, in my project, I employed it to fetch data from my API and store the retrieved stock data in a local array. When changing the stock name using the selector on the website, it fetched the data for the selected stock and updated the information on the website without requiring a page refresh. Moreover, I utilised useEffect to manage authentication checks by validating the cookie upon page refresh and to handle pagination for my data grid, which displayed historical stock data. A picture of the data grid can be found in Figure 12. MUI Datagrid displaying historic data.

4.4.3. React router for path management

The React Router module was employed to manage routing for different website paths, such as /login, /register, and /dashboard. Without this module, there would be only a single page containing all functionality, leading to a significant amount of code being rendered when visiting the page. By using React Router, I could specify individual page files for each distinct web path and perform the authentication check method specifically for the dashboard page. This approach enabled redirection of users to the login page if they lacked the necessary permissions.

4.4.4. Utilising the MUI material design library

I opted to utilise the MUI material design library for creating website components, allowing for a more consistent design. This choice also saved time, as I could avoid crafting entire components from scratch, which might not be as responsive and could require additional effort. MUI components are built using flexboxes, simplifying the process of connecting components or inserting data into them. Furthermore, MUI is a mobile-first library, eliminating the need to manipulate media CSS (Cascading Style Sheets) rules to ensure website compatibility with mobile devices. MUI offers native methods for overriding component designs, such as the 'sx' parameter or the 'styled' method, enabling the creation of modular, object-oriented components.

4.4.5. Type Safety of TypeScript

Additionally, as Vite was employed, TypeScript was used instead of JavaScript. TypeScript enables type-safe coding, requiring the specification of data types for elements such as variables. This approach ensures proper validation when data types are utilised by various methods, reducing the likelihood of simple bugs and eliminating the need to run code to test for syntax errors. An integrated TypeScript code server is available within an IDE (Integrated Development Environment) or code editor through an extension that reads code in real-time and provides corrections.

4.4.6. Native environment variable support in React

React natively supports environment variables, eliminating the need for a third-party library to use a .env file when handling sensitive information such as API keys on the frontend. This also simplifies the process of updating the API URL (Uniform Resource Locator) address if the API is moved to a different server or domain.

4.5. NodeJS HAPI API

4.5.1. Choosing the HAPI API Framework

In this project, I chose to utilise the HAPI framework for my API, as I had prior experience with Express, a different REST API. Exploring alternative solutions rather than relying on familiar technologies would offer a more comprehensive learning experience. Upon researching various API frameworks, I discovered HAPI and observed that it was employed by several well-known companies, such as Beats and Brave, as evidenced by the testimonials.

This indicated that HAPI could be a viable and effective option for my project. I made further research into its performance and discovered that it is more efficient based on online benchmarks compared to the competitors. (Choubey 2023)

4.5.2. HAPI's internal data handling and its security features

Throughout the time I was using the framework, I found that it can handle a lot of different data internally, rather than requiring a 3rd party middleware, such as session management which is something I managed to implement natively using their cookie-session plugin. By default, this has extra security features like cookie encryption using a password you define, (Figure 7. API Cookie Middleware) so I could use that to secure my different API routes from malicious users and make the dashboard less prone to session hijacking. As it is possible for a person to guess a cookie otherwise or modify a cookie through the inbuilt browser tools.

4.5.3. The API integration with the LSTM algorithm and IEXCloud

This API worked in the middle of the project receiving data from the LSTM algorithm like stock price predictions using IEXCloud's own library to interact with their API. I got the data for the current day and historical data, then stored them in an SQLite database using the Prisma ORM (Object-Relational Mapping) software. Figure 10. Fetching historical data from the API shows us how the data is fetched and saved. The data then is received by the dashboard by fetching the API using the fetch library that is inbuilt to later versions of Node.JS, which uses the REST (Representational state transfer) architecture design. The API then fetches the data from the database and sends it over making it easier to document predictions for each day.

The historical data that was used for the training of the LSTM algorithm was fetched using this API, which fetched the data from IEXCloud and saved it to the database in a similar manner. This result would be more cost-efficient, as each time a request is made to IEXCloud, it uses its own "credit" system that is limited and subscription-based.

4.5.4. Prisma ORM for database management

Prisma has a schema file that allows us to define the tables/models and the appropriate column with the specific data types, character limits, relationships and other needs. Also, it has versioning support, which means you can always go back to past changes if you wish to or see how the database was structured in the past if you need the information. This is a more secure approach as well to handle your database structure as it is clearly defined in the file and you do not need access to the database to work with it, instead, a dedicated DevOps person for instance would push your database changes. A snippet of my Prisma schema can be seen in Figure 6. Prisma Schema, which shows how easy it is to visualise the database without accessing it.

4.5.5. Email verification, Prisma and Nodemailer

I managed to implement email verification support by integrating both Prisma and Nodemailer modules. Nodemailer would handle the SMTP (Simple Mail Transfer Protocol) request and send over the verification link to the user, while Prisma would save the verification token and verify the user if the person opened the link. The token itself was generated randomly using the Node.JS Math library generating random characters and the database had a separate column for the expiration date, which was compared to the current date. If the token has expired, then a new one would be sent to the email if accessed a link with the older token. As it is a token used purely for verifying the account, it is not cryptographically secure, but the crypto library could be used to do that instead. You would preferably instead have captchas like hCaptcha or CloudFlare Turnstile on your frontend to prevent spam bots, which is something that many websites do instead for bot detection.

4.5.6. TypeScript safety in the API

The API in a similar manner was using TypeScript, just like the web frontend. Everything is type-safe and wrong data types cannot be inputted, such as an e-mail cannot be a number on the input or on the database side.

4.5.7. Third-party environment variable support for security

Environment support was added to the API using a third-party library called dotenv, as HAPI by default does not provide it. This is used to store the IEXCloud API token securely.

4.5.8. SSE (Server-sent events) for real-time data fetching

Additionally, the real-time data was fetched using this same API. This was done by using the SSE (Server-sent events) technology to open a continuous HTTP (Hypertext Transfer Protocol) connection to IEXCloud's API.

4.6.Docker

4.6.1. Designing to be microservice first and docker containers

As I designed my project software to be microservices, this allows me to run each program individually from each other in separate virtualised environments, known as containers in Docker. This makes it easier to deploy the programs like the API and the web dashboard in particular, as all the user is required to do is run the docker-compose file to set it all up. This makes it easy to run the software in a minimal environment that is less prone to bugs. Having a dedicated small operating system to run the software without any 3rd party packages could potentially conflict. Docker makes an isolated network for each container, in the worst-case scenario if one of the programs has a vulnerability like RCE (Remote Code Execution) due to bad code or if one of the dependencies allocates for a supply chain attack. Then the isolated environment prevents further damage from being done to the system or breaching other databases for instance that could be hosted on the same machine. This is how the docker containers look at the program Docker desktop after deployment Figure 9. Docker containers.

