Implementation of an Automated Trading System by Leveraging Data Science and AI Techniques

By

Anton Augustine Abraham

Supervior

Konstantinos Tsakalidis

Second Marker

Paul Spirakis

A DISSERTATION

Submitted to

The University of Liverpool

in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

23 September 2022

ABSTRACT

Implementation of an Automated Trading System by Leveraging Data Science and AI Techniques

By

Anton Augustine Abraham

Advancements in AI and data science, availability of better computational resources and open source financial data have all collectively led to massive improvements in research and developments in algorithmic trading. These recent advancements in research on algorithmic trading practices, give traders an opportunity to leverage the massive computational and infrastructure capabilities that the modern world offers. These developments will ultimately increase the effectiveness of algorithmic trading and offer potential for better returns compared to manual traders.

In this project, I build and implement a pipeline for automated trading of stocks by using data science and AI techniques. This pipeline consists of several steps which include, data extraction, relevant data preparation, building an LSTM recurrent neural network model which forecasts closing prices for the next trading day, development of a trading strategy based on this forecast coupled with few financial indicators and finally execution of trades on the stock markets using virtual paper trades, following the developed trading strategy. This implementation is also meant to study and adopt various researches in this area and provide a complete pipeline/methodology for developing and implementing an automated trading strategy using data science and artificial intelligence.

The LSTM models built for stocks of both Apple and Microsoft showcased measures which are pointed towards a fairly predictive forecast with an R2 Score greater than 97%. Evaluation of the performance of the trading strategy developed based on this model is carried out by measuring the growth of the virtual portfolio that Alpaca offered. This accounts for transaction costs and offers a simulation of its performance on the stock markets. The results of this study are promising and show potential for generation of returns higher than an annual rate of 10%.This is particularly true for days when the forecasted close price change (compared to previous day's close price) follows a similar trend to that of the actual changes.

DECLARATION

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Anton Augustine Abraham

ACKNOWLEDGEMENTS

I would like to express my sincerest thanks to my supervisor, Konstantinos Tsakalidis, for his support and guidance throughout the entirety of this project. I am extremely grateful for the meetings and talks we had during this time which really helped me both academically and professionally. I'd also like to thank my secondary marker, Paul Spirakis, for his thoughts and suggestions during the presentation of this project.

Furthermore, I would like to thank KX Systems, Google Colab and Alpaca, for providing me with the required licenses and accounts, to implement this project. I am also thankful to all the professors I had during my masters and to the University of Liverpool for providing me with the knowledge and skill sets, which helped me complete this research.

Finally, I cannot forget to thank my family and friends for all the un-conditional support that they have provided me, during this project.

TABLE OF CONTENT

Page

LIST OF TABLES **vi**

LIST OF FIGURES **vii**

Chapter 1. Introduction **1**

Chapter 2. Aims And Objectives **2**

Chapter 3. Background And Literature Review **3**

Chapter 4. Ethical Use Of Data **7**

Chapter 5. Design And Implementation **9**

Chapter 6. Evaluation **13**

Chapter 7. Learning Points **16**

Chapter 8. Professional Issues **17**

Chapter 9. Conclusion **19**

REFERENCES **20**

Appendix A. Additional Material **23**

- **A.1 Extracted Data** **23**
- **A.2 Adjusted Close Prices** **23**
- **A.3 Differencing** **24**
- **A.4 Standardization** **25**
- **A.5 LSTM Model** **25**
- **A.6 Trading with Alpaca** **26**
- **A.7 Original Design** **27**
- **A.8 Model Evaluation** **29**

LIST OF TABLES

Page

Table 1. Model Performance.....14

Table 2. AAPL - Trade Results.....15

Table 3. MSFT - Trade Results.....15

LIST OF FIGURES

Page

Figure 1. Proposed Pipeline 9

Figure 2. Data Extracted for AAPL 23

Figure 3. AAPL – Adjusted Close Prices 24

Figure 4. First Order Differencing 24

Figure 5. Standardization 25

Figure 6. Model-AAPL 25

Figure 7. Model-MSFT 26

Figure 8. Market Orders 26

Figure 9. Portfolio Growth 27

Figure 10. Original Design 28

Figure 11. Forecast - Differences 29

Figure 12. Forecast - Prices 29

Chapter 1. INTRODUCTION

1.1 Scope

The scope of this project is to implement an automated trading strategy for selected stocks by using a neural network model and certain technical indicators. The forecasting model is used alongside the technical indicators, to derive a trading rule, which is then used to make live paper trades during daily market hours.

1.2 Problem Statement

Advancements in AI and data science, availability of better computational resources and open source financial data have all collectively led to massive improvements in research and thereby the effectiveness of algorithmic trading. Implementing an automated day trading system which is able to make profits for a trader is of particular interest, in this context. The problem statement for this project is to build and test a pipeline for this implementation in a live market using paper trades, by leveraging data science and AI techniques.

1.3 Approach

The approach adopted is to build this pipeline which has the following components:

- Data Extraction for selected stocks from the internet.
- Data Preparation for the modeling process.
- Building a Long Short Term Memory (LSTM) Neural Network model which forecasts the close prices for the next trading day.

- Trading rule is generated based on the forecasted close price and certain financial trend indicators which are used to exploit price directional movements within a trading day.

- The generated trading algorithm is then implemented in the NASDAQ stock exchange with virtual paper trades using the Alpaca broker API platform.

1.4 Outcome

The final outcome of this project is the implementation of the proposed pipeline on selected stocks. This project is aimed at helping any beginner to build a similar trading system that contains a pipeline for data extraction, building a forecasting model along with a trading rule based on the model, and then to implement this strategy on a live market utilizing broker API's. The effectiveness of this strategy was tested for a few days for stocks of both Apple and Microsoft in the NASDAQ exchange.

The evaluation metrics for model performance showcase the capability of LSTM models to capture the actual trend in a time series and to predict stock prices fairly accurately. A return in between -0.20% to 0.15% were obtained for any of the trading day's tested, with the generated trading strategy. Positive returns were mostly obtained when the percentage change of the forecasted close price to the previous close price is similar in direction to the percentage change in actual close prices. Conversely, negative results were obtained when the forecasted change was in an opposite direction to the actual change of price.

Chapter 2. AIMS AND OBJECTIVES

2.1 Aims

The main aims of this project are:

1. Understand existing studies on algorithmic trading to forecast stock price values and to further research efforts in this area.

2. Study various approaches for generating and implementing automated trading rules in the stock market.

3. Propose a pipeline for learners in this area to:

    - Extract financial data from various public sources available

    - Program a trading strategy using technical analysis indicators and machine learning techniques.

    - Implement and test its effectiveness in a live market environment by availing free to use trading API's of brokers.

