CHARLES ADIKANKWU

CLOSE PRICE PREDICTION OF NASDAQ USING LSTM

MASTER THESIS

UOK- 2023

UNIVERSITY OF KYRENIA

INSTITUTE OF GRADUATE STUDIES

DEPARTMENT OF BIG DATA ANALYTICS

CLOSE PRICE PREDICTION OF NASDAQ USING LSTM

M.Sc. THESIS

Charles ADIKANKWU

Kyrenia

June, 2023

UNIVERSITY OF KYRENIA
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF BIG DATA ANALYTICS

CLOSE PRICE PREDICTION OF NASDAQ USING LSTM

M.Sc. THESIS

Charles ADIKANKWU

Supervisor
Assistant Professor Seda KARATEKE

Kyrenia

June, 2023

DECLARATION

I hereby declare that this is my original work and has never been presented for a degree or any award in any university or any academic institution of higher learning.

It is all the result of my own effort and under the supervision of Yrd. Doç. Dr. Seda KARATEKE

Student

Charles ADIKANKWU

Signature..........................

Date: 31/07/2023

Supervisor

Yrd. Doç. Dr. Seda KARATEKE

Signature..........................

Date: 31/07/2023

APPROVAL

The jury members certify that the study conforms to acceptable standards of scholarly presentation and is fully adequate in scope and quality as a dissertation for the degree of Master of science in **Big Data Analytics**

University of Kyrenia

Academic year: 2022-2023

For jury committee:

Prof. Dr. Metin ZONTUL

Prof. Dr. Semra ERBAŞ

Yrd. Doç. Dr. Seda KARATEKE

Institute of Graduate Studies Director:

Prof. Dr. Candan ÖZOĞUL

Acknowledgments

First and foremost, I am immensely grateful to Almighty God for His guidance, strength, and blessings throughout this journey.

To my highly esteemed advisor Assistant Prof. Seda Karateke for her resilient and tireless effort in seeing me through this fit. A special mention goes to my Head of Department (HOD) Prof. Semra Erbaş and Prof. Metin Zontul who has been with me from the beginning till the end, ensuring that I stay on track and achieve my goals. Their expertise, guidance, and encouragement have been invaluable.

I am indebted to my parents, Sir and Lady P.O Adikankwu, for their unwavering support, love, and sacrifices. Their encouragement and belief in my abilities have been a constant source of inspiration.

I am profoundly grateful to my loving brother, Sylvester Adikankwu, for his continuous financial support. His generosity and belief in my dreams have eased the financial burden and allowed me to focus on my studies.

To my darling wife, Mrs. Scholarstica Charles, words cannot express my appreciation for her prayers, encouragement, and unwavering faith in me, especially during moments when I felt like giving up. Her love and prayers have been my pillar of strength.

I extend my deepest gratitude to my big aunty, Mrs. Rose Edemodu, for her love, concern, and support throughout this academic journey. Her guidance and words of wisdom have been invaluable.

I would like to acknowledge Mr. Declan Emegano for his moral and academic support. His guidance, mentorship, and assistance have been instrumental in shaping my research.

Finally, I would like to express my gratitude to all those whose names I couldn't mention individually but have contributed in various ways to my academic journey. Your support, whether big or small, has made a difference.

Thank you all for being a part of my journey and for helping me realize this significant milestone in my life.

Charles Adikankwu

Abstract

This thesis examines the use of LSTM neural networks to forecast the closing price of the NASDAQ index using historical price data. The study makes use of a dataset that spans many years and includes daily closing prices, volume, and other NASDAQ index elements. LSTM networks are used to forecast future closing prices based on historical data, with an emphasis on a 60-day prediction window. Metrics like R^2, Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) are used for evaluating the performance of the LSTM model. The findings show that the LSTM model predicts future closing prices well, with lower error rates than baseline models.

The research additionally examines at how different input parameters and network structure influence the model's performance. Overall, the study indicates that LSTM networks are beneficial in predicting the price of index and explain their potential for application in financial forecasting applications.

Close price prediction of NASDAQ using LSTM

Adikankwu, Charles

M.Sc., Department of Big Data Analytics

July, 2023, (55) pages

Key Words: Machine Learning (ML), Deep Learning (DL), Long short-term memory (LSTM), Recurrent neural networks (RNNs), Root Mean Square Error, Keras, Tensor flow

Özet

Bu tez, geçmiş fiyat verilerini kullanarak NASDAQ endeksinin kapanış fiyatını tahmin etmek için LSTM sinir ağlarının kullanımını incelemektedir. Çalışma, uzun yıllara yayılan ve günlük kapanış fiyatları, hacim ve diğer NASDAQ endeks öğelerini içeren bir veri setinden yararlanmaktadır. LSTM ağları, 60 günlük bir tahmin penceresine vurgu yaparak, geçmiş verilere dayalı olarak gelecekteki kapanış fiyatlarını tahmin etmek için kullanılır. LSTM modelinin performansını değerlendirmek için R^2, ortalama mutlak yüzde hatası (MAPE) ve kök ortalama karesel hata (RMSE) gibi metrikler kullanılır. Bulgular, LSTM modelinin temel modellere göre daha düşük hata oranlarıyla gelecekteki kapanış fiyatlarını iyi tahmin ettiğini göstermektedir.

Araştırma ayrıca, farklı girdi parametrelerinin ve ağ yapısının modelin performansını nasıl etkilediğini inceler. Genel olarak, çalışma, LSTM ağlarının endeks fiyatını tahmin etmede faydalı olduğunu gösteriyor ve finansal tahmin uygulamalarında uygulama potansiyellerini açıklıyor.

LSTM kullanarak NASDAQ'ın yakın fiyat tahmini

Adikankwu, Charles

Yüksek Lisans, Büyük Veri Analitiği Anabilim Dalı

Temmuz, 2023, (55) sayfa

Anahtar Kelimeler: Makine Öğrenimi (ML), Derin Öğrenme, Uzun Kısa Süreli Bellek, Tekrarlayan Sinir Ağları, Kök Ortalama Kare Hata, Keras, Tensör akışı

Table of Contents

List of Figures

List of Tables

List of Abbreviations

NASDAQ: National Association of Securities Dealers Automated Quotations

AI: Artificial Intelligence

DL: Deep Learning

LSTM: Long Short-Term Memory

ML: Machine Learning

RMSE: Root mean square error

Pd: pandas

Np: numpy

Sns: seaborn

MAPE: Mean absolute percentage error

CNN: Convolutional neural network

RNN: Recurrent neural network

KNN: K-nearest neiggbour

RF: Random Forest

Len; length

SVM: Support vector machine

L-R: Linear regression

Adam: adaptive moment estimation

SL: Supervised learning

R-L: Reinforcement learning

NVB: Naïve Bayes

DT: Decision Tree

GBM: Gradient Boosting Machines

USL: Unsupervised learning

GRU: Gated recurrent unit

No. of neur.: Number of neurons

Chapter I

Introduction

This section represents the establishment the study that outlines the specific problem to tackle, what we tend to achieve, and its applicability and importance therein.

National Association of Securities Dealers Automated Quotations (NASDAQ) is the world's largest electronic exchange for trading securities. Its name comes from the abbreviation for National Association of Securities Dealers Automated Quotations and was later introduced into the security market. Historically, On February 8, 1971, NASDAQ stars its business operation and welcomed the first investors (Nasdaq: 50 Years of Market Innovation | Nasdaq, n.d.). The NASDAQ market is a stock exchange situated in the city of New York, United States. It is the second-parlargest stock industry in the world in terms of the overall value of shares transacted (Kovacs et al., 2021). The stock prices of various firms can be found on Yahoo Finance. Yahoo Finance is a daily stock market update service. In addition, the law of demand along with supply serves as the guiding principle behind the stock market (Subasi et al., 2021).

NASDAQ represents a virtual or e-market that involves the trading of funds such as stocks and commodities. It serves as one of the enormous recognized world stock exchanges, targeting advancement in technology(Nasdaq: 50 Years of Market Innovation | Nasdaq, n.d.).Historically, On February 8, 1971, NASDAQ stars its business operation and welcomed the first investors(Nasdaq: 50 Years of Market Innovation | Nasdaq, n.d.). The stock exchange market has long captured the attention and curiosity of traders, investors as well as scholars. NASDAQ is a famous trading in return for technological advancements equities that has attracted substantial awareness owing to its vitality and possibility of engaging in lucrative trading prospects within various financial marketplaces. (Arashi & Rounaghi, 2022). The precise forecasting of NASDAQ stock prices continues to be an aim for market players, since it may provide an edge over rivals and enhance the decision-making process for investments. Machine learning (ML), deep learning (DL), and, in particular, LSTM (Long Short-Term Memory) have enabled innovations in NASDAQ index of stock remain research and prediction. (Sahu et al., 2023).To better navigate the ever-changing stock market, investors and traders can employ these cutting-edge methods to better understand market dynamics, model temporal dependencies, and generate reliable predictions.

Advancements in ML and DL techniques have opened up new avenues for forecasting financial time series data. One such technique that has shown promise in capturing complex patterns and dependencies in sequential data is the LSTM neural network. LSTM models have demonstrated their effectiveness in a wide range of applications, including natural language processing, speech recognition, and, more recently, financial market prediction(Moghar & Hamiche, 2020). NASDAQ plays a crucial role in the global financial market by providing a platform for companies to list their stocks and for investors to buy and sell those stocks (Mehta et al., 2021a). It primarily focuses on technology, and other growth-oriented sectors, attracting many high-profile companies, including technology giants like Apple, Microsoft, and Amazon.