4.6.2. Load balancing and orchestration

Due to the Docker approach, it allows for load balancing between servers and employing solutions like Kubernetes for orchestration, which leads to better uptime/stability by distributing the API to multiple cloud services. (IBM 2023)

4.7.S&P Capital IQ Pro

In the preliminary process of studying the various subjects for my report, I found S&P's Capital IQ Pro platform to be of great value to this project as it could provide me with both due diligence in regard to market intelligence, but also in terms of realistic accessibility to professional investor intelligence. Shortly thereafter, I sent an email to S&P Global's market intelligence team, requesting access to the platform. The primary reason for requesting access to Capital IQ Pro is due to its comprehensive financial data insights and market intelligence. It provides users with in-depth financial data, analytics, research, and market tools that are useful for investment professionals, analysts, and researchers. The platform is not democratised for everyday retail investors as it is estimated to have a subscription-based fee of around $30.000 USD thus primarily utilised by professional investors (John 2023). Capital IQ Pro offers access to both public and private enterprise data, financial statements, mergers & acquisitions (M&A) transactions, initial price offerings (IPOs), fixed income data, as well as a diverse set of various industry-specific data sets. This amount of data accessible is of relevance for risk management and investment strategies.

4.8.Synopsis

To achieve the main objective of my project, I developed my own (LSTM) in Python from scratch using the NumPy library to perform the mathematical operations needed. LSTM is a type of recurrent neural network that is more suitable for sequential data like the time series data I was using for it. The algorithm was trained on historical stock price data to learn patterns and trends to make predictions about future stock prices, but it has compatibility to work with real-time data to make predictions of the current day like the next hour for instance.

To evaluate if the model is practical for a retail investor to forecast future financial market values, I primarily used it to predict the market short-term, more specifically intraday levels on low-end volatile stocks with a high market capitalisation. The reason is that it is less prone to significant fluctuations caused by real-world events that could drive the accuracy of the prediction. Events that can cause fluctuation can be social media influences such as social sentiment, geopolitics or the quarterly earnings of a company for instance. The stock that was forecasted was Apple (NASDAQ: AAPL) due to its Beta being 1.30 (5-month value); which measures the volatility, or systematic risk, of financial security, as it compares to the broader market. (Mirzayev 2021) 1.30 refers to the stock volatility being 30 per cent more fluctuant than its corresponding stock exchange. In comparison; Tesla (NASDAQ: TSLA) has a Beta of 2.01 meaning volatility of 101% compared to the NASDAQ exchange. These volatility levels, which do not solely correspond to its traded volume, make predictions and forecasting for ML a lot harder due to the unpredictability of its "sudden" movements. There are thousands of stocks to choose from, the importance for our test is its stability and accuracy which prevails under certain financial conditions. Special Purpose Acquisition Companies (SPACs) were also not accounted for in our test due to their fundamental nature, though this provides an interesting hypothesis for future work.

I visualised the predictions generated by the LSTM algorithm by creating a web dashboard using the React framework and using Matplotlib, which is a Python library to graph data and in this case; used to show how the predictions compare to the historical data. The dashboard displays historical data for a given stock and the predicted values generated by the LSTM algorithm alongside it, making it easy to make your own judgement on it and compare. The data comes from an API I have created myself in NodeJS, that uses the HAPI framework to create the routing.

The user management of the dashboard is handled by the HAPI API, so the passwords are hashed and salted on the server side using the bcrypt hashing function and the details then are saved to an SQLite database using Prisma, which is a database ORM (Object-Relational Mapping). Therefore, all the queries made using Prisma and the schema are all cross-compatible with other databases like MySQL, MongoDB and PostgreSQL. Prisma has recently demonstrated improved compatibility with serverless architecture, making it increasingly suitable for microservice-first services (Adams et al. 2023). This enhancement in Prisma's performance aligns well with the growing trend towards adopting serverless computing solutions, which offer better scalability, cost-efficiency, and ease of management for modern applications. Consequently, this development further strengthens Prisma's position as a valuable tool for developers working with complex data manipulation tasks within a serverless environment.
This leads to my system appearing as follows: Figure 11. System Diagram.

5.Results

5.1.Introduction

This section will cover the results produced by my own LSTM algorithm that I created in Python using the NumPy library. The results will be compared to the actual closing price of the stock. Other algorithms will be tested as well such as the original RNN algorithm that the LSTM algorithm is based upon. GRU and LSTM algorithms from other modules like Keras and Brain.JS will also be showcased to provide a generalised overview for comparison.

Brain.JS additionally uses five data points, which are the closing price, high price, low price, open price and daily volume. The other algorithms/software only use the closing price of each day.

5.2.Intraday

Intraday is defined as "within the day" hence its name. We are therefore forecasting the closing stock price of Apple (NASDAQ: AAPL) for 23/03/2023 using historical data from 28/02/2022 till 22/03/2023. This is how the historical data of AAPL would look after it has been fetched Figure 3. Historical IEXCloud Data.

Type	Result	Accuracy of % of actual based on delta	Delta of actual
Actual close price	158.93		
My LSTM algorithm	158.42	99.68%	-0.51
Brain.JS LSTM	159.67	99.54%	0.49
Keras LSTM	159.56	99.60%	0.63
Keras RNN	164.21	96.73%	5.28
Keras GRU	162.10	98.03%	3.17

Table 3. Intraday AAPL 23/03/2023

I must point out that the closing price on 31/11/2022 was 147.61, and on 29/11/2022 it was 140.77, which leads to a significant gap in results compared to the actual price on 01/12/2022. However, if I examine my timestep predictions, they respectively produced 149.41, 150.47, and 148.23, which are closer to the actual closing price. This serves as an effective method to double-check and evaluate the reliability of the result. Instead, one can average out the three numbers and examine recent historical data to determine if the prediction is plausible. Here is an illustration of how the predictions are made: Figure 14. LSTM Predictions.