2.2 Objectives

- Implement a stock price prediction model which is able to forecast stocks prices.

- Generate a trading rule based on this model which can show at least a 10% annualized return on the virtual capital, provided by the trading API's.

- Comparison against standard benchmark financial models: Testing of standard financial models is carried out to compare the formulated trading rule against other benchmarks.

Chapter 3. Background And Literature Review

Johnson (2011), in his book on Algorithmic Trading and DMA provides an introduction to several Direct Market Access strategies to help traders achieve better execution for their

trades. He provides a detailed introduction on market structures, different trading algorithms, and on the different types of order placement and execution tactics in the market.

Conventional financial forecast models similar to Box et.al (2015), uses a linear combination of historical stock prices and other covariates, to predict expected stock returns, by adding a noise term to their model which they hope will capture market uncertainty. The use of machine/deep learning algorithms in data driven financial modeling, in recent times, have led to major improvements in predicting stock prices and directional changes.

Creamer et.al. (2010), proposed an automated trading system that uses boosting techniques combined with several expert recommendations. They were able to demonstrate positive abnormal returns for a large group of stocks by incorporating a risk management layer on top of their algorithm that selects trades with a stronger prediction and by avoiding trades during a continued history of negative performance.

Performance based Regularization and Cross Validation techniques were applied for the portfolio optimization problem discussed by Ban et.al (2018) and they showed that the adoption of these machine learning techniques improved the performance of their data driven strategy compared to the Sample Average Approximation method and other models that they tested.

Neural Network based models have been tested for many price prediction use cases like foreign exchange rate prediction as in Khashei & Bijari (2010, 2012). They used the outputs of ARIMA and input financial data as the inputs for their NN model, and demonstrated a better prediction for time series forecasting of USD and GBP exchange rates, compared to the results of ARIMA and other benchmark models. Hsieh et.al. (2011), presented an integrated system where wavelet transforms and Recurrent Neural Network based on artificial bee colony algorithm where combined to forecast stock price movements. Chen et.al (2017) developed a neural network framework which purely focused on the returns generated from high frequency data.

Another approach to integrate a classical financial model ARMA-GARCH (Auto Regressive Moving Average Model with Generalized Auto Regressive Conditional Heteroskedasticity) with artificial neural networks was adopted by Sun et.al (2019), to predict the directional change of prices under a high frequency (HF) scenario. This proposed model they say is able to capture the intra-day patterns for stock market shock forecasting, without strong distributional assumptions. They used custom designed feature selection and cross validation methods and showed that their proposed solution yielded a better trading strategy compared to ARMA—GARCH and other traditional financial models.

Better infrastructure and computational capabilities, along with cutting edge technological advancements have given quantitative traders the opportunity to exploit profitable opportunities faster with high frequency trades. Virgilio (2019), collates opinions of several academic researchers for the potential of making abnormal profits using High Frequency trading. Even though there seems to be a considerable lot who demonstrated of having generated large profits with sufficiently huge investments, he notes that the increase in competition over recent years may have reduced those profits.

Jiang (2021) surveys more than 100 published researches (conducted in recent years), relating to the applications of deep learning in stock market prediction. It provides beginners in this area with a general workflow that can be adopted for an implementation. They provide an overview of the various data sources, different neural network structures and describe the commonly used evaluation metrics that are often adopted in these researches. They also had a focus on highlighting the availability of these implementations and its reproducibility which could help beginner researchers adopt them as baselines.

Lu et al. (2020), proposed a CNN and LSTM based approach to forecast stock prices. They used Convolutional Neural networks to extract features from the data and then pass it on to the LSTM model to predict the time series of stock prices. They used various inputs like the opening price, highest price, lowest price, closing price, volume, ups and downs and turnover

to analyze the time sequence characteristics of this data. They used the previous ten days of data to forecast the next day's close price. They conclude that the CNN-LSTM approach can be a reliable method for stock price forecasting with high prediction accuracies.

Gao et.al. (2020), tried to predict the next day's index price using four different types of deep learning models. Three of them were conventional models like multi-layered perceptrons, Long Short Term Memory and Convolutional Neural Networks while the fourth and best performing one was an attention-based neural network. They utilized a feature space consisting of technical indicators, daily trading data and macro-economic factors to derive these models.

Conegundes et al. (2020), investigated the potential of adopting deep reinforcement learning to day trade stocks in the Brazilian Stock Exchange. They used a Deep Deterministic Policy Gradient algorithm to determine the allocation of assets, during day trading operations. Liu et.al (2021) proposed an open-source framework "FinRL", which helps easy implementation of multiple Deep Reinforcement Learning algorithms for various financial applications like stock/crypto trading, portfolio allocation, high frequency trading, market regulations, etc. It provides a full pipeline to help quantitative traders, design a strategy and execute them using trading API's. The three layer architecture they have allows the user to choose the type of application in finance, type of deep relationship learning agent and the type of market environment which supplies the data required and the trading environment to execute trades.

A similar data science pipeline was implemented by Zhang et.al (2022) and demonstrated that it is applicable to extend on to conventional algorithms like, the moving average crossover, volume-weighted average price, sentiment analysis and statistical arbitrage techniques. The proposed framework, which is able to extract necessary financial data, program a trading strategy, and then execute and test its effectiveness on the live markets. They conclude that this pipeline is suitable for algorithmic trading and were able to demonstrate this implementation in the crypto trading markets.

Long Short Term Memory

Long Short Term Memory (LSTM) is a particular variant of recurrent neural networks, which provides a better performance for modeling problems with long term dependencies. Conventional RNN based models suffer from the issue of vanishing gradients, which makes it difficult for the model to learn long-term dependencies. LSTM's were proposed in 1997 by Hochreiter et.al., as a solution for the vanishing gradient problem observed in traditional recurrent neural networks.

While the traditional RNN's has only one layer in the neural cell, LSTM's have four layers which interact with each other. The common LSTM structure is composed of a cell, an input gate, a forget gate and an output gate. The cell state of the LSTM unit is capable of remembering values over different time intervals and the three gates are used to control and regulate the flow of data into and out of the cell state. This gated mechanism is a way to pass information selectively through LSTM's.

The forget gate of the LSTM decides about the information to be discarded from the cell state. The input gate decides which information is to be added newly into the cell state and the output gate determines the output. LSTM based networks are particularly suited for time-series based data, since there can be lags of unknown durations observed between important events in a time series. This makes LSTM's a popular choice for stock price prediction problems in recent years. Apart from time series predictions, LSTM's are largely used in a variety of sequence prediction tasks such as speech recognition, language translation, sentiment analysis, video analysis etc.