1.1 Statement of the problem

The present research addresses the difficulty of effectively forecasting the closing price of NASDAQ equities using LSTM neural networks. Notwithstanding advances in the use of machine learning and deep learning methods, the efficacy and limits of LSTM models in predicting the extremely changing and unpredictable character of NASDAQ prices remain unknown and need more research.

- The accessibility and caliber of previous past data
- The LSTM model intricacy and execution,
- The ability of the model's predictions to apply to various time frames and stock market indices
- The model's capacity to adjust itself to real-time fluctuations in markets
- The danger of overfitting and possible discrepancies in selecting a model are all particular problems that must be tackled.

The precision of close price forecasts is essential in determining the course of action of NASDAQ traders, investors, and financial organizations. Inadequate prediction models can result in inefficient investing choices, lost trade possibilities, and substantial financial losses.

1.2 Purpose of the study

- The reason for this thesis is to explore the use of LSTM neural networks in forecasting the closing value of NASDAQ equities. We hope to establish a comprehensive prediction platform that may aid traders, investors, and financial organizations to arrive at knowledgeable choices by using historical price data and the capability of LSTM models.

- Utilizing the previous dataset, assessing the usefulness and accuracy of the LSTM neural network in forecasting the closure price of NASDAQ equities. We want to evaluate the LSTM model's capacity to identify inherent trends, patterns, and fluctuations in markets by training it on a huge volume of historical price data.
- Furthermore, we want to figure out possible limits and places for enhancement with the use of LSTM models to forecast NASDAQ prices. We will study the elements that may impact the model's efficacy, which include accessibility of data, model intricacy, adaptability, real-time versatility, overfitting, and choosing a model bias, using thorough evaluation and validation.

1.3 Research questions

- How reliable is the LSTM in projecting the closing values of the NASDAQ stock market index?
- To what degree can LSTM adequately apprehend the NASDAQ data's intrinsic structures and trends?
- How generalizable is LSTM in forecasting NASDAQ index close prices over diverse periods and market conditions?
- Can LSTM beat conventional forecasting models in predicting NASDAQ index close prices??
- What is the impact of different training and hyperparameter configurations on the efficacy of LSTM?

1.4 Significance of the study

The results of this study would not just add to the current body of literature on stock market prediction, but will also give vital insights into the practical use of LSTM neural networks, particularly for near-price prediction in the framework of the NASDAQ. In addition, the results of the study may have practical consequences for traders, investors, and financial institutions, allowing users to arrive at better-informed trading choices and perhaps increase their earnings.

1.5 Limitation of the study

- Data Accessibility: The accessibility and efficacy of past price data for NASDAQ equities could hinder the analysis. Inadequate or erratic data can have an impact on the preciseness and dependability of the LSTM model's predictions.
- Applicability: The LSTM model's findings and outcomes may vary depending on the set of data, time frame, and NASDAQ stocks investigated. It is critical to assess the model's forecasts' application to various stock indexes, alternative time frames, and volatile market situations.
- Overfitting: Overfitting happens when the LSTM model behaves well in the training data but performs poorly with unknown data. It is critical to tackle the danger of overfitting and use methods like normalization and validation methodologies to overcome this constraint.
- Model Selection Bias: The optimal LSTM model may be chosen subjectively based on the scholar's decisions and inclinations. This bias can have an impact on the outcomes and analysis of the study outcomes, hence it is critical to recognize and resolve this constraint.
- Market Complexity: Economic circumstances, geopolitical events, and investor emotions all have an impact on the NASDAQ market. The LSTM model might fail to account for every one of these intricacies, and its projections may be impacted by elements that aren't specifically included in the model, resulting in results restrictions.
- Future Market Uncertainty: Because of the underlying unpredictability and volatility in the market for stocks, forecasting stock values is fundamentally difficult. The model developed by LSTM may be limited in its ability to reliably anticipate upcoming price changes, particularly during times of market instability or unforeseen situations.
- Model Selection: The selection of the best LSTM neural network model may be subject to researcher bias and may affect the results of the study.

Chapter II

Literature review

The study includes research on the use of LSTM and other similar approaches for financial time series forecasting, with an emphasis on the NASDAQ stock market index.

2.1 LSTM for Financial Time Series Prediction

According to financial historical perspectives, 2018 and 2023 have witnessed a surge in how LSTM is used financially in the prediction of market values. The foundation created earlier in historical times, was utilized by the researchers as they investigated more complex LSTM structures and approaches. For instance, studies by (Che et al., 2018) proposed a deep LSTM model that focuses on the mechanism for stock market forecasting and exhibited an increase in accuracy in comparison to conventional LSTM models (Che et al., 2018).

2.1.1 LSTM-Based Stock Market Prediction:

In LSTM-Based Stock Market Prediction, many scholars such as Zhang et al. (2018) during this era continued to focus on LSTM-based stock market prediction, including the NASDAQ index. they developed a hybrid model combining LSTM with a variational mode decomposition (VMD) technique which captures both long-term (LT) and short-term (ST) dependencies in stock market data. According to the results of their investigation, the hybrid model demonstrated superior performance in terms of both accuracy and stability of prediction. Additionally, (Mehta et al., 2021b) proposed a multi-task LSTM model that, in addition to previous stock data, utilized external elements such as news mood and economic indicators to predict NASDAQ prices. Their research indicated that the incorporation of external information improved the prediction capability of LSTM models.

2.1.2 Comparative Analysis with ARIMA and Other Models:

The purpose of this research was to compare the accuracy of LSTM models in forecasting stock market trends to that of ARIMA and other more conventional models. This study aims to shed light on LSTM models' potential for accurate stock market forecasting and highlight any advantages they may have over more traditional approaches to time series forecasting. These were made evident by the study by, (Kobiela et al., 2022) compared LSTM with

ARIMA and recurrent neural networks (RNNs) for predicting the NASDAQ index. Zhang et al. showed that LSTM outperformed both ARIMA and RNNs, suggesting its superior predictive ability. On top of that, (Rather, 2021) conducted a comparative study between LSTM and deep neural network (DNN) models for NASDAQ prediction. They discovered that LSTM models consistently performed better in terms of accuracy and resilience than DNN models, which confirmed the usefulness of LSTM in stock market forecasting.

2.1.3 Feature Engineering and Data Preprocessing:

To improve the performance of LSTM models, researchers extended their investigations into various feature engineering and data preparation strategies. As a result, (Rather, 2021) proposed a deep learning-based framework that integrated LSTM with variational mode decomposition and empirical mode decomposition to extract meaningful features from stock market data. In comparison to more conventional LSTM models, their findings indicated a considerable improvement in the accuracy of prediction. Moreso, during this period the problem of limited data availability and model overfitting. (Zhao et al., 2022) presented a self-attention mechanism that would be implemented into LSTM models for NASDAQ prediction. This would help address the issues that were previously mentioned. Their approach effectively captured temporal dependencies and mitigated the overfitting issue, resulting in enhanced predictive performance.

2.1.4 Ensemble Approaches:

The term ensemble approaches refer to the utilization of numerous models or methods to enhance the accuracy of predictions or capacity for decision-making in several domains, such as, but not restricted to ML, data extraction, and other forecasting tools. This model-the ensemble was explored as an effective means to improve prediction accuracy. (Tang et al., 2021) developed an ensemble model combining LSTM with wavelet transform and support vector regression for NASDAQ prediction which captured both local and global patterns in the data, leading to improved prediction accuracy. Similarly, according to (Liu & Wang,) 2021) the combination of LSTM with other machine learning methods, such as random forests (RF) and gradient boosting, produces ensemble models. The ensemble models demonstrated enhanced performance compared to individual models, indicating the potential benefits of combining diverse prediction techniques.

Again, in another study by Adesola Adegboyea, Michael Kampouridis (Adegboye & Kampouridis, n.d.) Forex Market Trend Reversal Prediction Using Machine Learning Classifiers and Regression Techniques. According to them, the DC technique is an alternative strategy for data sampling that functions following the occurrence of events. As a result, a one-of-a-kind technique called Directional Change-based methodology makes use of ML algorithms to anticipate reversals in trends. Traders can take preventative actions before a loss takes place, which will, as a result, enhance their bottom lines.

2.2 Artificial Intelligence (AI)

(AI) is used to describe machines that are capable of performing tasks that would normally require cognitive abilities, such as perceiving, reasoning, knowledge acquisition, and making decisions. It spans a variety of subfields, such as ML natural language processing (NLP), image processing, and robots, and it has applications in a variety of industries, including healthcare, finance, transport systems, and more. The adoption of AI must be conducted in a manner that is ethical and responsible because there are several ethical concerns surrounding the technology. These concerns include the loss of jobs, security of data, bias in algorithms, and the effect on autonomy among humans. According to (Liu & Wang, 2021) AI is any computational program that can make decisions, solve challenges, and adapt like a person. Since the 1960s, the use of machines, mathematical methods, and early AI information approaches in healthcare, particularly in clinical decision-making, has been studied. The goal of AI-first was to create robots that can think and reason through complicated problems in the same way that humans do, and so share the same key cognitive qualities. Since then, AI has progressed significantly, the history of AI and its subsets; machine learning and deep learning, is depicted in Figure 1, with AI ideas and implementations progressively becoming a reality in scientific research centers (Huang et al., 2019). In AI, software applications can now imitate the neuronal mechanism of the neocortex in the brain, in which most intellectual capabilities, thinking, and cognitive tasks occur. AI is progressively gaining applicability in health care, because of advances in processing speeds, supercomputers, and improved AI learning algorithms. It was noted in

1959, that LA Samuel, an American developer in the computer games sector and AI, coined the phrase "machine learning(Wiederhold & McCarthy, 2010b)" as a multidisciplinary discipline that employs approaches to enable computer systems to "learn" from a particular data collection without having to be expressly programmed to do so (Wiederhold & McCarthy, 2010a).