Type	Result	Accuracy of % of actual based on delta	Delta of actual
Actual close price	148.08		
My LSTM algorithm	141.96	95.78%	6.12
My LSTM algorithm timestep	149.41	99.11%	1.33
Brain.JS LSTM	148.79	99.52%	0.63
Keras LSTM	151.29	97.86%	3.21
Keras RNN	162.99	90.41%	14.91
Keras GRU	162.83	90.51%	14.75

Table 4. Intraday AAPL 01/12/2022

Type	Result	Accuracy of % of actual based on delta	Delta of actual
Actual close price	155.46		
My LSTM algorithm	155.58	99.92%	0.12
Brain.JS LSTM	156.27	99.48%	0.81
Keras LSTM	157.22	98.87%	1.24
Keras RNN	162.25	95.73%	3.79
Keras GRU	162.29	95.70%	3.83

Table 5. Intraday AAPL 07/09/2022

I will have to note that I have not worked as much before with Keras, which means some of the Dropout rates and other parameters could be off causing the results to not be entirely correct. Also, Brain.JS handles the weights differently, so each run could sometimes produce vastly different results, so an average is taken based on three tests. Additionally, the training data was reduced for each test as it went from testing 2023 to testing 2022 with fewer data. Brain.JS prediction is shown in the following Figure 15. Brain.JS LSTM Prediction. Also, Keras looks as follows in Figure 16. Keras LSTM Prediction, as you can see with Keras that it typically produces multiple results and would have to average out the result. There might be a better way to handle it with Keras, but that is what Keras produces for me.

5.3.Real-time (15-minute intervals)

IEXCloud has the option to fetch real-time stock prices in 15-minute intervals and it would be possible to get more immediate data. The standard interval to receive data for retail investors is also 15-minute intervals on stocks (e.g., Yahoo Finance, Bloomberg, Google Finance). Though receiving more immediate stock prices requires signing a vendor agreement with e.g., Nasdaq and a Data Feed request with UTP (Unlisted Trading Privileges). (IEXCloud 2022) We are forecasting the closing price of Apple (NASDAQ: AAPL) for 02/03/2023 using real-time 15-minute interval data. The data collected were somewhat at random intervals due to the strain of having to collect and note down the results every day. There would be packet loss or network problems at times, causing the data that is received to be corrupted. This resulted in random intervals of data collection spanning 2-3 weeks of data prior to 02/03/2023. The last data collected was 30 minutes before the Nasdaq market closed on 02/03/2023. Here is a snippet to show the format of the real-time data that is sent by IEXCloud Figure 4. Real-time IEXCloud Data.

Type	Result	% Accuracy of actual based on delta	Delta of actual
Actual close price	145.91		
My LSTM algorithm	145.82	99.94%	0.09
Brain.JS LSTM	145.73	99.88%	0.18

Table 6. Real-time AAPL 02/03/2023

Unfortunately, I could not test the other algorithms using Keras, as they would require reprogramming to handle the real-time data, while Brain.JS could handle the raw data.

6.Conclusion

In conclusion, the report has successfully demonstrated the development, implementation, and evaluation of a deep learning algorithm based on LSTM networks for predicting short-term intraday stock prices. I designed the system to be microservice-first, incorporating Docker containers for improved deployment and security. Additionally, load balancing and orchestration possibilities have been explored due to the potential in enhancing stability and uptime.

Instead of employing the S&P Capital IQ Pro platform, I analysed how it works as a platform from an institutional investor's perspective, providing insights into the professional investor domain. The LSTM model was primarily tested on low-end volatile stocks with high market capitalisation, such as Apple (NASDAQ: AAPL), to minimise the impact of real-world events that could affect prediction accuracy. A web dashboard using the React framework and Matplotlib has been created and utilised to visualise the predictions generated by the LSTM algorithm.

The literature review provided a comprehensive understanding of the stock market's complexities, including supply and demand, market efficiency, general economic trends, company-specific factors, and investor sentiment. The review also explored the application of ML, particularly LSTM networks, in predicting stock prices and identifying profitable investment opportunities. ML advancements reveal their potential in revolutionising the financial sector and altering the playing field between retail and institutional investors. However, these advancements come with potential risks such as increased market volatility and overreliance on predictions.

After reviewing existing literature, I delved into several radical hypothetical outcomes that might stem from advancements in ML in finance, which could impact the relationship between retail and institutional investors. These hypothetical scenarios included an AI-driven predictive model that accurately forecasts stock prices, a market efficiency paradox, an AI arms race, and self-reinforcing feedback loops that amplify market trends. By exploring these scenarios, we can gain a deeper understanding of the potential risks and challenges that may arise and inform future research and policy discussions in the field.

My LSTM algorithm consistently produced high accuracy levels: 99.68%, 95.78%, and 99.92% for the three test scenarios. Brain.JS LSTM models were also fairly accurate, with 99.54%, 99.52%, and 99.48% accuracy. Keras LSTM models showed slightly lower accuracy but were still competitive. Keras RNN and GRU models demonstrated lower accuracy compared to the LSTM models. The high accuracy of the LSTM models is due to the nature of a single-time step prediction, so the accuracy would reduce when intending to predict prices e.g., one week in advance. This model would be a suitable tool particularly for intra-day traders in markets like forex (Foreign Exchange Markets) or with financial securities such as futures due to its high-frequency short-term trading.

Despite some limitations and challenges identified, such as handling real-time data and occasional network problems, my project realises the potential of LSTM networks in predicting short-term intraday stock prices. With continuous advancements in ML, the financial sector could undergo significant changes, making it crucial for future research and regulation to address potential risks and challenges that may arise as a result. The insights derived from the literature review highlight the need for vigilance when investing and emphasise the importance of continued exploration into ML's impact on the stock market.

7.Future work

Considering there is a lack of literature in relation to the average retail investor, it becomes apparent that further fundamental assessments could be conducted towards this market segment. ML for retail investing has been rather rigid and inelastic with most of the ML tools being utilised by institutional investors. The primary usage has been conducted utilising algorithmic trading where trades are executed automatically by following certain trading rulesets, i.e., indicators, sell-at-loss, sell-at-profit, et cetera. An increase in the availability of ML tools has been causing a drift towards more accessibility to these tools, resulting in their utilisation in the retail investor market. With the ongoing boom of technological advancements in AI in general, there will be more variables that can be accounted for by ML tools leading to a power dynamic shift between retail investors and institutional investors.

The current method used for predicting market trends focuses on a limited set of data points, mainly related to daily closing and opening prices. In future research, it would be beneficial to incorporate a wider range of data points, such as sentiment analysis, alternative training methods like deep reinforcement learning for correcting inaccurate predictions, and company filings to get an up-to-date assessment of the company's status. By using a more diverse set of data, including non-numerical information, the model can provide deeper insights. While the existing approach is effective for short-term predictions with relatively consistent patterns, it struggles with longer-term predictions due to the limited variety of data sources.

Further development work can be done on the web dashboard by adding more data points and other forms of metrics such as sentimental analysis to make better investment decisions. This would allow retail investors to see the bigger picture of a financial security and to gain more profits in terms of risk management.