Chapter 4. Ethical Use Of Data

Autonomous trading strategies have a huge role to play in the financial markets particularly in this age of abundance of information and rapid technological advancement. These solutions may make the market more effective and help reduce transaction costs for quantitative traders. However, the ethical aspects of it will need to be seriously considered as untoward ma-

nipulations are not to be permitted in the financial markets. Wellman et.al. (2017) discusses the major ethical issues in implementing these strategies and concludes that there exists sufficient concerns around agent misbehavior which may impact the financial markets vulnerably.

The data sources provided by yfinance and Alpaca are all publically available and are free to use for the public. The major features in the data are the prices observed during a day, which are available publically. In addition, the broker API's that I have used to extract real time data and test the implementation are regulated by financial authorities. Alpaca is regulated by the US Securities & Exchange Commission (SEC) and Financial Industry Regulatory Authority (FINRA). Since the purpose of this implementation is purely for research and not for trading with real money or to make investment advices, Alpaca doesn't require a license to be purchased.

Installation and use of kdb+ database requires a license and this was provided by KX Systems for educational/research purposes. But, since the contract was provided for use only in an offline set-up and the installation was not to be integrated into a live trading environment, kdb+ architecture for data storage and retrieval was not integrated into this implementation.

I have read the ethical guidelines to be followed for this dissertation and have followed them fully. I have not collected human data as part of this project and I have not violated any of the ethical considerations that are important for such an implementation. This project is only intended for research and educational purposes, and was not intended to trade in the markets with real money to make profits.

Chapter 5. DESIGN AND IMPLEMENTATION

The proposed implementation was built using Python Programming Language using Google Colab Notebooks. As described, the implementation pipeline consisted of five major steps, as summarized in Figure 1. These steps are further explained in the sections below.

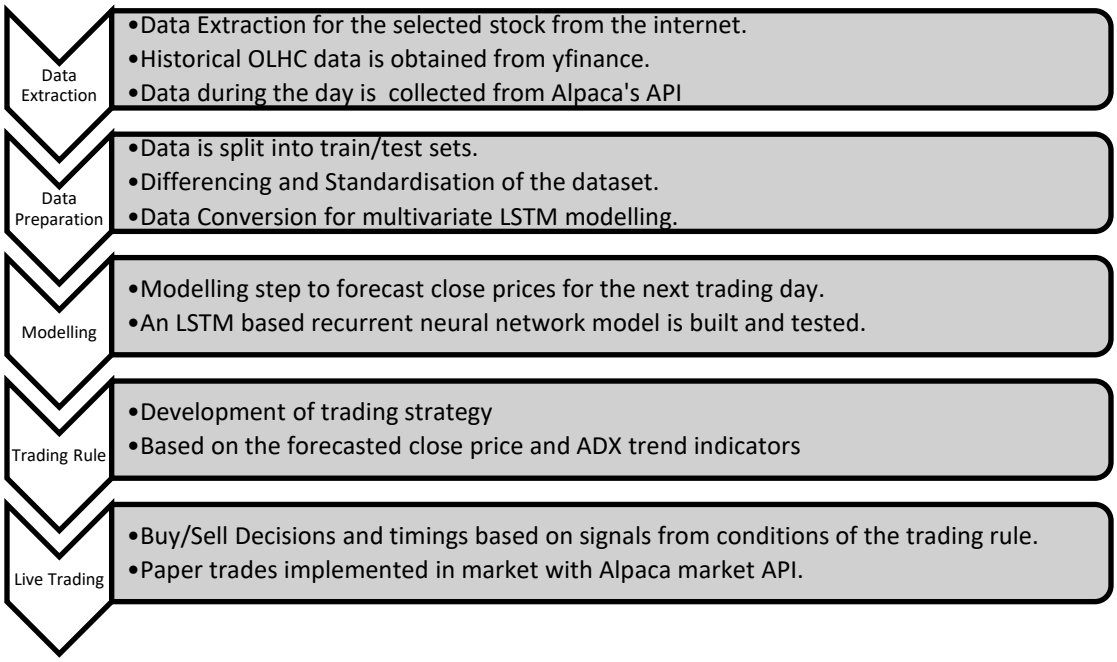

Figure 1. Proposed Pipeline

5.1 Data Extraction

I have primarily chosen stocks of Apple (AAPL) and Microsoft (MSFT) for this analysis. The data extracted consisted of six features for each trading day. These included prices of opening, closing, high, low and the adjusted close values, along with the volume of trades during any particular day. Historical data for modeling and testing were extracted from yfinance. A total of 5682 trading days between the start of 2000 to the 17th of August 2022 was extracted from yfinance, for both training and testing the model.

The data required for implementing live trades needed to be present for shorter time intervals (of minutes) and this was extracted using the Alpaca API. Historical data for modeling was saved on the drive for the modeling step, while, data required for making trade decisions during the day was extracted for intervals of a minute while the trading algorithm was being executed. Figure 2 in Appendix A.1 shows an extract of the data for stock of AAPL from yfinance.

5.2 Data Preparation

The first step was to split the data into the training and test partitions. 5273 trading days in between 4th January 2000 to 31st December 2020, was used for training the model. The remaining days until 17th August 2022, with 409 trading days were used for evaluation. Figure 3 (Appendix A.1) shows the adjusted close prices for AAPL during the training period.

A first order differencing was applied for all price features to remove/reduce the effect of trend and seasonality in the data. For example, the close price of the current day is subtracted from the close price of the previous day, to derive the differenced close price for the current day. This reduces the temporal dependence on time, to an extent, for these features. Figure 4. First Order Differencing, shows the adjusted close prices for AAPL after differencing.

Standardization of data is also carried out, before the modeling step, for all input features. This step converts the differenced prices of each feature into a vector of mean zero and unit variance. Standardization is important to be carried out before feeding the input features into a neural network model, as it generally speeds up learning and leads to a faster convergence. Figure 5. Standardization, describes the standardized close price values for AAPL after the differencing operation.

The feature for prediction is taken to be the adjusted close price for the next day, and not the actual close price, as it is considered a more accurate measure of a stocks intrinsic value.

Adjusted close price accounts for the various corporate actions like new stock offerings, dividends and stock splits.

The modeling step is approached as a multivariate time series forecast problem and a certain number of time steps are used to make a prediction for the next trading day. The number of time steps can be tested for a varied number of values, but I have used a fixed number of ten time steps in this implementation. Lu et al. (2020) and a significant number of other researchers have also adopted ten as the number of time steps. There are researches which use other values for time steps with seven being the significant other value that is majorly used. All the price features for the previous ten days are used to generate the predicted value of the adjusted close price for the current/next trading day.