The majority of traditional treatments were created with an 'average patient in mind. However, due to the diversity of human pathophysiology, the same fundamental rubrics are frequently insufficient when modeling patients. As a direct consequence of this, the recommendations contained within the guidelines might not be suitable for every patient. Therefore, artificial intelligence techniques such as ML and DL are being applied in the construction of normalized prediction systems that may assist physicians and increase patient-specific decision-making. These, in turn, would assist health practitioners in regaining time and boosting patient-provider interactions (Wiederhold & McCarthy, 2010a)

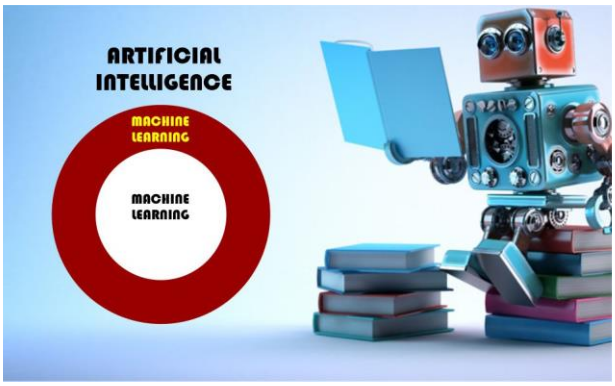


Figure 1: AI, ML, and DL Were Originally Founded in the 1950s, 1980s, and 2010s, respectively (Huang et al., 2019)

2.2.1 Machine Learning (ML)

Historically, Arthur Samuel(Wiederhold & McCarthy, 2010b), an American pioneer in the field of computer games and artificial intelligence, is credited with coining the term "machine learning" in 1959. According to him, ML provides computers the ability to learn without being explicitly taught. Today, machine learning is quite common and is used in a variety of contexts, including prediction, recognition of images and speech, medical diagnosis, and work in the financial industry and trading. Primarily, ML employs mathematical models to analyze data, understand underlying patterns, and provide insights that may be used to make judgments and predictions about real-world occurrences (Huang et al., 2019). Its discipline is concerned with the invention and development of algorithms that help computers improve their performance over time by analyzing data. To learn, the machine must examine its previous experiences for beneficial regularities and patterns. The automatic generation of models, such as rules and patterns, from data, is a key emphasis of machine learning research (Kumbure et al., 2022). ML is based on animal and human learning processes, but it can simultaneously explain how they learn. The main puzzle is how machines learn and why humans need them. According to (Nilsson, 1996) "machine learns anytime it modifies its structure, program, or data in such a way that its upcoming effectiveness increases" in response to its surroundings. In a database, these modifications might take the form of insertions or updates. There is a requirement for a machine that learns since some data are difficult to clarify and illustrating is a better option; the systems must adapt to some conditions or variations that are not projected at the time of their design (Kumbure et al., 2022). Despite comparable mathematical foundations, ML has several benefits over traditional statistics. These merits are seen in decreased feature selection duration, nonparametric and non-linearity relationship interpretation, and loss of data, which is prevalent in numerous imputations(Williamson et al., 2021). Additionally, in ML, the connections between variables might be difficult to decipher, particularly in a black box design (i.e., ANN) (Soni et al., 2022). The time it takes to examine the data is faster with computational power than with the traditional technique. Machine learning can be used to choose features, classify them, or accomplish all of these things simultaneously. Overall, it is categorized into three major learning groups: SL, USL, and R-L (Vijh et al., 2020). Linear and LR, KNN, SVM, NVB, and DT are some of the most often used machine learning techniques.

Meanwhile, because SVM can operate when kernels automatically actualize mapping data to a feature space using non-linear transformations, it is the favored choice in many applications due to its ability to accomplish strongly sparse and noisy data. Furthermore, due to artificial neural network (ANN) black-box problem variable association possibly isn’t easy to interpret, ensemble approaches have lately gained attraction, with Gradient Boosting

Machines GBM and RF emerging as the greatest often used approaches among academics (Roy et al., 2020).

Linear Regression (LR) is frequently used as a benchmark against which other machine learning methods are measured. KNN is one of the most basic ML methods, although it has some drawbacks, including computing capacity in selecting the nearest neighbor, particularly, when dealing with high-dimensional data. Due to various reasons such as data inequity, a small number of variables, changing continuous variables into categorical data, and false-positive and false-negative difficulties, some studies don't depend on accuracy in determining the performance of a mathematical algorithm. As a result, when analyzing and contrasting the effectiveness of various models, the domain in the receiver operating curve is more accurate (T. Singh et al., 2022) Choosing ML methods and orthogonal evidence as well as the efficacy of an ML-based prediction model is determined by aspects to include variations in data, the extent of data, and the breadth of data. Creating data collecting criteria and standardizing ML, as concerns about data security as well as privacy protection, are among the obstacles to ML (T. Singh et al., 2022).

In addition, the following are the three primary types of machine learning, which are classified according to the amount and kind of supervision that the algorithms receive while they are being trained:

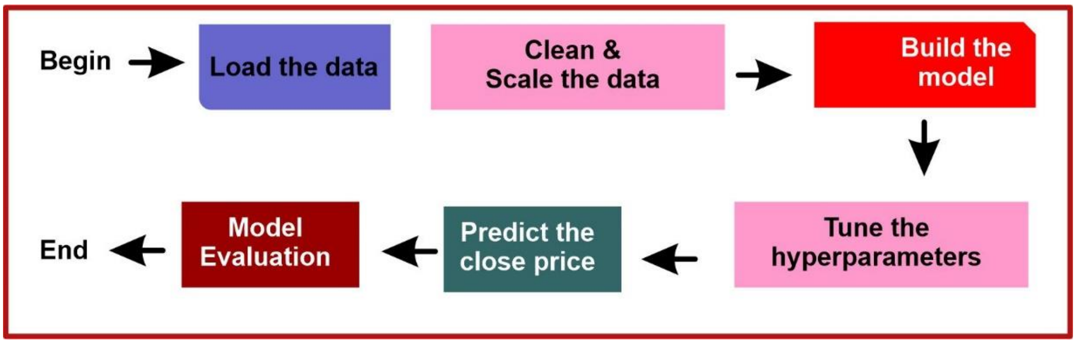


Figure 2: Diagram representing the study architecture

2.2.1.1 Supervised learning

This is a category that includes ML models known as given inputs and outputs, for instance, image and handwriting recognition, as well as the interpretation of ECG data. In this category, an expert utilizes a database to record a collection of variables and the results of those variables. This information is then utilized to develop a prediction model that can classify the results of a set of observations. (V. Singh et al., n.d.). It is crucial to the functioning of both artificial neural networks as well as many biological networks. Additionally, supervised learning has a broad range of applications and is capable of resolving any issue. It is necessary to manually add labels to assist the computer in achieving the desired results in supervised learning. As a result, it has obvious implications for clinical directing, and it can be applied to financial trends.

2.2.1.2 Unsupervised learning

Unsupervised learning focuses on the fundamental trends and correlations in the data. Clustering, principal component analysis, and self-organizing maps are all examples of types of learning that occur without supervision. (Bhandari et al., 2022). Since no label is provided for the data set, the machine must determine what it should be called. In addition, unsupervised learning algorithms have been effectively used in identifying Manipulations of Stock Prices Through the Use of Deep Unsupervised Learning (*IEEE Xplore Full-Text PDF:* n.d.-a):

2.2.1.3 Reinforcement learning

The field of machine learning known as reinforcement learning helps agents improve their trading decisions by taking into account past stock data, market trends, and feedback signals. The purpose of this learning category is to maximize the reward for the proposed algorithm throughout the learning procedure, not to achieve the intended goal (Bhandari et al., 2022).

2.2.2 Deep Learning

Walter Pitts and Warren McCullough of the University of Chicago proposed deep learning (DL) in 1944, and eventually transferred to the Massachusetts Institute of Technology

(Wiederhold & McCarthy, 2010a) DL is a branch of machine learning in which a connection involving a collection of inputs and outputs is learned using neural networks with multiple layers. The network is trained by exposing it to a collection of input data and output categories, and the model develops specific parameters by implementing and changing network weights to reduce an error function till the model outputs near the real data values as feasible. Deep neural networks' power comes from their capacity to find unique associations in data without relying on human-selected characteristics(Shen & Shafiq, 2020a). DL isn't a stand-alone technique. The training of a deep neural network may be done in two ways: supervised and unsupervised learning strategies. Specific learning techniques such as the residual network, have emerged as a result of the field's rapid growth in recent years. As a result, DL is increasingly being viewed as a stand-alone learning strategy. AI is implemented via ML, while ML is accomplished using DL. DL has the following drawbacks (Tang et al., 2021)

1) To produce an accurate model, DL models need a lot of training data. Certain biological samples, on the other hand, may only be found in limited numbers in the actual world.
2) Traditional and basic ML approaches may be utilized to tackle issues in some sectors, with no need for complicated DL approaches.

Deep learning networks involve a hierarchical illustration of data in multilayered connections, where each layer is a high-level abstraction of the interpretation from the preceding stage of the neural networks. Machine-learning methodologies most often demand structured data. As a result, DL models are well-matched for picture reasoning and interpretation, complicated image analysis, and sound and voice sample identification (Wang et al., 2021)

Chapter III

Methodology

This study involved the development and training of an LSTM simple neural model for forecasting to predict the end-of-day price of NASDAQ using the past 60-day stock price and historical trading samples obtained from the financial data platform Yahoo Finance. The trained model was specifically for Nasdaq stock, and the performance of the model was visualized to evaluate its accuracy.