Further research could be directed towards developing a model that adapts to black swan events in unpredictable situations, such as the COVID-19 pandemic, economic downturns like recessions, or geopolitical conflicts such as the Ukraine war. This approach could be advantageous in enhancing risk management for both industry professionals and retail investors, ultimately aiming to prevent or mitigate financial crises, rather than focusing solely on profit maximisation. This could potentially be approached by training for longer-term predictions and using data from influential events to see the specific estimated impact it has on the market as a whole and its correlating financial securities. Another example of different data points would be the evaluation of SPACs that serve a singular purpose which is to merge or acquire a private company to then list it on its corresponding market exchange. It would therefore provide an interesting challenge for AI to evaluate SPACs given their lack of historical data. However, AI can identify potential red flags or areas of concern in SPACs' financials, management, or industry, helping investors to avoid high-risk investments. It would also minimise human biases in evaluating the potential success of a SPAC, leading to more objective assessments. Such evaluations would for example take points on data stemming from the Security Exchange Commission (SEC).

An indispensable aspect to consider in future research is the long-term and short-term impacts of government regulation on the deployment and usage of ML tools in the retail finance market. Effective regulation is crucial for ensuring ethical practices, averting technological misuse, and preserving market fairness. On one hand, well-structured regulation can encourage the responsible employment of ML tools, safeguard retail investors from potential risks, and bolster market transparency. On the other hand, excessively stringent or inadequately executed regulations might impede innovation and decelerate the adoption of advantageous technologies. Striking a balance between the necessity for regulation and fostering innovation poses a critical challenge for governments and policymakers in the future. The future research undertaken on this subject would be particularly beneficial to policymakers at the Government or institutional investors who utilise ML tools in the financial market.

To summarise this section, future work in the realm of ML applications for retail financial markets should concentrate on broadening the diversity of data utilised in models, enhancing adaptability to unpredictable events, and pursuing a measured approach to regulation. As technological advancements persist, it is imperative for researchers and practitioners to delve deeper into these areas, ultimately empowering retail investors and cultivating a more equitable financial environment.

8.Reflection

While preparing this report, I have gained valuable insights into the application of ML algorithms for the retail financial markets and recognised that there are certain areas where improvements could be made to deepen my understanding.

Firstly, the approach for data collection and preprocessing could be more systematic and consistent. Ensuring data combined with non-systematic data points are gathered at regular intervals and from market opening would have provided more reliable results. As of preparing this report, companies are releasing their quarterly earnings which causes fluctuations. These fluctuations can be both positive and negative for a company and its stock. Additionally, more attention should have been given when preprocessing the data, which would have allowed for a smoother integration of various data sources.

Secondly, my analysis could have benefited from considering a broader range of ML algorithms and techniques. Whilst the LSTM algorithm was proved to be effective, exploring other algorithms, the Transformers model, for example, might have resulted in more accurate and robust predictions for time-series forecasts. Comparing these alternatives would enable a more comprehensive understanding of their strengths and weaknesses.

Thirdly, it is acknowledged that my research would have been more comprehensive if a wider range of data points, such as sentiment analysis, alternative training methods, and company filings, have been incorporated. As mentioned earlier in this section, it is currently “earnings season” ergo it would be interesting to see how my algorithm would analyse previous quarterly earnings assessments which happen four times a year thus four times in our historical data points that span a year. By using a more diverse set of data, including non-numerical information, the ML models could have provided deeper insights and improved the accuracy of longer-term predictions.

It is worth noting that ethical approval was sought from the university's ethics committee. The process of applying and obtaining approval could have commenced earlier in the project. Upon reflection, the Easter break should have been accounted for because the committee would not have been able to review the submitted application, which inevitably led to a delay. Consequently, ethical approval was received only about a week prior to submission, resulting in the incompleteness of the third objective (Hooper 2023). The proposed study under this objective would have been beneficial for the project itself and from a research perspective, given the recent innovative upswing in AI/ML.

In conclusion, reflecting on my project allows me to identify areas for improvement and further exploration. These insights will undoubtedly be valuable as I continue to develop my skills and knowledge in the fields of computer science and financial technology. As I move forward, I will strive to apply these lessons to future research endeavours, ultimately contributing to the development of more advanced and equitable ML applications in the financial sector.