Data for LSTM modeling is prepared with a 3 Dimensional input shape of the form: (number of samples, number of time steps, number of features). That is, each sample for training will consist of all the features of the previous ten time steps. The target value for this particular sample will then be attributed as the adjusted close price for the current/next trading day.

5.3 Modeling

The approach to predict close prices for the next trading day is by viewing it as a multivariate time series forecast problem. The proposed model that is adopted in this implementation is an LSTM based recurrent neural network, which is particularly suitable for learning long term dependencies over time. The model structure adopted is a stacked network with 2 LSTM layers and a final dense layer that outputs the predicted value.

The model for each stock was trained separately and hence different parameter combinations were tested for each of them. The training dataset for each stock was split into different validation splits by making use of Time Series Split functionality provided by sklearn package in Python. This was performed so as to tune the models with the optimal hyper-parameter com-

bination. The final model structure for both AAPL and MSFT are described in Figure 6. Model-AAPL and Figure 7. Model-MSFT, respectively.

The predicted value from the model for any test sample will need to be inverse transformed from the standardization to get the predicted difference from the previous trading day. This value can then be added to the previous adjusted close price to obtain the forecasted value for the current/next trading day.

5.4 Trading Rule

The trading rule that is tested in this implementation is based on the forecasted close price and the Average Directional Movement Index (ADX) indicator. ADX value for a particular financial instrument indicates the trend strength for a time series of price values. Along with it, two Directional Indicators are also obtained which helps identify if the current trend is positive or negative. These trend indicators are used to exploit price directional movements within a trading day.

Trading Signals are generated to Buy/Sell stocks based on if different conditions are met in respect to the current price of a stock, the forecasted close price for the day, and the ADX indicators. The volume of stocks bought/ sold are currently set as fixed values based on the strength of the generated signal, but this isn't ideal and can be further improved with other approaches to determine the volume of trades.

5.5 Trading

The execution of trades starts when the NASDAQ stock exchange opens for any trading day. The trading strategy runs during the day and starts executing its exit strategies when the market is close to end on any trading day. It either takes a profit when the profit generated is above a defined margin, or takes a loss if these requirements are not expected to be met be-

fore close. Thus, all positions bought/sold during the trading day are closed before the market ends. The trading strategy is implemented as virtual paper trades, by leveraging Alpaca API's paper account.

5.6 Design Changes

The original design proposed (as in Figure 10. Original Design) has a few changes compared to the one which was finally implemented. The major change was the problem definition as it was re-formulated from predicting the direction of price movement, to a forecasting problem to predict actual prices. Hence this changed the modeling approach from a classification problem to that of a time series forecast based approach.

I intended to integrate kdb+ architecture for efficient data storage, retrieval and processing of time series data. This was proposed to be implemented by storing historical information in the drive memory of the database, and by adding the data for the current day to this database at the end of the day. This was to be implemented by using the ticker plant configuration for storing data related to different stocks. This was not incorporated as I faced issues in running the whole implementation in my local system and thereby having to rely on Google Colab to develop this. The kdb+ license was not to be used in an online set-up and it couldn't be connected to Colab as the software is installed in the local machine. Hence, kdb+ architecture for efficient data storage and retrieval was not integrated into this implementation.

Another change in the design approach is the use of financial indicators to derive the trading rule. The use of ADX based indicators was utilized to exploit intra-day price movements.

Chapter 6. EVALUATION

Analyzing the model's forecasted values on the test dataset, demonstrates certain interesting points about its overall performance. The final predicted values for the adjusted close prices

of stocks shows a very promising result, compared to the actual prices. An R2 Score higher than 97% and a mean absolute percentage error of lesser than 1.5%, for models of both the stocks shows that this approach is able to forecast close prices fairly accurately. Figure 12. Forecast - Prices describes a plot of the predicted and actual close prices for stocks of MSFT on the test dataset.

Since the output from the model is the difference between prices between the current days to the previous one, I calculated these metrics on these values as well. The results on these difference values don't look great, with a negative R2 Score, showcasing an inability to predict and capture the trend for actual differences adequately. But, since most of the researches base their analysis on the forecasted prices and not on the differenced prices, this evidence is given priority for the final evaluation of the models performance. Table 1. Model Performance describes the results on the test data for models developed for both AAPL and MSFT.

Stock	AAPL		MSFT	
Evaluation Metric	Forecasted Prices	Forecasted Differences	Forecasted Prices	Forecasted Differences
Mean Square Error (MSE)	7.548	7.588	22.774	22.467
Root Mean Square Error (RMSE)	2.747	2.755	4.772	4.740
Mean Absolute Percentage Error (MAPE)	1.41%	2.85E+12	1.32%	2.62E+12
R2 Score	97.30%	-4.34%	97.76%	-1.99%

Table 1. Model Performance

These evidences suggest that there are massive advantages in using an LSTM recurrent neural network for time series prediction on stock prices. Still, there are massive improvements to be made in adequately capturing the trend of stock prices. Addition of further features to the data by methods such as sentiment analysis, or use of technical indicators, etc., as features can be thought of as a potential next step to better the performance. Figure 4. First Order Differencing illustrates that there are larger differences observed during the end of the training period. Even larger values of differences are observed during the test period. This

is to be naturally expected for a time series on prices, and hence, potential solutions for improvements are to be found. Higher order differencing and removal of stationarity (even though LSTM's do not generally require corrections for stationarity) can also be tested to boost performance. Training with a larger dataset with more number of trading days or model's trained closer to the days requiring forecast can also be considered as further steps.

Test Day	Close Prices($)			Change(%) from Previous Close		Cash($) : Trading Account		Profit/Loss	P/L(%)	ARR(%)
	Previous Close	Predicted Close	Actual Close	Predicted	Actual	Start	End			
22-08-2022	171.52	172.21	167.57	0.4011%	-2.30%	1,00,000.00	99,835.99	-164.01	-0.16%	-33.88%
23-08-2022	167.57	168.26	167.23	0.4130%	-0.20%	99,835.99	99,770.79	-65.2	-0.07%	-15.18%
24-08-2022	167.23	167.91	167.53	0.4082%	0.18%	99,770.79	99,898.48	127.69	0.13%	38.03%
25-08-2022	167.53	167.44	170.03	-0.0551%	1.49%	99,898.48	99,887.58	-10.9	-0.01%	-2.71%

Table 2. AAPL - Trade Results

Table 2. AAPL - Trade Results demonstrates the results of trading with the developed trading strategy with stocks of Apple. Most of the trading days were found to have returned negative returns. But for those days, we see that the predicted change in close price (compared to the previous day's value), followed an opposite direction to the actual changes observed. A similar result is obtained while applied for stocks of Microsoft, wherein positive returns where received when the change in forecasted price followed the same direction as the actual movement. This is illustrated in Table 3. MSFT - Trade Results.