3.1 COLLECTION OF DATA

The data utilized in this research was gathered from the Yahoo Finance platform https://finance.yahoo.com/quote/%5EIXIC/history?period1=1365552000&period2=1681084800&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true, It's a portal that offers news concerning finance, statistics, and analytics. It also provides a comprehensive past financial dataset for stocks, bonds, commodities, and other financial instruments. For every trading session, the dataset includes information that includes starting and closing prices, volume, and other financial measures. The gathering of data is available for download at no cost in a variety of forms, including CSV, JSON, and XML.

The obtained dataset covers ten (10) years (10^{th} April 2013 – 10^{th} April 2023). It has a total of two thousand five hundred seventeen (2517) samples and seven (7) characteristics.

The first stage of data collecting is identifying equities to be picked from each market segment. As a result, the NASDAQ dataset was used to indicate the exchanged item in this investigation. (Pan & Mishra, 2016). define it as a stock exchange based in the United States. NASDAQ was established in 1971 and is widely regarded as the first computerized stock exchange. It is a major participant in the global financial market and a vital forum for corporations seeking to obtain funds through stock issues. Because of its emphasis on innovative technology, it has become a prominent exchange for technology firms and an icon of the tech-driven economy. (Anagnostopoulos & Rizeq, 2021)

NASDAQ offers a market mechanism based on trading through a network of market makers instead of a centralized exchange (Musciotto et al., n.d.).

This allows for faster and more efficient trading, as well as increased competition and lower trading costs. The second phase of the collection of data involves looking out for the specific type of financial data that is required for the dataset. The third phase of collection of data entails acquiring historical stock data from a suitable source (*NASDAQ Composite (^IXIC) Historical Data - Yahoo Finance,* n.d.). As outlined in this chapter, the Long-Short Term Memory (LSTM) network being utilized will undergo training and testing. The ten-year data is saved locally in CSV format. Lastly, the data cleaning was carried out to address any missing values that may arise due to holidays or instances when the stock market is closed on a trading day.

The table below shows the data set of the first 100 rows out of 2517 rows of the data set

Table 1: NASDAQ Data set of first 100 samples

Date	Open price	High price	Lowest value	Closing Price	Adjusted Close	Volume
4/10/2013	3246.06	3299.16	3245.8	3297.25	3297.25	1769870000
4/11/2013	3289.59	3306.95	3287.74	3300.16	3300.16	1829170000
4/12/2013	3292.39	3296.5	3271.02	3294.95	3294.95	1471180000
4/15/2013	3277.58	3283.4	3213.46	3216.49	3216.49	1779320000
4/16/2013	3239.05	3265.84	3231.45	3264.63	3264.63	1515400000
4/17/2013	3236.25	3236.98	3186.08	3204.67	3204.67	1902730000
4/18/2013	3212.24	3212.97	3154.96	3166.36	3166.36	1766000000
4/19/2013	3169.32	3210.03	3168.33	3206.06	3206.06	1738850000
4/22/2013	3217.4	3241.16	3198.74	3233.55	3233.55	1628340000
4/23/2013	3252.8	3275.89	3241.52	3269.33	3269.33	1684770000
4/24/2013	3262.21	3277.12	3255.44	3269.65	3269.65	1738590000
4/25/2013	3279.82	3301.28	3279.29	3289.99	3289.99	2012230000
4/26/2013	3284.07	3287.48	3268.03	3279.26	3279.26	1721970000
4/29/2013	3290.31	3315.33	3289.42	3307.02	3307.02	1594110000
4/30/2013	3308.05	3328.79	3298.58	3328.79	3328.79	1984270000
5/1/2013	3325.35	3330.02	3296.5	3299.13	3299.13	1884600000
5/2/2013	3306.15	3344.9	3305.81	3340.62	3340.62	1757480000
5/3/2013	3371.41	3388.12	3370.3	3378.63	3378.63	1745570000
5/6/2013	3382.33	3396.21	3381.44	3392.97	3392.97	1500410000
5/7/2013	3398.84	3402.24	3381.04	3396.63	3396.63	1709800000
5/8/2013	3394.89	3413.27	3389.8	3413.27	3413.27	1756400000
5/9/2013	3408.94	3428.54	3403.43	3409.17	3409.17	1826220000
5/10/2013	3414.84	3436.6	3411.59	3436.58	3436.58	1689730000
5/13/2013	3429.53	3447.1	3426.67	3438.79	3438.79	1615510000
5/14/2013	3439.72	3468.67	3439.72	3462.61	3462.61	1820520000

5/15/2013	3455.67	3475.48	3452.31	3471.62	3471.62	1843910000
5/16/2013	3473.16	3485.95	3462.24	3465.24	3465.24	1945760000
5/17/2013	3483.41	3499.2	3473.04	3498.97	3498.97	1828610000
5/20/2013	3490.46	3509.41	3488.13	3496.43	3496.43	1745260000
5/21/2013	3495.46	3512.15	3486.88	3502.12	3502.12	1776780000
5/22/2013	3503.48	3532.04	3446.96	3463.3	3463.3	2179330000
5/23/2013	3426.07	3467.13	3422.51	3459.42	3459.42	1820670000
5/24/2013	3438.28	3459.47	3429.31	3459.14	3459.14	1449210000
5/28/2013	3497.9	3514.8	3475.39	3488.89	3488.89	1748070000
5/29/2013	3471.67	3479.53	3450.4	3467.52	3467.52	1794650000
5/30/2013	3473.21	3503.82	3473.04	3491.3	3491.3	1737320000
5/31/2013	3478.22	3500.67	3455.84	3455.91	3455.91	1968270000
6/3/2013	3460.76	3465.84	3419.39	3465.37	3465.37	2054100000
6/4/2013	3467.02	3482.75	3430.02	3445.26	3445.26	1871640000
6/5/2013	3432.85	3446.15	3397.91	3401.48	3401.48	1813890000
6/6/2013	3404.41	3424.05	3378.24	3424.05	3424.05	1802700000
6/7/2013	3437.84	3471.73	3429.43	3469.22	3469.22	1646810000
6/10/2013	3475.68	3484.81	3465.54	3473.77	3473.77	1556520000
6/11/2013	3436.62	3466.57	3426.57	3436.95	3436.95	1560370000
6/12/2013	3458.14	3459.18	3395.91	3400.43	3400.43	1630200000
6/13/2013	3398.54	3451.03	3387.61	3445.36	3445.36	1584740000
6/14/2013	3442.31	3448.4	3419.32	3423.56	3423.56	1458030000
6/17/2013	3449.97	3468.56	3436.34	3452.13	3452.13	1581830000
6/18/2013	3456.29	3488.31	3456.09	3482.18	3482.18	1675090000
6/19/2013	3483.59	3485.45	3443.2	3443.2	3443.2	1649200000
6/20/2013	3405.14	3412.94	3355.93	3364.64	3364.64	2041500000
6/21/2013	3367.81	3377.3	3326.86	3357.25	3357.25	2921900000
6/24/2013	3326.38	3344.66	3294.95	3320.76	3320.76	2030960000
6/25/2013	3350.59	3358.31	3327.69	3347.89	3347.89	1657280000
6/26/2013	3375.7	3383.7	3365.48	3376.22	3376.22	1671280000
6/27/2013	3395.79	3412.79	3395.41	3401.86	3401.86	1689800000
6/28/2013	3389.3	3422.2	3382.75	3403.25	3403.25	3630410000
7/1/2013	3430.48	3454.43	3430.31	3434.49	3434.49	1586750000
7/2/2013	3430.69	3453.29	3415.23	3433.4	3433.4	1685190000
7/3/2013	3420.27	3455.42	3417.88	3443.67	3443.67	935980000
7/5/2013	3468.48	3479.46	3441.78	3479.38	3479.38	1254400000
7/8/2013	3493.81	3495.51	3475.39	3484.83	3484.83	1521720000
7/9/2013	3501.25	3508.81	3484.79	3504.26	3504.26	1633520000
7/10/2013	3502.11	3522.99	3502	3520.76	3520.76	1567340000
7/11/2013	3557.79	3579.29	3552.52	3578.3	3578.3	1744210000
7/12/2013	3579.58	3600.08	3576.57	3600.08	3600.08	1615820000

7/15/2013	3601.09	3609.59	3591.54	3607.49	3607.49	1449130000
7/16/2013	3611	3611.35	3589.65	3598.5	3598.5	1590540000
7/17/2013	3608.13	3615.79	3600.69	3610	3610	1564340000
7/18/2013	3610.03	3624.54	3607.09	3611.28	3611.28	1719390000
7/19/2013	3581.9	3589.05	3578.57	3587.61	3587.61	1785460000
7/22/2013	3599.87	3601.92	3587.46	3600.39	3600.39	1507010000
7/23/2013	3606.7	3606.7	3576.96	3579.27	3579.27	1620350000
7/24/2013	3605.26	3606.28	3573.53	3579.6	3579.6	1856660000
7/25/2013	3589.46	3606.19	3579.2	3605.19	3605.19	2203970000
7/26/2013	3584.85	3613.33	3581.26	3613.16	3613.16	1796060000
7/29/2013	3604.29	3618.86	3592.8	3599.14	3599.14	1545720000
7/30/2013	3612.36	3629.12	3606.33	3616.47	3616.47	1763580000
7/31/2013	3627.66	3649.35	3624.77	3626.37	3626.37	1942380000
8/1/2013	3654.18	3678.5	3653.74	3675.74	3675.74	1863290000
8/2/2013	3671.11	3689.59	3663.88	3689.59	3689.59	1683270000
8/5/2013	3682.67	3694.19	3681.34	3692.95	3692.95	1471860000
8/6/2013	3685.39	3690.32	3654.67	3665.77	3665.77	1444200000
8/7/2013	3658.53	3663.2	3633.59	3654.01	3654.01	1659780000
8/8/2013	3672.21	3675.71	3649.64	3669.12	3669.12	1702950000
8/9/2013	3664.27	3677.83	3649.69	3660.11	3660.11	1546570000
8/12/2013	3645.78	3673.51	3645.39	3669.95	3669.95	1422420000
8/13/2013	3675.37	3691.06	3648.82	3684.44	3684.44	1644730000
8/14/2013	3683.97	3686.55	3668.74	3669.27	3669.27	1589370000
8/15/2013	3625.36	3626.77	3600.96	3606.12	3606.12	1742510000
8/16/2013	3603.78	3621.46	3598.65	3602.78	3602.78	1520430000
8/19/2013	3601.88	3623.48	3589.03	3589.09	3589.09	1381050000
8/20/2013	3596.77	3625.26	3593.14	3613.59	3613.59	1308280000
8/21/2013	3603.68	3630.23	3589.02	3599.79	3599.79	1438510000
8/22/2013	3614.14	3639.21	3613.93	3638.71	3638.71	927400000
8/23/2013	3659.21	3660.66	3643.86	3657.79	3657.79	1499890000
8/26/2013	3661.81	3684.22	3652.26	3657.57	3657.57	1404230000
8/27/2013	3616.06	3629.95	3573.57	3578.52	3578.52	1640040000
8/28/2013	3579.11	3607.36	3578.8	3593.35	3593.35	1370650000