Appendix

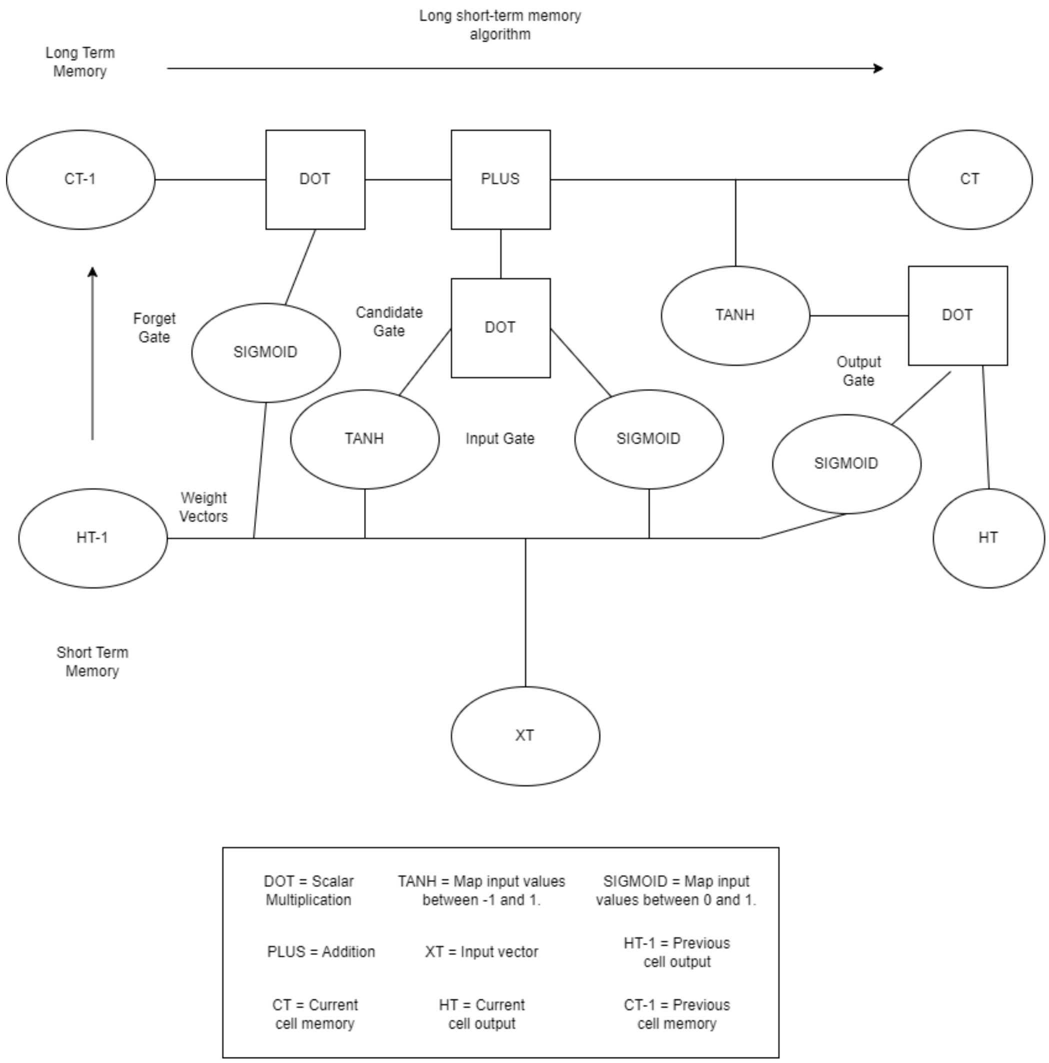


Figure 1. LSTM Diagram

```
# Xavier's initialization for weights
xavier_stddev: float = np.sqrt(2.0 / (input_size + lstm_cell_size))
self.forget_weights = np.random.randn(input_size,
                                      lstm_cell_size).T * xavier_stddev  # Weight matrix for forget gate
self.input_weights = np.random.randn(input_size,
                                     lstm_cell_size).T * xavier_stddev  # Weight matrix for input gate
self.output_weights = np.random.randn(input_size,
                                      lstm_cell_size).T * xavier_stddev  # Weight matrix for output gate
self.candidate_weights = np.random.randn(input_size,
                                         lstm_cell_size).T * xavier_stddev  # Weight matrix for candidate valu
self.output_hidden_weights = np.random.randn(lstm_cell_size,
                                             output_size).T * xavier_stddev  # Weight matrix for output layer
```

Figure 2. Xavier Initialization

```
You, 2 days ago | 1 author (You)
[        You, 2 months ago • Include the
  {
    "close": 160.25,
    "fclose": 160.25,
    "fhigh": 160.34,
    "flow": 157.85,
    "fopen": 158.86,
    "fvolume": 59256343,
    "high": 160.34,
    "low": 157.85,
    "open": 158.86,
    "priceDate": "2023-03-24",
    "symbol": "AAPL",
    "uclose": 160.25,
    "uhigh": 160.34,
    "ulow": 157.85,
    "uopen": 158.86,
    "uvolume": 59256343,
    "volume": 59256343,
    "id": "HISTORICAL_PRICES",
    "key": "AAPL",
    "subkey": "",
    "date": 1679616000000,
    "updated": 1679711886000
  },
  {
```


Figure 3. Historical IEXCloud Data

```
"iexAskPrice": 152.34,
"iexAskSize": 500,
"iexBidPrice": 152.32,
"iexBidSize": 100,
"iexClose": 152.32,
"iexCloseTime": 1678307502353,
"iexLastUpdated": 1678307502353,
"iexMarketPercent": 0.014136027130389223,
"iexOpen": 152.855,
"iexOpenTime": 1678285800991,
"iexRealtimePrice": 152.32,
"iexRealtimeSize": 86,
"iexVolume": 528164,
"lastTradeTime": 1678307502353,
"latestPrice": 152.32,
"latestSource": "IEX real time price",
"latestTime": "3:31:42 PM",
"latestUpdate": 1678307502353,
"latestVolume": 37362973,
"low": null,
"lowSource": null,
"lowTime": null,
"marketCap": 2409998814720,
"oddLotDelayedPrice": null,
"oddLotDelayedPriceTime": null,
"open": null,
"openTime": null,
"openSource": "official",
"peRatio": 25.9,
"previousClose": 151.6,
"previousVolume": 56182028,
"primaryExchange": "NASDAQ",
"symbol": "AAPL",
"volume": 37362973,
"week52High": 178.3,
"week52Low": 123.98,
"ytdChange": 0.1689201077375728
},
```


Figure 4. Real-time IEXCloud Data

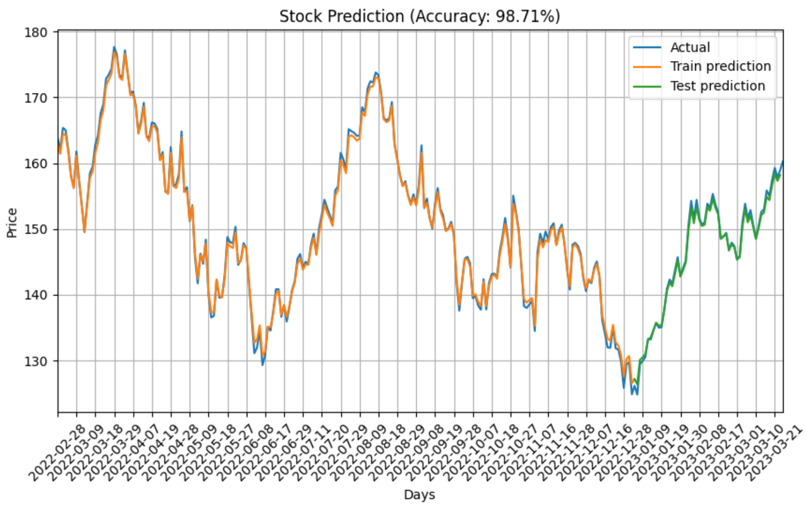
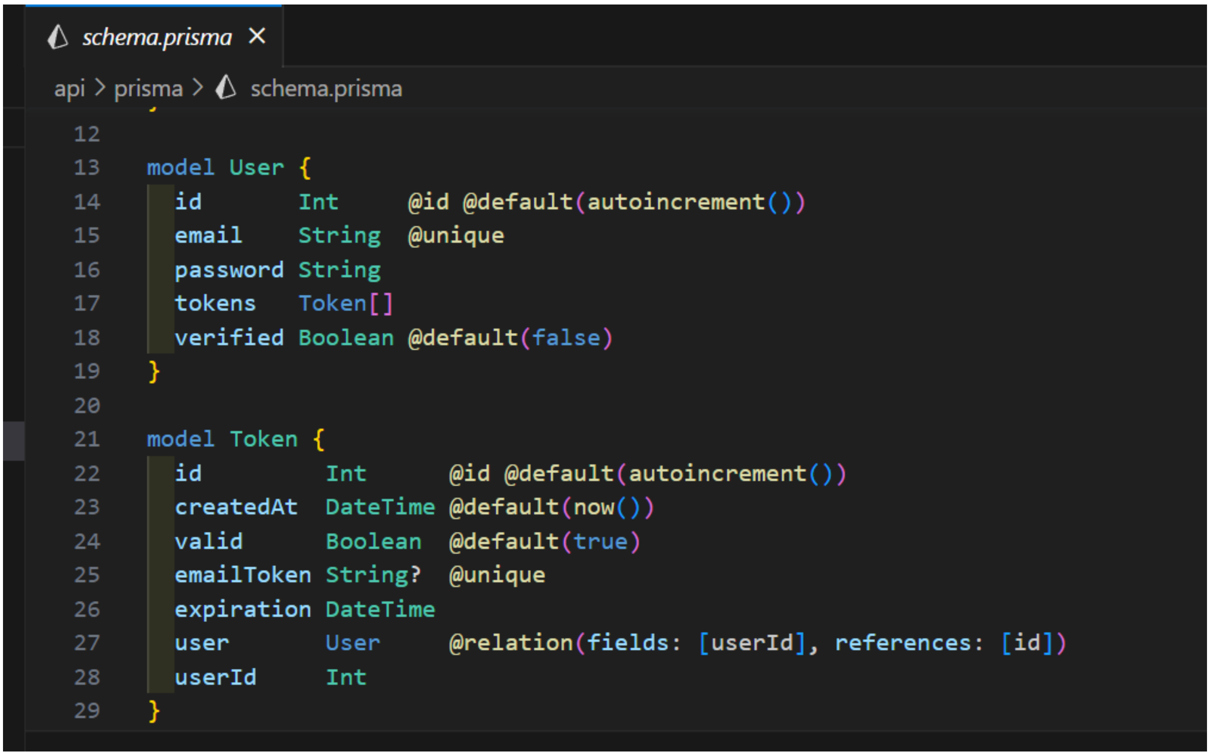