Test Day	Close Prices($)			Change(%) from Previous Close		Cash($) : Trading Account		Profit/Loss	P/L(%)	ARR(%)
	Previous Close	Predicted Close	Actual Close	Predicted	Actual	Start	End			
19-09-2022	244.74	243.75	244.52	-0.4044%	-0.09%	1,00,169.85	1,00,210.38	40.53	0.04%	10.73%
20-09-2022	244.52	243.56	242.45	-0.3932%	-0.85%	1,00,210.38	1,00,222.82	12.44	0.01%	3.18%
21-09-2022	242.45	241.67	238.95	-0.3233%	-1.44%	1,00,222.82	1,00,360.03	137.21	0.14%	41.17%

Table 3. MSFT - Trade Results

This illustrates that with a better model with improved forecasts that could accurately predict the change in close price to the previous value; a profitable trade strategy could be implemented in live stock exchange markets. It might also be a better option to treat it just as a

classification problem (as in the original design) and predict the direction of price movement than forecasting actual prices, as pointed out by a lot of previous researches. Another point worth noting was the fact that the market data provided by Alpaca was delayed by 15 minutes (due to the use of a free subscription) and hence, there is reason to expect better returns if real time data is available.

Chapter 7. LEARNING POINTS

This dissertation has been a great learning experience for me and there were multiple takeaways from it. The proposal and literature review stage of this project helped me understand a great deal about the existing research and developments in this area. There were multiple design changes observed from the initial proposal, but these helped me read and understand a lot more of other studies. Discussions and meetings with my supervisor were also really helpful and they helped shape the design and enhancements for this implementation.

I learnt a lot about the different financial markets, market access strategies, trading algorithms and about the various order placement and execution strategies in these markets. I was also able to develop my knowledge about various financial indicators which are relevant for price prediction. Alpaca offered the possibility to execute virtual trades in the markets and these helped me get an understanding of how to extend these steps into the markets with cash trades.

The initial plan was to use R software for this implementation, but it was later changed into Python and I used Google Colab as the platform. This was done so as to utilize the abundance of packages in Python for the problem at hand. Development of this implementation has greatly improved my coding capabilities in Python. As the original design had an integration of kdb+ database for data storage and retrieval, this helped me understand the basics of the database and its query language Q. Even though kdb+ wasn't incorporated finally, this initial study gave me an understanding of the capabilities that this time series database offers in terms of execution speeds and efficiency.

Having initially tried making a classification model to predict the expected direction of trend, it helped me view the problem statement differently. I was able to learn a great deal about the major data preparation steps needed to build an LSTM forecast model to predict prices. This project gave me the opportunity to understand recurrent neural networks (and LSTM's in particular) and neural network based forecast models in the context of stock price prediction. I was also able to understand about the various approaches that can be adopted as cross validation procedures in relation to a time series forecast model.

Chapter 8. PROFESSIONAL ISSUES

This project was carried out following the Code of Conduct set by the British Computer Society (BCS), the Chartered Institute for IT. This helps ensure relevant professional standards to be met during the course of this work. The Code of Conduct has four key principles which have all been adhered to during this dissertation.

8.1 Public Interest

This project does not compromise the public health, security, wellbeing and privacy of other individuals or for the environment. I have also not performed any action that could manipulate the financial markets inadvertently. Third party rights and compliances to agreements with them have been thoroughly followed. I have done this research respecting the agreements made with both KX Systems and Alpaca. I have also not discriminated against any individual on any grounds to fulfill my research activities. I have also used publically available sources of information and generated reproducible code, so as to promote equal access to the benefits of this implementation for anyone of the public.

8.2 Professional Competence and Integrity

I undertook this project with an awareness of my professional competence and the belief of developing it over the course of this work. I have not falsely claimed of my skills or awareness of this topic and gained unfairly over it. This project was also implemented to develop my competence in these fields over the duration of this work. I have undertaken this research having knowledge of the Legislations/agreements that had to be complied. I believe I have accepted feedback and alternative viewpoints from my supervisors and acted positively for their honest criticisms of my work. I also declare that I haven't bribed or harmed others to complete this implementation.

8.3 Duty to Relevant Authority

I declare that I have carried out this project with due care and diligence for University of Liverpool's assessment requirements. I have not plagiarized or falsely claimed ownership of work done by others to fulfill these requirements. I accept full responsibility for the work that I've undertaken and there is no conflict of interests between me and the University on this regard. I also vouch not to be misrepresenting or withholding information to complete this dissertation.

8.4 Duty to the Profession

I declare that I have carried out this project accepting my duty to the profession and have not taken part in activities which could bring the profession into disrepute. I have always strived to maintain and improve professional standards and to uphold the reputation of BCS. I also vouch to have acted with respect and integrity to fellow members of the profession and hope to have added value to any fellow members referring this project of my implementation for their development.

Chapter 9. CONCLUSION

The major problem statement for this project was to implement a pipeline for automated trading using data science and AI techniques, understand existing research in the area and help extend research developments in this field. A pipeline which consisted of data extraction, forecasting close prices with an LSTM model, development of a trading strategy and then executing and testing this strategy on live markets was developed using Python and Alpaca. It was also demonstrated that positive returns (greater than 10% annually) could be generated during specific trading days.

The Long Short Term Memory Models that were developed for stocks of Apple and MSFT showed a Mean Absolute Percentage Error of 1.41% and 1.32% respectively. The R2 Scores obtained for both these models were 97.3% and 97.76% respectively. These metrics showcase the capability of LSTM models to capture the actual trend in a time series and to predict stock prices fairly accurately.

Trading based on the developed strategy generated positive returns when the change in forecasted close prices follows a similar trend to that of the actual change (compared to the previous close price). This was observed while trading for both stocks of Apple and Microsoft. This showcases the potential for generating profits with the trading strategy that was formulated.

It might also be a better option to treat the modeling approach as a classification problem (as in the original design) and predict the direction of price movement than forecasting actual prices, as pointed out by a lot of previous researches. Addition of a larger feature set consisting of sentiment analysis from various sources, integrating trade level data, use of financial indicators, etc., can all be further steps in improving the performance of the model and thereby the trade strategy. Use of a much larger set of technical indicators, or models that could predict the volume and timing for executing buy/sell decisions could also be incorporated further for improved trade results.

REFERENCES

Alpaca Learn | Developer-First API for Crypto and Stocks. (2021). *Broker-Dealer, Investment Advisor, or No License? How to start your research journey on regulatory licenses*. [online] Available at: https://alpaca.markets/learn/broker-dealer-investment-advisor-or-no-license-how-to-start-your-research-journey-on-regulatory-licenses/ [Accessed 7 Jul. 2022].

alpaca.markets. (n.d.). *Overview*. [online] Available at: https://alpaca.markets/docs/introduction/ [Accessed 7 Jul. 2022].