3.2 PREPROCESSING OF DATA

Data preprocessing is a critical step in data science that involves cleaning and transforming raw data into a format suitable for analysis. In general, raw data may contain errors, inconsistencies, missing values, or other issues that can make it difficult to analyze. Data preprocessing involves identifying and addressing these issues so that the data can be used effectively in analysis (Mishra et al., 2020)

Once data has been collected, it must undergo preprocessing before training. One such preprocessing task is feature selection(Shen & Shafiq, 2020b).

Table 2: first five rows of the target samples for prediction

Date	Close
2013-04-10	3297.50000
2013-04-11	3300.1599
2013-04-12	3294.949951
2013-04-15	3264.629883
2013-04-16	3264.629883

In this study, the price at which it closes is selected as the feature. Given that the range of closing prices can significantly vary in NASDAQ, the prices at which it closes are normalized between 0 and 1 to standardize the range

```
# scale the data
scaler = MinMaxScaler(feature_range=(0,1))
#transforming our data based on two input values (0 and 1) above this code
scaled_data = scaler.fit_transform(dataset)

scaled_data
```

```
array([[0.01015352],
       [0.01037925],
       [0.0099751 ],
       ...,
       [0.69505191],
       [0.68500855],
       [0.69207542]])
```

Figure 3: Source code for normalized data process

3.3 MODEL BUILDING AND TRAINING

The study utilized Python 3.10 and Jupyter Notebook to create and train the LSTM network. Jupyter Notebook is a web application that enables the writing of Python notebooks in the browser. The aim was to make the study replicable by others when pushed to any version control system. In addition, the open-source deep-learning libraries Tensorflow and Keras were used to create and train the neural network. These libraries provide a Python API that enables the creation and training of neural networks and includes implementations of loss functions and optimization algorithms.

The LSTM network had five inputs, each consisting of a vector of the most recent 60 days' open, close, low, high, and volume. The network comprised two hidden layers and a hidden state with 50 features. Additionally, it had a single output since the model's task was to perform analysis, which is a single scalar value between zero and one.

During the training of the model, the study employed an implementation of the Adam optimization algorithm. The stock's training sample was used to train the network using gradient descent with an epoch set to one.

The chosen configurations and hyperparameters are used to compile the LSTM model. The Keras summary () function is used to print the model’s summary representation, After compiling the model, one trains the model with the training datasets and then makes predictions.

```
#create a new data frame with only the close column
close_data_df = df1.filter(['Close'])
#converting the data frame to a numpy array
dataset = close_data_df.values
#get the number of rows to train the model on
training_data_len = math.ceil(len(close_data_df) * .8)

print(training_data_len)
2014

#convert the x_train and y_trainto numpy arrays to use them in training LSTM models
*x-train, y-train =numpy.array(x-train), numpyp.array(y-train)
#Reshape x_train the data because the LSTM network expects the input to be 3D in the
form of no. of samples, no. of time steps, and number of features
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
x_train.shape
```

(1954, 60, 1)

```
# Build the LSTM model
model = Sequential()
#Add LSTM layers to our model
model.add(LSTM(30, return_sequences=False, input_shape=(x_train.shape[1], 1)))
# model.add(LSTM(50, return_sequences=False))
# model.add(Dense(25))
model.add(Dense(1))

#model compilation
model.compile(optimizer='adam', loss='mn_squared_error')
#An optimizer is used to improve upon the loss function while the loss function is used to measure how well the model did in training

#Train the model
model.fit(x-train, y-train, batch-size=128, epoch=50) #fit is another name for train
#batch size is the total no. of training present in a single batch
#epoch is the no. of iterations when an entire dataset is passed forward and backward through a neural network
#Create the testing data set
```

Figure 4: source code of splitting, building, and model training

3.4 Deep learning model (LSTM) employed

Deep learning is a type of machine learning that involves training artificial neural networks on large amounts of data to automatically identify patterns and make decisions. LSTM is a specific type of RNN designed to deal with sequences of data by allowing the network to learn long-term dependencies in the data. LSTMs achieve this by using memory cells that can hold data for an extended period and gates that regulate the dissemination of data into and out of these cells. As a result, LSTMs can be used for a variety of duties such as language modeling, voice recognition, and sentiment analysis. (Bengio et al., 2021)

- **Where LSTMs are used in DL**

  LSTMs are widely used in deep learning for a variety of tasks, including but not limited to:
- **Natural Language Processing (NLP)**

  LSTMs are commonly utilized in applications such as language modeling, machine interpretation, sentiment analysis, and text categorization.
- **Speech Recognition**

  LSTMs can be used for speech recognition by learning to transcribe speech into text.
- **Time Series Forecasting**

  Time Series prediction with LSTMs refers to using LSTMs for predicting future values in time-series data. Time-series data is a collection of values that are recorded at regular intervals over time, such as stock prices, weather data, or sales data. The purpose of time series prediction is to utilize prior data to create forecasts about forthcoming events.

  Due to LSTMs having the capacity to understand and recall associations among occurrences or details that happen distantly in time while having an influence on each other in sequential data, one possible use case for them is in predicting future values of a time series, commonly known as time series forecasting. In the instance of stock prices, for instance, the LSTM can gain insight into patterns in the data such as long-term trends or seasonality, and use this knowledge to forecast future values. The data is supplied into the network one time step at a time in time series prediction using LSTMs, and its network utilizes this data to predict the next time step for the present time step. This procedure keeps happening for every data point, permitting the network to develop a long-term knowledge of the data and produce precise predictions.

3.5 LSTM Recurrent Neural Network and Its learning Ability

LSTM is a sort of RNN which is frequently utilized for sequence modeling tasks. Techniques such as speech recognition, language translation, and sequential data analysis can be applied using this design. LSTMs are intended to solve the disappearing and expanding gradient issues that plague standard RNNs. Because RNNs employ the same set of weights for all time steps, gradients can either disappear (i.e., become extremely little) or explode (i.e., become very big) during backpropagation. LSTMs, on the other hand, employ a more complicated design that comprises a memory cell, an input gate, a forget gate, and an output gate. The memory cell is in charge of retaining data throughout a period whereas the gates govern the passage of data within and outside of the cell. This enables LSTMs to deliberately select what data to preserve or dismiss when required, resulting in them being well-suited for tasks involving the capture of long-term connections in sequential data, such as dependencies across time, where modeling is critical.

LSTMs have been proven to be successful for an extensive number of sequence modeling applications in general, and they are widely utilized in both research and industry.

RNNs are a sort of neural network intended to analyze data that is repetitive, otherwise known as sequential data which includes time series or natural language text. RNNs can keep an internal state or memory that stores details from past time steps, allowing them to simulate temporal relationships in data.

However, the vanishing gradient problem has typically been a barrier for RNNs, as gradients grow tiny in size making it impossible for the network to learn long-term relationships. The network works by sending gradients backward in time and multiplying them using the same weights at every node. This could cause the gradient to become infinitely small or big, hindering the network's ability to learn.

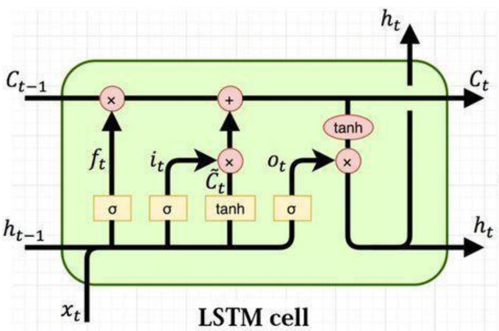
The intricate framework is made up of cell state, input activation function, reset gate, and output gate used by LSTM networks to overcome this difficulty. Long-term memory is maintained by the cell state, while information entry and exit are controlled by gates.

3.5.1 Each of the elements of the LSTM architecture is explained briefly below:

- **Memory Cell:** This serves as a storehouse for information from previous time steps, which may subsequently be passed on to later time steps.
- **Input Gate**: This gate regulates the data flow from the current input to the memory cell, determining which information is deemed significant enough to be retained in the memory cell.

- **Forget Gate:** This gate governs the transfer of information from the previous memory cell to the current time step, determining which data is no longer pertinent and should be discarded.
- **Output Gate`:** By controlling which bits from the memory cell are sent on to the LSTM's output at any given time step, this gate plays a crucial role in the network's overall operation.