Figure 5. AAPL Matplotlib LSTM Prediction




```
model User {
  id       Int     @id @default(autoincrement())
  email    String  @unique
  password String
  tokens   Token[]
  verified Boolean @default(false)
}

model Token {
  id         Int      @id @default(autoincrement())
  createdAt  DateTime @default(now())
  valid      Boolean  @default(true)
  emailToken String?  @unique
  expiration DateTime
  user       User     @relation(fields: [userId], references: [id])
  userId     Int
}
```

Figure 6. Prisma Schema

```
server.auth.strategy('session', 'cookie', {
  cookie: {
    ttl: 24 * 60 * 60 * 1000,
    clearInvalid: true,
    name: 'sid-cm3203',
    password: process.env.COOKIEPASSWORD || '2kGt3ENW3rjmh78d7gqc6pPeQS9o4rMKPKcWyAk6KU4LTch39NUGjNBL3ULoQaVC',
    isSecure: false,
    isSameSite: 'Lax',
    path: '/',
    domain: '127.0.0.1',
  },
  validate: async (request, session: any) => {
    const user = await prisma.user.findUnique({
      where: {
        id: session.id,
      },
    });
    if (!user) {
      return { isValid: false };
    }
    return { isValid: true, credentials: user };
  },
});
```

Figure 7. API Cookie Middleware

```
# MSE (Mean Squared Error)
def mse(actual, prediction):
    return np.mean((actual - prediction) ** 2)


# RMSE (Root Mean Squared Error)
def rmse(actual, prediction):
    return np.sqrt(mse(actual, prediction))


# MAE (Mean Absolute Error)
def mae(actual, prediction):
    return np.mean(np.abs(actual - prediction))


# MAPE (Mean Absolute Percentage Error)
def mape(actual, prediction):
    mask = actual != 0
    return np.mean(np.abs((actual - prediction)[mask] / actual[mask])) * 100


# R2 (R Squared)
def r2(actual, prediction):
    return 1 - (np.sum(np.square(actual - prediction)) / np.sum(np.square(actual - np.mean(actual))))
```

Figure 8. Methods to calculate machine learning loss

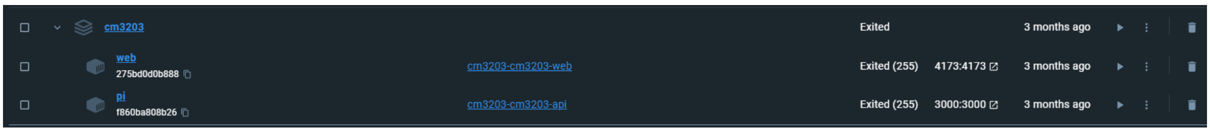


Figure 9. Docker containers

```
// Fetch historical data for a symbol and save into json file and database. Get data from 2021 till now.
async function fetchHistoricalData(key: string, workspace: string, id: string, from: string, to: string) {
  const data = await client.apperate.queryData({ key, workspace, id, from, to });

  for (const priceData of data) {
    const stock = await prisma.stock.upsert({
      where: { name: priceData.symbol },
      update: {},
      create: { name: priceData.symbol },
    });

    await prisma.historicalPrice.create({
      data: {
        priceDate: new Date(priceData.priceDate),
        open: priceData.open,
        high: priceData.high,
        low: priceData.low,
        close: priceData.close,
        volume: priceData.volume,
        stock: { connect: { id: stock.id } },
      },
    });
  }

  fs.readFile('historical.json', function (err, json) {
    if (err) {
      // If there's an error reading the file, write new data to it and log success/error
      fs.writeFile('historical.json', JSON.stringify(data), function (err) {
        if (err) {
          console.error('Error writing file:', err);
        } else {
          console.log('New file created and saved!');
        }
      });
    } else {
      // If file exists, parse the JSON data and check for updates
      const jsonFile = JSON.parse(json.toString());
```