Ban, G.-Y., El Karoui, N. and Lim, A.E.B. (2018). Machine Learning and Portfolio Optimization. *Management Science*, [online] 64(3), pp.1136–1154. doi:10.1287/mnsc.2016.2644.

Bcs.org. 2022. [online] Available at: <https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf> [Accessed 23 September 2022].

Box, G.E.P. and Al, E. (2015). *Time series analysis : forecasting and control*. Hoboken, New Jersey: John W iley & Sons.

Chen, H., Xiao, K., Sun, J. and Wu, S. (2017). A Double-Layer Neural Network Framework for High-Frequency Forecasting. *ACM Transactions on Management Information Systems*, [online] 7(4), pp.1–17. doi:10.1145/3021380.

Cheng, J.-H., Chen, H.-P. and Lin, Y.-M. (2010). A hybrid forecast marketing timing model based on probabilistic neural network, rough set and C4.5. *Expert Systems with Applications*, [online] 37(3), pp.1814–1820. doi:10.1016/j.eswa.2009.07.019.

code.kx.com. (n.d.). *Developing with kdb+ and the q language - Kdb+ and q documentation*. [online] Available at: https://code.kx.com/q/ [Accessed 8 Jul. 2022].

Conegundes, L., and. Pereira, A. C. M., 2020, "Beating the Stock Market with a Deep Reinforcement Learning Day Trading System," *International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9206938.

Creamer, G. and Freund, Y. (2010). Automated trading with boosting and expert weighting. *Quantitative Finance*, 10(4), pp.401–420. doi:10.1080/14697680903104113.

Gao, P., Zhang, R. and Yang, X., 2020. The Application of Stock Index Price Prediction with Neural Network. *Mathematical and Computational Applications*, 25(3), p.53.

Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735-1780.

Hsieh, T.-J., Hsiao, H.-F. and Yeh, W.-C. (2011). Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied Soft Computing*, 11(2), pp.2510–2525. doi:10.1016/j.asoc.2010.09.007.

Jiang, W., 2021. Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, 184, p.115537.

Johnson, B. (2011). *Algorithmic trading & DMA : an introduction to direct access trading strategies*. London: 4Myeloma Press.

Khashei, M. and Bijari, M. (2010). An artificial neural network (p,d,q) model for time series forecasting. *Expert Systems with Applications*, [online] 37(1), pp.479–489. doi:10.1016/j.eswa.2009.05.044.

Khashei, M. and Bijari, M. (2012). A new class of hybrid models for time series forecasting. *Expert Systems with Applications*, 39(4), pp.4344–4357. doi:10.1016/j.eswa.2011.09.157.

Liu, X.-Y., Yang, H., Gao, J. and Wang, C. (2021). FinRL: Deep Reinforcement Learning Framework to Automate Trading in Quantitative Finance. *SSRN Electronic Journal*. doi:10.2139/ssrn.3955949.

Lu, W., Li, J., Li, Y., Sun, A. and Wang, J., 2020. A CNN-LSTM-Based Model to Forecast Stock Prices. *Complexity*, 2020, pp.1-10.

Novotny, P.A. (2018). *KDB+ FOR ELECTRONIC TRADING : q, high frequency financial data and algorithmic trading.*

Sun, J., Xiao, K., Liu, C., Zhou, W. and Xiong, H. (2019). Exploiting intra-day patterns for market shock prediction: A machine learning approach. *Expert Systems with Applications*, 127, pp.272–281. doi:10.1016/j.eswa.2019.03.006.

www.quantmod.com. (n.d.). *quantmod: Quantitative Financial Modelling Framework*. [online] Available at: http://www.quantmod.com/ [Accessed 7 Jul. 2022].

Virgilio, G.P.M. (2019). High-frequency trading: a literature review. *Financial Markets and Portfolio Management*, [online] 33(2), pp.183–208. doi:10.1007/s11408-019-00331-6.

Wellman, M.P. and Rajan, U. (2017). Ethical Issues for Autonomous Trading Agents. *Minds and Machines*, 27(4), pp.609–624. doi:10.1007/s11023-017-9419-4.

Zhang, L., Wu, T., Lahrichi, S., Salas-Flores, C.-G. and Li, J. (2022). A Data Science Pipeline for Algorithmic Trading: A Comparative Study of Applications for Finance and Cryptoeconomics. *arXiv:2206.14932 [cs, econ, q-fin]*. [online] Available at: https://arxiv.org/abs/2206.14932?context=cs [Accessed 7 Jul. 2022].

APPENDICES

Appendix A. ADDITIONAL MATERIAL

A.1 Extracted Data

Figure 2 describes an extract of the extracted data for stock of AAPL.

Date	Open	High	Low	Close	Adj Close	Volume
2022-08-11	170.059998	170.990005	168.190002	168.490005	168.490005	57149200
2022-08-12	169.820007	172.169998	169.399994	172.100006	172.100006	67946400
2022-08-15	171.520004	173.389999	171.350006	173.190002	173.190002	54091700
2022-08-16	172.779999	173.710007	171.660004	173.029999	173.029999	56377100
2022-08-17	172.770004	176.149994	172.570007	174.550003	174.550003	79542000

Figure 2. Data Extracted for AAPL

A.2 Adjusted Close Prices

Figure 3 shows the adjusted close prices during the training period.

Figure 3. AAPL – Adjusted Close Prices

A.3 Differencing

The following figure (Figure 4) shows the differenced close prices for the training period.

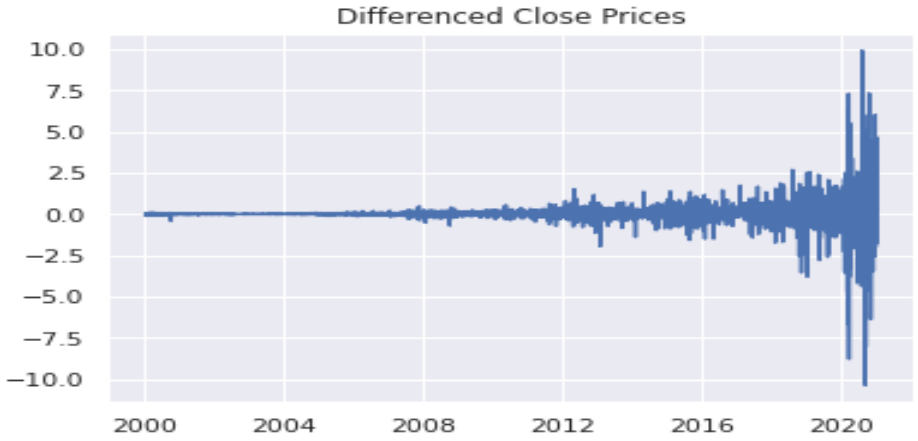

Figure 4. First Order Differencing

A.4 Standardization

The following figure (Figure 5) shows the standardized close prices (after differencing) for the training period.