  During training, the LSTM network figures out how to adjust the contents of a memory cell in response to new information and the network's stored history. LSTM is a type of recurrent neural network that differs from traditional RNNs. and can comprehend the sequence-dependent relationships among the items in a sequence It is built particularly for dealing with time series data. LSTMs have an edge over traditional RNNs in that they can successfully solve exploding and vanishing gradient concerns.



$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$
$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$
$$h_t = \tanh(C_t) * o_t$$

Figure 5. Structure of LSTM

3.5.2 Disadvantages

The learning process poses the challenge of achieving a suitable equilibrium between underfitting and overfitting. When the model is too basic to grasp the intricacy of the data, underfitting happens, while overfitting happens when a model is overly complicated and matches the training data too precisely, whereas underfitting arises when the model is too simple and fails to capture the underlying patterns in the data, leading to poor performance on new data. To address this, techniques such as regularization, early stopping, and cross-validation can be used. Overall, the learning ability of LSTM involves adjusting the weights and biases of the network to improve its ability to predict output based on input data, while also avoiding issues such as the vanishing gradient problem and overfitting. Additionally, predicting the price of stock trends is a time-series challenge as it involves predicting future values of stocks determined by preceding stock prices in a sequential time-series fashion. For instance, if a stock was priced at $700 three days ago, it is improbable that the stock price today will be $200; thus, the sequence of information is crucial in this context. Traditional feed-forward neural networks handle each input as a distinct entity without carrying any states between inputs(Poornima & Pushpalatha, 2019). As a result, they are unable to retain patterns over many time steps.

RNN is a more effective method for solving time-series tasks because it contains a response mechanism that preserves sequential information and shares features that have been learned from various inputs (Poornima & Pushpalatha, 2019)

The result of a layer is looped back or transmitted again to the same layer and then fed into the subsequent input. LSTM is a modified version of a recurrent neural network capable of capturing and understanding long-term relationships across any time frame. LSTM units have memory cells that are used to store, maintain, read, and write information (Staudemeyer & Morris, 2019) (Butt et al., 2021). Gates within memory cells are used to control how information flows by taking values from the past to process later inputs. In this project, the LSTM network is employed to forecast stock price movements.

3.6 WHAT IS RNNs

RNNs constitute a type of neural network that is intended for analyzing data in sequence, which includes time series or natural language text. unlike standard feedforward neural networks, which receive a fixed-size input and create a fixed-size result, RNNs may accept arbitrary-length inputs and generate arbitrary-length outcomes. The capacity of RNNs to keep an internal state or memory that collects information from past time steps is a critical characteristic. This enables them to represent temporal relationships in data, which is necessary for numerous practical uses.

3.6.1 Here is a brief overview of how RNNs work:

The RNN accepts an input value x(t) and provides a result h(t) as well as an updated hidden state h(t+1) for each time-point t.

A vector symbolizes the hidden state h(t), which grasps Data from the previous period and is updated using a set of weights and biases distributed across every time-steps.

The output h(t) might be utilized for forecasting or given back into the network as the following time-steps input.

The RNN is enhanced during training to minimize a loss function that assesses the disparity between the expected and true results. This is accomplished by the use of temporal backpropagation, a version of the backpropagation method that caters to the network's temporal dependencies.

The vanishing gradient problem presents a challenge for traditional RNNs due to gradients growing so tiny that the network is unable to acquire long-term connections. Gradients travel backward across the network, multiplied by the same set of parameters at each time step. This occurrence, also known as the vanishing gradient problem, can lead gradients to get exceedingly tiny with time, preventing the network from learning properly. To overcome this challenge, many types of RNNs, such as LSTM and GRU networks, have been created. Such networks work well for difficulties needing the capturing of significant interdependencies in sequential data over lengthy intervals.

RNNs are believed to be beneficial for a wide range of sequence modeling uses, such as speech recognition, language translation, and time-series analysis.

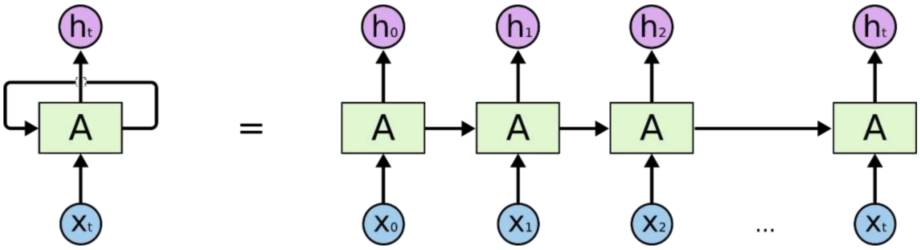


Figure 6. Standard RNN structure shape

3.7 LSTM vs RNN

LSTM and RNN neural networks are designed to handle sequential data. While both types of networks have the same fundamental structure and method of processing sequential input, they vary in their design and capacity to deal with long-term data dependencies.

The following constitute a few important distinctions between LSTM and RNN:

- **Architecture:** LSTM networks employ a more complicated design that consists of a memory cell, an input gate, a forget gate, and an output gate. These gates regulate the passage data within and outside of the memory cell, permitting the network to recall or discard information appropriately over some time. Traditional RNNs, on the other hand, simply feature a single recurrent layer that adjusts the hidden state at each time step.
- **Handling Long-Term Dependencies:** LSTM networks were designed expressly to address the vanishing gradient problem, which occurs when gradients grow too tiny, making it difficult for the network to recognize long-term relationships. The network may dynamically hold or delete data over some time by including memory cells and gates. Because of the vanishing gradient challenge, RNNs may have difficulty with dependency over time.
- **Performance:** In general, LSTM networks outperform RNNs. They nonetheless need greater computing power and could prove more challenging to train.
- **Training:** Both LSTM and RNN networks may be improved using time-based backpropagation, a variation of the backpropagation technique that caters to the network's periodic dependencies. Training LSTM networks, on the other hand, might be challenging considering the increased number of parameters and the necessity of equalizing the memory cell and gates. In summary, LSTM networks are a more advanced sort of recurrent neural network that excels at modeling long-term relationships in sequential data. RNNs, while simpler, can still be successful for certain kinds of sequential data and might prove less difficult to train and utilize.

3.8 Learning Tool Setup/Libraries

The following software and libraries were implemented and set up to create the LSTM architecture for this research:

- **Python 3.10:** a widely-used programming language that is widely used in the field of data science and machine learning.
- **Jupyter Notebook** is an interactive computing environment accessible through the web that enables users to produce, exchange, and modify documents that include executable code, equations, visuals, and descriptive text. It supports more than 40 programming languages, including Python, R, Julia, and Matlab, among others. (Butt et al., 2021)
- **TensorFlow** is a freely available deep learning platform created by Google that is used to build and train ML models. It provides a high-level API for building neural networks and other machine-learning models (Smilkov et al., 2019).
- Keras library: an open-source deep learning library API for Python. It provides a high-level API for building and training neural networks on top of TensorFlow, Theano, and other deep-learning frameworks (Smilkov et al., 2019).
- **sci-kit-learn** library: a popular machine learning library for Python that provides a wide range of tools and algorithms for data preprocessing, feature selection, model selection, and evaluation. It also provides many common machine learning (Géron, n.d.).

Chapter IV

4.1 RESULT

This chapter presents an evaluation of the LSTM recurrent neural network's effectiveness with the selected configurations. The preceding section explains the way the data was transformed into normalized form and how specific measures were considered in reshaping as well as splitting the data into two various sets (training and testing). It is aimed to estimate the price at which the NASDAQ index closes, characterized by complicated, erratic, as well as unstable patterns, presented in Figure 7 below.

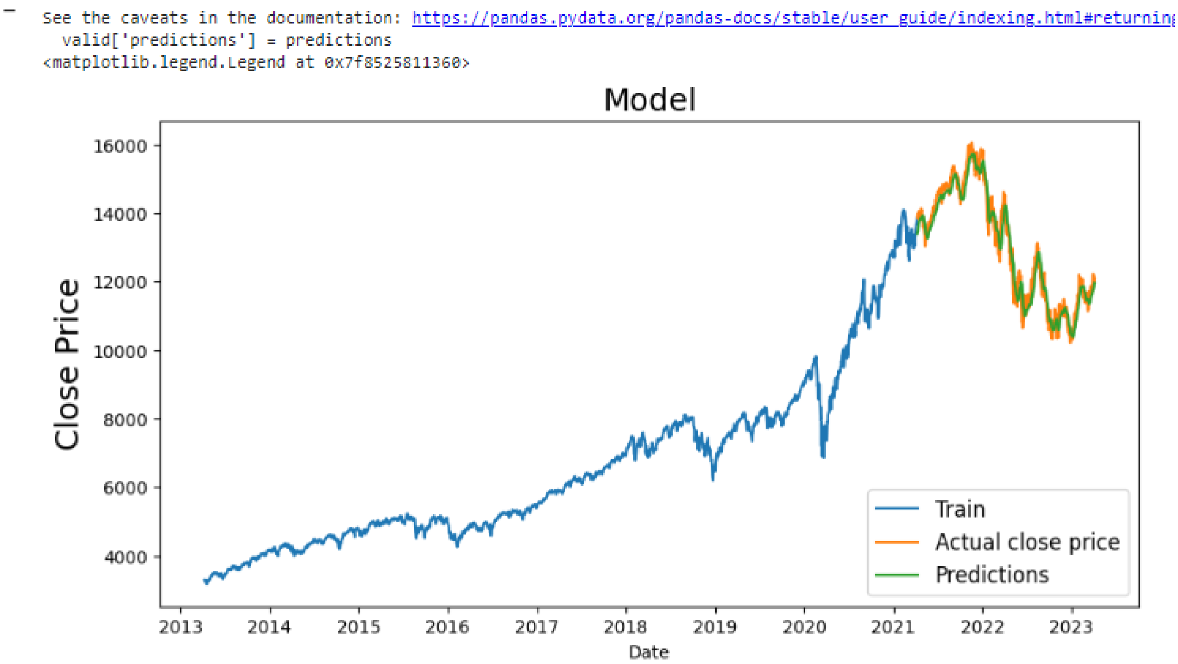


Figure 7: NASDAQ close price along with prediction on the trained data

The blue line on each plot represents the information that was used in training the model, while the orange line represents the actual close price of NASDAQ. The green line shows the predicted price of the stock based on the model's calculations.