Figure 10. Fetching historical data from the API

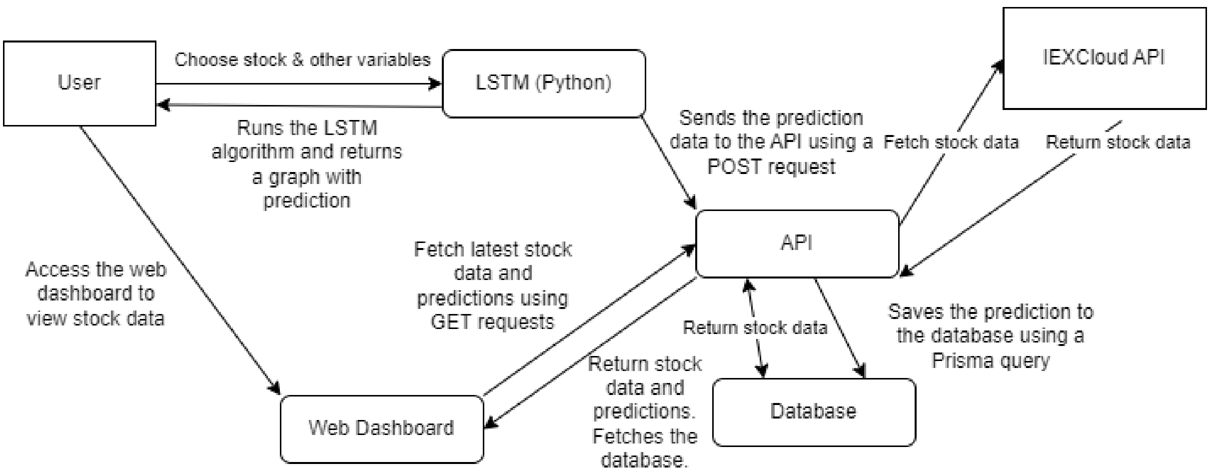
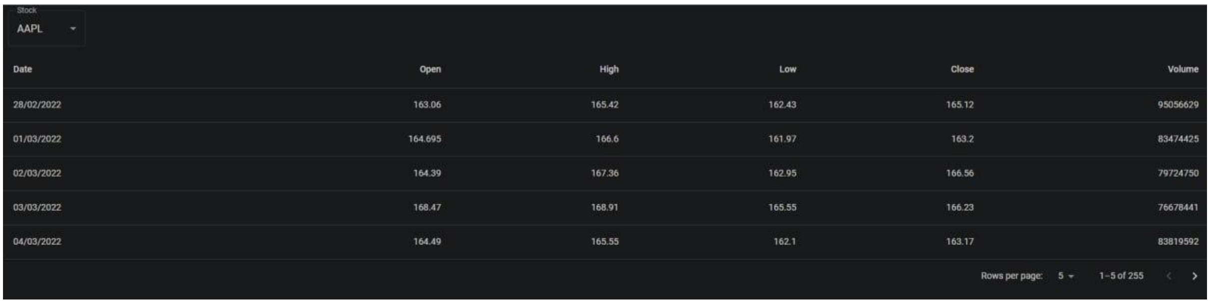


Figure 11. System Diagram

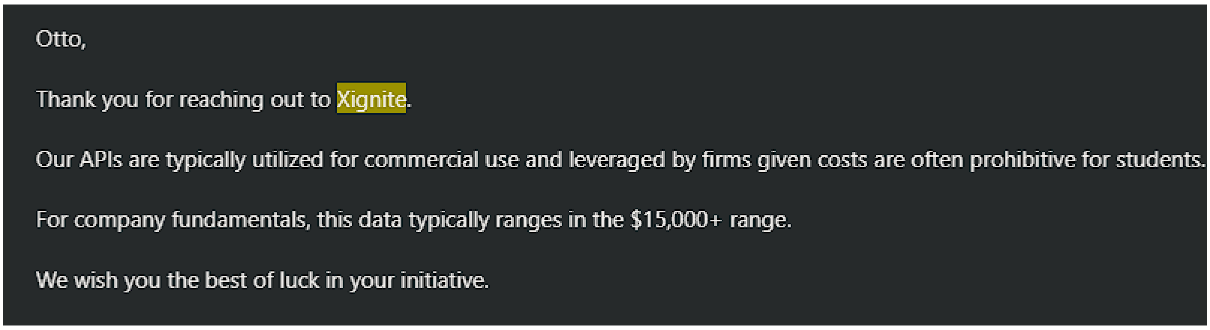


Stock
AAPL

Date	Open	High	Low	Close	Volume
28/02/2022	163.06	165.42	162.43	165.12	95056629
01/03/2022	164.695	166.6	161.97	163.2	83474425
02/03/2022	164.39	167.36	162.95	166.56	79724750
03/03/2022	168.47	168.91	165.55	166.23	76678441
04/03/2022	164.49	165.55	162.1	163.17	83819592

Rows per page: 5 1–5 of 255

Figure 12. MUI Datagrid displaying historic data



Otto,

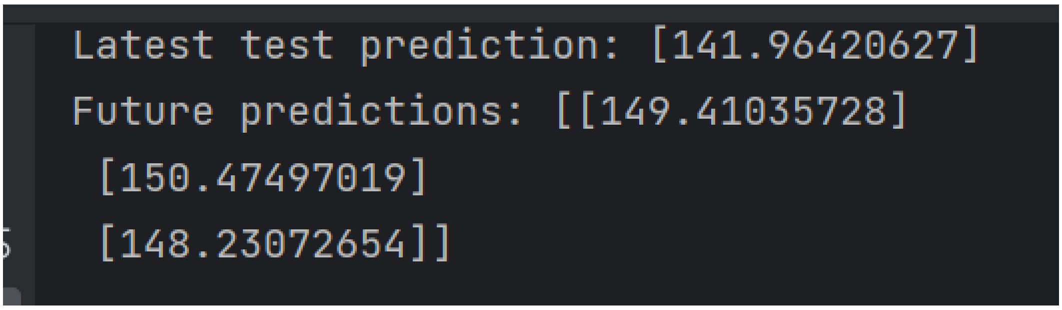
Thank you for reaching out to Xignite.

Our APIs are typically utilized for commercial use and leveraged by firms given costs are often prohibitive for students.

For company fundamentals, this data typically ranges in the $15,000+ range.

We wish you the best of luck in your initiative.

Figure 13. Xignite Email



```
Latest test prediction: [141.96420627]
Future predictions: [[149.41035728]
 [150.47497019]
 [148.23072654]]
```

Figure 14. LSTM Predictions

```
iterations: 920, training error: 169721.9954543455
iterations: 930, training error: 169721.99542696186
iterations: 940, training error: 169721.99188457997
iterations: 950, training error: 169721.99338576596
iterations: 960, training error: 169721.99403275506
iterations: 970, training error: 169721.9947147104
iterations: 980, training error: 169721.99500764176
iterations: 990, training error: 169721.99499730885
155.13427257537842
```

Figure 15. Brain.JS LSTM Prediction

```
Epoch 1000/1000
2/2 [==============================] - 0s 122ms/step - loss: 0.0033
1/1 [==============================] - 3s 3s/step
[[155.2834 ]
 [169.1567 ]
 [145.0544 ]
```

Figure 16. Keras LSTM Prediction

```
RMSE:  0.00010813636774475332
MSE:  0.011693474029028527
MAE:  0.0026926439415759233
MAPE:  0.001798363069647855
R2:  0.9999143464718427
```

Figure 17. My LSTM evaluation metrics

References

Bracke, P., Datta, A., Jung, C. and Sen, S. 2019. Machine Learning Explainability in Finance: An Application to Default Risk Analysis. *SSRN Electronic Journal*. Available at: https://www.ssrn.com/abstract=3435104 [Accessed: 14 March 2023].

Britannica. 2023. *Supply and demand | Definition, Example, & Graph | Britannica*. Available at: https://www.britannica.com/topic/supply-and-demand [Accessed: 8 May 2023].

Brownlee, J. 2019. *Understand the Impact of Learning Rate on Neural Network Performance*. Available at: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/ [Accessed: 7 May 2023].