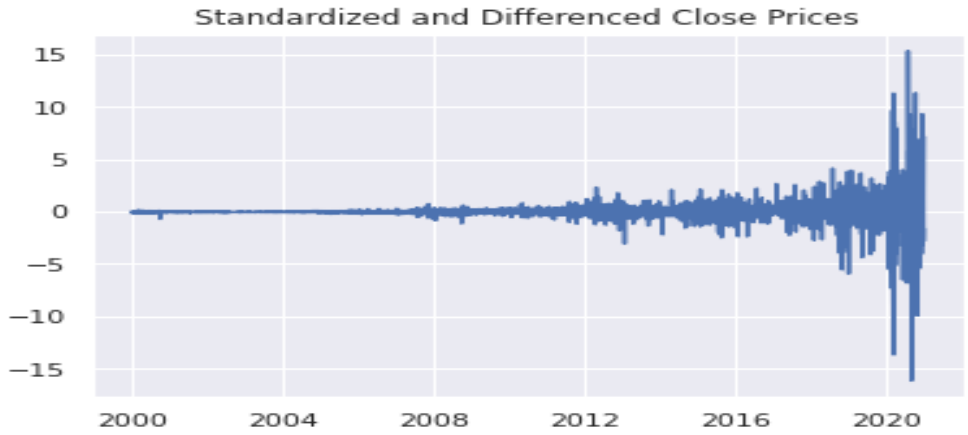

Figure 5. Standardization

A.5 LSTM Model

Figure 6 shows the final model structure for AAPL with 10 neurons in the first LSTM layer and 3 neurons in the second layer.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_10 (LSTM)              (None, 10, 10)            640

 lstm_11 (LSTM)              (None, 3)                 168

 dense_5 (Dense)             (None, 1)                 4

=================================================================
Total params: 812
Trainable params: 812
Non-trainable params: 0
```

Figure 6. Model-AAPL

Figure 7 shows the final model structure for MSFT with 7 neurons in the first LSTM layer and 6 neurons in the second layer.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_10 (LSTM)              (None, 10, 7)             364

 lstm_11 (LSTM)              (None, 6)                 336

 dense_5 (Dense)             (None, 1)                 7

=================================================================
Total params: 707
Trainable params: 707
Non-trainable params: 0
```

Figure 7. Model-MSFT

A.6 Trading with Alpaca

Figure below shows the market orders received in Alpaca by the trading rule for a specific trading day. The final Sell was the exit order by the algorithm for AAPL.

Order History

Asset	Order	Quantity	Average Cost	Notional	Amount	Status
AAPL	Market SELL 08/24/2022 08:17 PM	195	$167.92		$32,744.40	Filled
AAPL	Market BUY 08/24/2022 08:06 PM	5	$167.88		$839.40	Filled
AAPL	Market BUY 08/24/2022 08:06 PM	5	$167.88		$839.40	Filled
AAPL	Market BUY 08/24/2022 08:04 PM	3	$167.84		$503.52	Filled

Figure 8. Market Orders

The figure below shows the growth of the portfolio, during a day of positive return for MSFT.

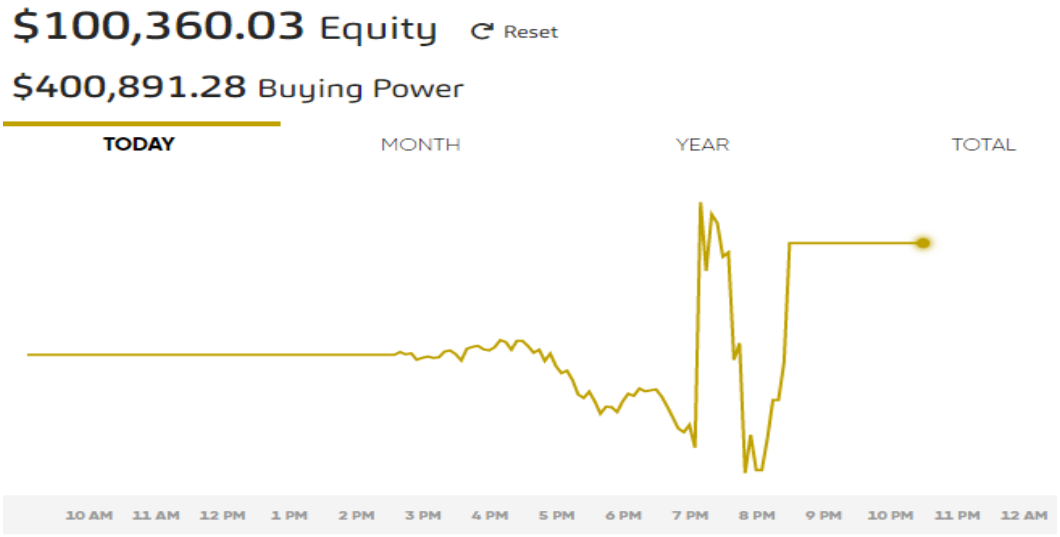

Figure 9. Portfolio Growth

A.7 Original Design

Figure below shows the original design of the proposed implementation.