The x-axis on the plots represents the data in years, with the data spaced out nicely for clarity. The y-axis displays the closing price value for each day. By visualizing the results in this way, it is easier to see how well the model performed and identify any areas where it may need improvement.

4.2 MODEL ASSESSMENT METRICS

Single-layer and multilayer LSTM architecture was put in place to forecast when the price closed. Numerous ways were taken into consideration bearing several neurons of diverse numbers. The accuracy of the forecast and how reliable the model is considerably evaluated by computing three distinct effectiveness metrics known as RMSE, MAPE, and R. This is analyzed as follows using the formula presented below.

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

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{A_i - F_i}{A_i}\right|$$

R^2 = I - $\frac{SS_{res}}{SS_{tot}}$

Where: $SS_{res} = \sum_i(y_i - f_i)^2 = \sum_i e_i^2$

And $SS_{tot} = \sum_i(y_i - \bar{y}^2$

Where:

y_i = Main time sequence

$\bar{y}_i$ = Estimated time sequence average value

$\hat{y}_i$ = Time series predictions determined by the model,

$\bar{\hat{y}}_i$ = The projected time sequence' average value

n = The total number of observations.

A_i = Actual value

F_i = Forecast value

RMSE is a popular evaluation metric in data science projects, particularly in regression problems where the goal is to predict numerical values. RMSE measures the average distance between the estimated values and the observed values, thus providing an estimate of the model's accuracy. It is beneficial because it assigns greater weight to significant errors than to minor errors., which is especially important in cases where the magnitude of the errors is significant. RMSE is also useful because it is interpretable in the same units as the target variable, making it easier to communicate the model's performance to stakeholders. The RMSE is a useful metric for comparing the relative accuracy of various regression models and the success of various data science methods.

Three prediction metrics are RMSE, MAPE, and R. The square root of the average squared deviation between real and predicted observations is quantified by RMSE. MAPE projects the average magnitude of the inaccuracies relative to the actual data. The coefficient of determination known as R^2 is a statistical metric that quantifies the fraction of the variation explained by an independent variable or factors in a regression model for a dependent variable. It is also known as the fraction of the variation in the dependent variable that can be predicted by the independent variable(s). The performance of the model is better when the values of the RMSE and MAPE are relatively small. A higher value of R^2 shows a stronger resemblance that exists among the predicted and real series. Additionally, scores of performances are computed by reversing the transformation applied to the result realized from scaled data.

Multiple independent executions are performed for each model to account for its stochastic nature. The mean RMSE score from these replicates is used as the main benchmark for selecting the model, subsequently, considering the mean MAPE and R^2 scores as well. The optimal model would be the one with the lowest RMSE and MAPE values, and the highest possible R^2-value.

Table 3: Experimental setting/Environment and Architecture used

Configuration of the machine	Jupyter notebook
Workspace/Libraries	Python 3.10, TensorFlow, and Keras APIs
Framework	LSTM Layers

Table 3 describes the experimental setting employed in this work. The experiment is carried out in Python utilizing the TensorFlow and Keras APIs. It also details the machine setup that was used in all trials.

4.3 Hyperparameter optimization

I employed the Keras tuners python library in the evaluation of my model. To begin, several other libraries such as pandas for reading the dataset, Keras from the TensorFlow, layers, and keras-tuners were employed.

To select the optimal model architecture, a comprehensive exploration of possibilities is conducted. The entire process of model selection is categorized into two main groups: (a) a layered LSTM and (b) a multilayered LSTM. A layered LSTM model is characterized by the inclusion of a solitary LSTM layer, alternatively multilayered LSTM model incorporates multiple LSTM layers within its architecture. Within each of these segments, the procedure is additionally segmented into two stages: (a) fine-tuning hyperparameters, and (b) model training extensively using the corresponding optimized hyperparameters. During the initial stage, the hyperparameters, including the optimizer, epoch, and batch size, are fine-tuned with the help of the evaluation data. In the case of the layered LSTM architecture, several predictive models are examined using distinct configurations. These configurations include the number of neurons in the LSTM layer, such as 12, 30, 50, and 80, as well as the use of Adam as the optimizer. Additionally, the models are trained for varying numbers of epochs, including 50 and 100, and with different batch sizes, specifically 32 and 128. For each combination of hyperparameters, the model is executed multiple times, and the average RMSE score is computed. The best parameter configuration for the respective model is selected in line with the least average root mean square error observed in evaluation data

Table numbered (5) presents the optimal hyperparameters for each of the four-layered LSTM predictive models.

The lowest average RMSE value of 0.6936 in Table 7 is associated with the Adam optimizer using an epoch of 50 and a batch size of 12. The remaining architectures are evaluated, and their validation results are presented in Tables 5-8 in detail. Similarly, the optimal hyperparameters are determined for four distinct multilayer LSTM models featuring different neuron configurations, specifically (12, 6), (30, 15), (50, 20), and (80, 50). In this case, the notation (n1, n2) represents the configuration of the initial and subsequent hidden layers. The same concept applies as the number of unobservable layers appreciates.

4.4 A layered LSTM outcome

In the earlier segment, we explored the procedure of determining the optimal hyperparameters for every individual model using a comprehensive, meticulous, and evidence-based approach. By employing the optimal configuration listed in Table numbered 5 and 6, we proceed to fit each of the four-layered models using the training dataset (which constitutes 80% of the total dataset). Subsequently, we evaluate the mean performance scores on the test dataset to determine the optimal model. To ensure the robustness of the models, each experiment is repeated extensively, with a significant number of replications. The findings indicate that the predictive model featuring 30 neurons, a batch size of 32, and 100 epochs surpasses its counterparts, as it demonstrates the lowest RMSE value, as depicted in Table 6. Hence, the layered LSTM predictive model with 30 neurons has the potential to be regarded as the top-performing option compared to other alternatives

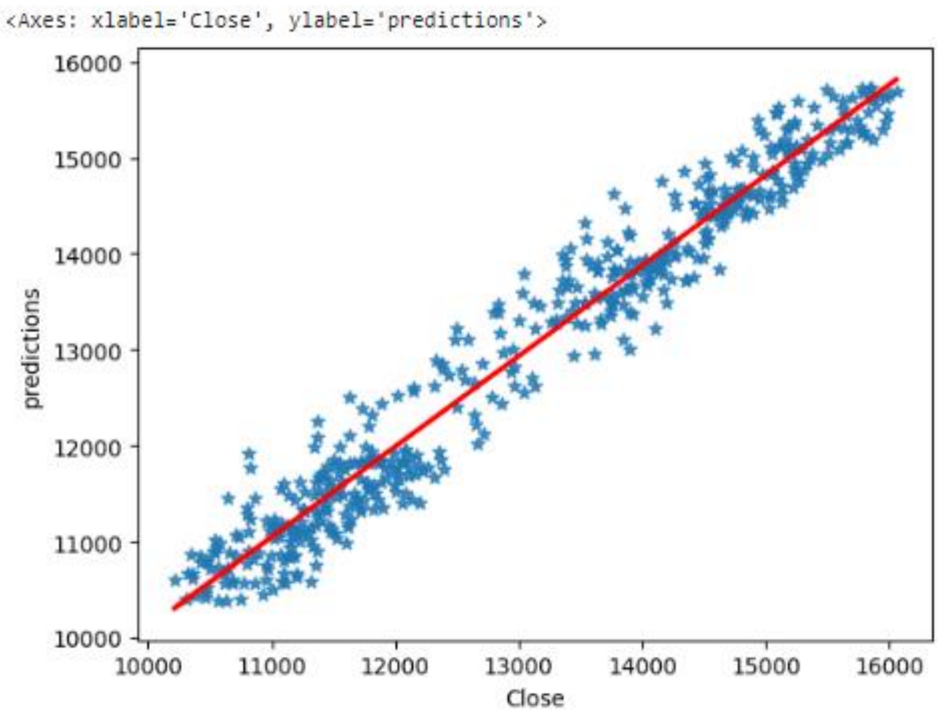


Figure 8. Scatter plot illustrating the relationship between the real and forecasted closing price model

Table 4: Shows the relationship between the real and forecasted price for the test data

Date	Close	Predictions
2013-04-09	13900.190430	13822.927734
2013-04-12	13850.000000	13895.32172
2013-04-13	13996.099609	13894.690430
2013-04-15	13857.839844	14011.43359
2013-04-16	14038.759766	13909.102539

The figures labeled as Figure 7 present a visual representation of the preciseness of the closing price prediction achieved by the best model with 30 neurons. Figure 8 visualization showcases a scatter plot that specifically demonstrates the comparison between the real price and the forecasted prices for the test data. This graph is useful in determining how well the model fits the data. The red dotted line in Figure 8 represents the most appropriate linear relationship (y = x). Although the estimated ending price has slightly diverged away from the actual ending price, this can be attributed to the irregular market conditions caused by different holidays. The proficiency of the predictive model in the test data is also remarkable, as depicted in the aforementioned Figure 7. The model exhibits a good fit even under unconventional market conditions as represented in Table 4 which shows the relationship between the close and predicted price.