Cascone, L., Sadiq, S., Ullah, S., Mirjalili, S., Siddiqui, H.U.R. and Umer, M. 2023. Predicting Household Electric Power Consumption Using Multi-step Time Series with Convolutional LSTM. *Big Data Research* 31, p. 100360. doi: 10.1016/j.bdr.2022.100360.

Chen, J. 2023. *What are Options? Types, Spreads, Example, and Risk Metrics*. Available at: https://www.investopedia.com/terms/o/option.asp [Accessed: 8 May 2023].

Choubey, M. 2023. *Express vs Fastify vs Hapi vs Koa: Hello world performance comparison*. Available at: https://medium.com/deno-the-complete-reference/express-vs-fastify-vs-hapi-vs-koa-hello-world-performance-comparison-dd8cd6866bdd [Accessed: 26 March 2023].

Cvitanić, J. and Zapatero, F. 2004. *Introduction to the economics and mathematics of financial markets*. Cambridge, Mass: MIT Press.

Dawid and Joanna. 2020. *Node.js vs Python: which one is better for your application?* Available at: https://www.merixstudio.com/blog/nodejs-vs-python/ [Accessed: 7 May 2023].

European Parliament. 2021. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS*. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206 [Accessed: 21 March 2023].

Fletcher, T. 2012. Machine learning for financial market prediction. Available at: https://discovery.ucl.ac.uk/id/eprint/1338146/.

Google. 2022. *Reducing Loss: Learning Rate | Machine Learning*. Available at: https://developers.google.com/machine-learning/crash-course/reducing-loss/learning-rate [Accessed: 7 May 2023].

Guida, T. 2018. *Big Data and Machine Learning in Quantitative Investment*. 1st ed. Wiley. Available at: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119522225 [Accessed: 14 March 2023].

Hart, R. 2021. *London-Based Hedge Fund Closes After Betting Against GameStop, Becoming*

One Of First Meme Stock Casualties. Available at: https://www.forbes.com/sites/roberthart/2021/06/22/london-based-hedge-fund-closes-after-betting-against-gamestop-becoming-one-of-first-meme-stock-casualties/ [Accessed: 7 May 2023].

Hasso, T., Müller, D., Pelster, M. and Warkulat, S. 2022. Who participated in the GameStop frenzy? Evidence from brokerage accounts. *Finance Research Letters* 45, p. 102140. doi: 10.1016/j.frl.2021.102140.

Henrique, B.M., Sobreiro, V.A. and Kimura, H. 2019. Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications* 124, pp. 226–251. doi: 10.1016/j.eswa.2019.01.012.

Hooper, O. 2023. *The impact of machine learning in the retail financial market*. Available at: https://forms.gle/kEjRNpVTp2XQJkSw8 [Accessed: 8 May 2023].

IBM. 2023. *Top 7 Benefits of Kubernetes*. Available at: https://www.ibm.com/cloud/blog/top-7-benefits-of-kubernetes [Accessed: 7 May 2023].

IEXCloud. 2022. *Stream Data Using SSE - IEX Cloud*. Available at: https://iexcloud.io/documentation/using-core-data/streaming-data-using-sse.html [Accessed: 7 May 2023].

James, G., Witten, D., Hastie, T. and Tibshirani, R. 2013. *An Introduction to Statistical Learning*. New York, NY: Springer New York. Available at: http://link.springer.com/10.1007/978-1-4614-7138-7 [Accessed: 14 March 2023].

John, S. 2023. *CapIQ - Overview and Guide to Capital IQ in Finance*. Available at: https://www.wallstreetoasis.com/resources/data/capiq [Accessed: 9 May 2023].

Klebnikov, S. 2022. *War Stocks Are Surging As Russia-Ukraine Conflict Rages On: Lockheed Martin, Northrop Up 20%*. Available at: https://www.forbes.com/sites/sergeiklebnikov/2022/03/04/war-stocks-are-surging-as-russia-ukraine-conflict-rages-on-lockheed-martin-northrop-up-20/ [Accessed: 26 April 2023].

Krauss, C., Do, X.A. and Huck, N. 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research* 259(2), pp. 689–702. doi: 10.1016/j.ejor.2016.10.031.

Malkiel, B.G. 2008. *A Random Walk Down Wall Street: The Time-tested Strategy for Successful Investing*. Rev Ed edition. New York: W.W. Norton & Co.

Martel, J. 2023. *The Best Bloomberg Terminal Alternatives of 2023*. Available at: https://www.investopedia.com/best-bloomberg-terminal-alternatives-5114517 [Accessed: 8 May 2023].

Mirzayev, E. 2021. *Figuring out the Beta of a Private Company*. Available at: https://www.investopedia.com/articles/personal-finance/050515/how-calculate-beta-private-company.asp [Accessed: 4 May 2023].

National Artificial Intelligence Initiative Office. 2020. *The National Artificial Intelligence Initiative (NAII)*. Available at: https://www.ai.gov/ [Accessed: 25 April 2023].

Nvidia. 2020. *Technology | GeForce*. Available at:

https://www.nvidia.com/en-gb/geforce/technologies/cuda/technology/ [Accessed: 8 May 2023].

OpenAI. 2023. GPT-4 Technical Report. Available at: http://arxiv.org/abs/2303.08774 [Accessed: 26 April 2023].

Pascanu, R., Mikolov, T. and Bengio, Y. 2013. On the difficulty of training Recurrent Neural Networks. Available at: http://arxiv.org/abs/1211.5063 [Accessed: 26 February 2023].

Shefrin, H. 2007. *Beyond greed and fear: understanding behavioral finance and the psychology of investing*. 1. issued as an Oxford Univ. Pr. paperback. Oxford: Oxford Univ. Press.

Statista. 2022. *Most used web frameworks among developers 2022*. Available at: https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/ [Accessed: 28 March 2023].

Vats, P. and Samdani, K. 2019. Study on Machine Learning Techniques In Financial Markets. In: *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*. pp. 1–5. doi: 10.1109/ICSCAN.2019.8878741.

Wayback Machine. 2023. *Grow and scale with a clear, predictable pricing model.* Available at: https://web.archive.org/web/20230131080634/https://iexcloud.io/pricing/ [Accessed: 8 May 2023].

Wong, W. 2019. *NodeJS vs. Python 3 Performance*. Available at: https://dev.to/wrrnwng/nodejs-vs-python-3-performance-1ok6 [Accessed: 7 May 2023].

Yang, G., Yuan, E. and Wu, W. 2022. Predicting the long-term CO2 concentration in classrooms based on the BO–EMD–LSTM model. *Building and Environment* 224, p. 109568. doi: 10.1016/j.buildenv.2022.109568.