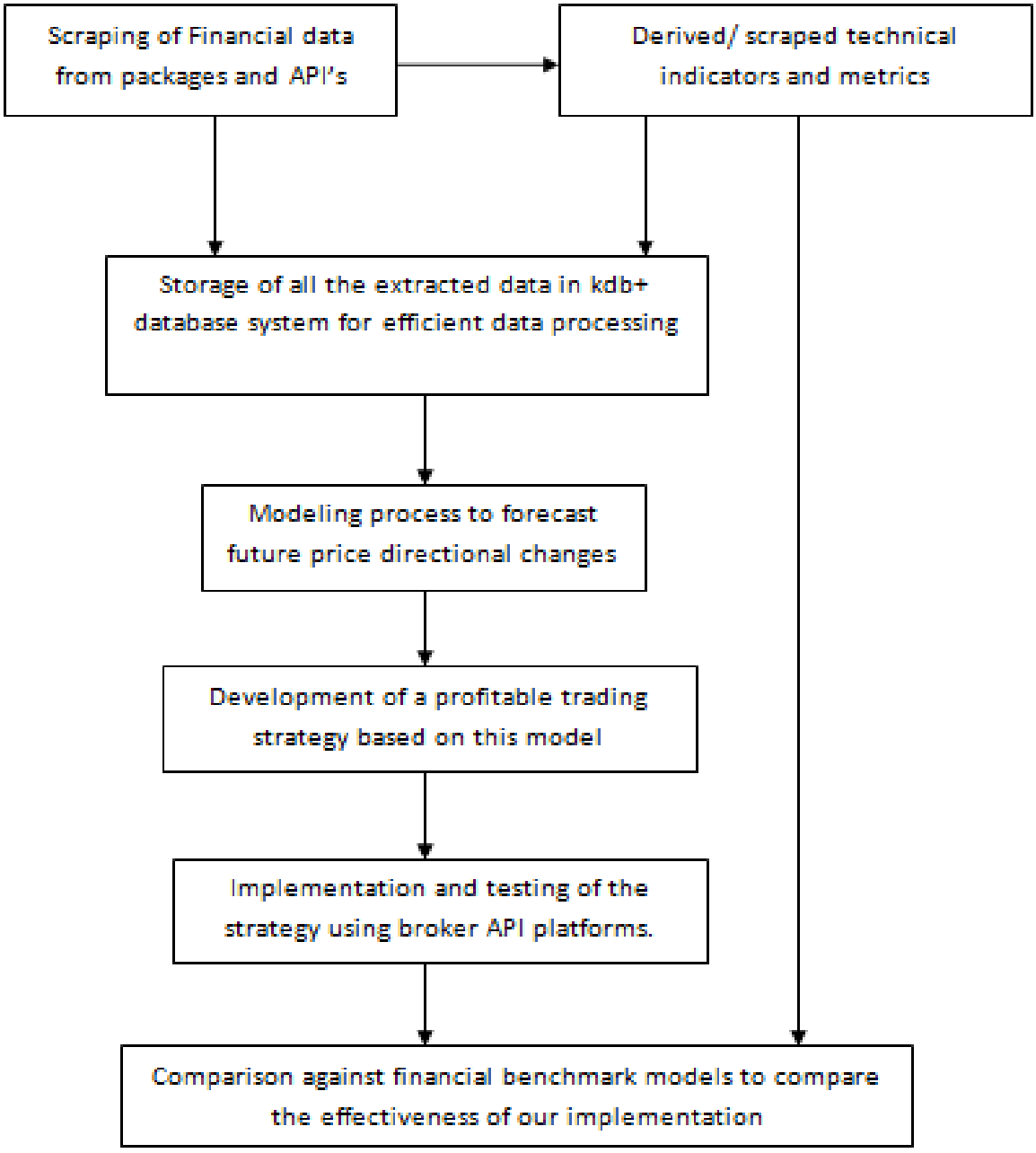

Figure 10. Original Design

A.8 Model Evaluation

Figure below describes a plot of the predicted and actual close price differences (compared to the previous days value) for stock's of MSFT on the test dataset.

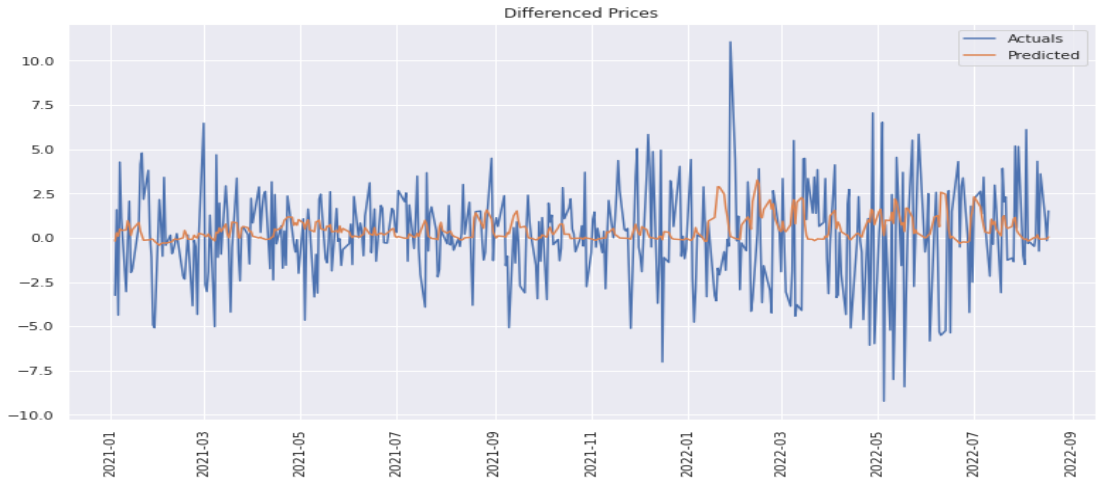

Figure 11. Forecast - Differences

Figure below describes a plot of the predicted and actual close prices for stock's of MSFT on the test dataset.

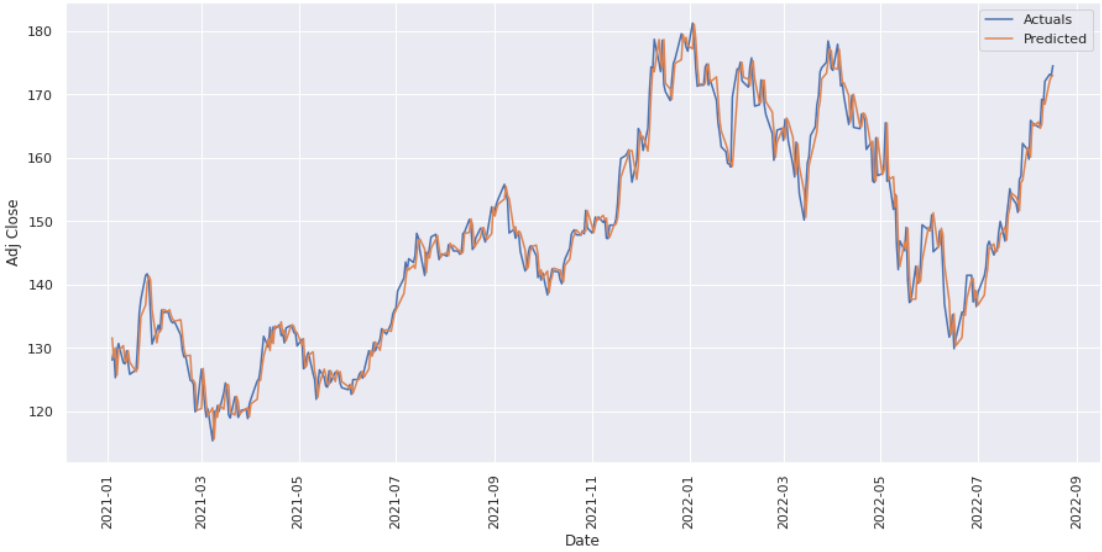

Figure 12. Forecast - Prices