Table 5: A record of the optimization parameters for layered LSTM models for 50 epochs

Optimizer	Adam
Epochs	50

No. of neur.	Batch size	RMSE	MAPE	R^2
12	32	32.6426	0.01387	0.7585
	128	24.6113	0.02318	0.6637
30	32	31.1444	0.01710	0.4159
	128	24.7313	0.03574	0.6341
50	32	59.2932	0.01605	0.4655
	128	37.4523	0.03429	0.4956
80	32	124.6219	0.06972	0.3254
	128	25.3061	0.04264	0.5441

Table 6: A record of the optimization parameters for a layered LSTM model for 100 epochs

Optimizer	Adam
Epochs	100

No. of neur.	Batch size	RMSE	MAPE	R^2
12	32	10.7734	0.04923	0.7443
	128	60.1246	0.04507	0.5887
30	32	5.7797	0.03410	0.6188
	128	85.4173	0.02168	0.9418
50	32	54..8548	0.03774	0.6487
	128	32.5335	0.02696	0.2544
80	32	34.1514	0.02572	0.7923
	128	50.7751	0.02958	0.9023

Table 7: A record of the optimal hyperparameters for multi-layer LSTM models for 50 epochs

Optimizer	Adam
Epochs	50

No. of neur.	Batch size	RMSE	MAPE	R^2
(12, 6)	32	85.0829	0.03471	0.7931
	128	56.0019	0.06711	0.9872
(30, 15)	32	16.4976	0.06740	0.8031
	128	78.4024	0.05644	0.6498
(50,20)	32	**0.6936**	0.02107	0.8644
	128	20.5958	0.04403	0.4707
(80, 50)	32	64.5518	0.03836	0.3516
	128	32.8810	0.03874	0.9779

Table 8: A record of the optimal hyperparameters for multi-layer LSTM models for 100 epochs

Optimizer	Adam
Epochs	100

No. of neur.	Batch size	RMSE	MAPE	R^2
(12, 6)	32	42.2317	0.03082	0.5221
	128	31.0906	0.02978	0.2562
(30, 15)	32	31.9910	0.02996	0.8544
	128	29.4023	0.03038	0.2856
(50, 20)	32	88.3632	0.03715	0.4457\
	128	64.5963	0.03115	0.3525
(80, 50)	32	50.2423	0.03463`	0.4485
	128	51.4108	0.03200	0.6355

A Multilayered LSTM outcome

Based on the forecasting outcomes derived from the previously mentioned multi-layer LSTM architecture, it can be inferred that a model with (50, 20) neurons demonstrates remarkable predictive performance. Notwithstanding the impressive performance of the single-layer LSTM model, we seek to explore the potential for further improvements using a multi-layer or stacked LSTM architecture. The main aim of this study is to enhance the precision of the predictions while ensuring the simplicity of the model. Utilizing the optimal hyperparameters outlined in Tables 7 and 8, we proceed to fit the four multilayered LSTM predictive models using the model fitting samples. The selection of the optimal predictive model is based on the typical outcome metrics obtained on the validation dataset.

The values provided in Table 7 for the (50, 20) neuron and epoch 50 configurations represent the assessment measure of the multilayered LSTM models. Furthermore, it indicates that the single-layered predictive model does not enhance the evaluation in comparison to the multi-layered LSTM. This might be due to overtraining which is in other words referred to as overfitting or may be due to some unforeseen intricacies in the model design. The improvement of this model may be subjected to future work.

Chapter V
Conclusion

The objective of this study was to investigate the predictive capability of models trained using an LSTM neural network to anticipate the closing price of NASDAQ indices. After training models for a total of 60 days of trading and assessing their accuracy based on the results depicted in the plot, it was found that the models' predictive ability was insufficient for investing decisions. Therefore, further improvements are required to achieve better accuracy.

The study suggests that before moving on to more challenging tasks of concise and investing-wise predictions, there is a need to improve the ability to predict the closing price. This could serve as a useful guide for other researchers working in this area. However, it should be noted that most people who trade stocks do not succeed, and even professional traders suffer losses. Therefore, a machine learning model's success should not only be measured against average human performance but also against the exceptional few who are successful for non-random reasons.

To create exceptionally performing models, the study proposes taking inspiration from highly skilled human traders who use technical analysis to identify patterns associated with high-magnitude moves in the stock market. The challenge is to train a deep learning model to recognize such patterns accurately enough to be applicable in real life. By imitating and automating the process performed by elite traders, it may be possible to develop more effective models.

Future work

Several improvements can be implemented in this study. One potential avenue for future work involves incorporating additional financial indicators, both fundamental and technical, to enhance the accuracy of predictions. Fundamental analysis focuses on an organization’s effectiveness, while technical analysis evaluates past trends in stock prices. Going further, subject expertise as regards macroeconomic and psychological aspects of trading can also be contemplated during the feature selection process. An additional avenue for enhancing prediction accuracy involves incorporating sentiment analysis on financial news articles, as stock prices can be affected by the tone and sentiment expressed in such news articles.

References

Adegboye, A., & Kampouridis, M. (n.d.). *Machine Learning Classification and Regression Models for Predicting Directional Changes and Trend Reversal in FX Markets*.

Anagnostopoulos, I., & Rizeq, A. (2021). Conventional and neural network target-matching methods dynamics: The information technology mergers and acquisitions market in the USA. *Intelligent Systems in Accounting, Finance, and Management*, *28*(2), 97–118. https://doi.org/10.1002/ISAF.1492

Arashi, M., & Rounaghi, M. M. (2022). Analysis of market efficiency and fractal feature of NASDAQ stock exchange: Time series modeling and forecasting of the stock index using ARMA-GARCH model. *Future Business Journal*, *8*(1). https://doi.org/10.1186/s43093-022-00125-9

Bengio, Y., Lecun, Y., & Hinton, G. (2021). Deep learning for AI. *Communications of the ACM*, *64*(7), 58–65. https://doi.org/10.1145/3448250

Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. C. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, *9*, 100320. https://doi.org/10.1016/j.mlwa.2022.100320

Butt, H., Raza, M. R., Ramzan, M. J., Ali, M. J., & Haris, M. (2021). Attention-Based CNN-RNN Arabic Text Recognition from Natural Scene Images. *Forecasting*, *3*(3), 520–540. https://doi.org/10.3390/forecast3030033

Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, *8*(1), 1–12. https://doi.org/10.1038/s41598-018-24271-9

Géron, A. (n.d.). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*.

Huang, S., Yang, J., Fong, S., & Zhao, Q. (2019). *Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges*.

IEEE Xplore Full-Text PDF: (n.d.-a). Retrieved June 2, 2023, from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9497085

IEEE Xplore Full-Text PDF: (n.d.-b). Retrieved March 4, 2023, from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9056562

Kobiela, D., Krefta, D., Król, W., & Weichbroth, P. (2022). ARIMA vs LSTM on NASDAQ stock exchange data. *Procedia Computer Science*, *207*, 3830–3839. https://doi.org/10.1016/j.procs.2022.09.445

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. In *Expert Systems with Applications* (Vol. 197). Elsevier Ltd. https://doi.org/10.1016/j.eswa.2022.116659

Liu, J., & Wang, R. (2021). *Research performance prediction for scientists using LSTM neural networks*.

Mehta, P., Pandya, S., & Kotecha, K. (2021a). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. *PeerJ Computer Science*, *7*, 1–21. https://doi.org/10.7717/PEERJ-CS.476/SUPP-2

Mehta, P., Pandya, S., & Kotecha, K. (2021b). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. *PeerJ Computer Science*, *7*, 1–21. https://doi.org/10.7717/peerj-cs.476

Mishra, P., Biancolillo, A., Roger, J. M., Marini, F., & Rutledge, D. N. (2020). New data preprocessing trends based on an ensemble of multiple preprocessing techniques. In *TrAC - Trends in Analytical Chemistry* (Vol. 132). Elsevier B.V. https://doi.org/10.1016/j.trac.2020.116045

Moghar, A., & Hamiche, M. (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. *Procedia Computer Science*, *170*, 1168–1173. https://doi.org/10.1016/J.PROCS.2020.03.049

Musciotto, F., Piilo, J., & Mantegna, R. N. (n.d.). *High-frequency trading and networked markets*. https://doi.org/10.1073/pnas.2015573118/-/DCSupplemental.y

NASDAQ Composite (^IXIC) Historical Data - Yahoo Finance. (n.d.). Retrieved March 4, 2023, from https://finance.yahoo.com/quote/%5EIXIC/history?p=%5EIXIC&guccounter=1

Nilsson, N. J. (1996). *INTRODUCTION TO MACHINE LEARNING AN EARLY DRAFT OF A PROPOSED TEXTBOOK*.

Pan, L., & Mishra, V. (2016). *Stock Market Development and Economic Growth: Empirical Evidence from China*.

Poornima, S., & Pushpalatha, M. (2019). Prediction of rainfall using intensified LSTM-based recurrent Neural Network with Weighted Linear Units. *Atmosphere*, *10*(11). https://doi.org/10.3390/atmos10110668

Rather, A. M. (2021). LSTM-based Deep Learning Model for Stock Prediction and Predictive Optimization Model. *EURO Journal on Decision Processes*, *9*. https://doi.org/10.1016/j.ejdp.2021.100001

Ribeiro, A. H., Tiels, K., Aguirre, L. A., & Schön, T. B. (2020). *Beyond exploding and vanishing gradients: analyzing RNN training using attractors and smoothness*.

Roy, S. S., Chopra, R., Lee, K. C., Spampinato, C., & Mohammadi-Ivatlood, B. (2020). Random forest, gradient boosted machines and deep neural network for stock price forecasting: A comparative analysis on South Korean companies. *International Journal of Ad Hoc and Ubiquitous Computing*, *33*(1), 62–71. https://doi.org/10.1504/IJAHUC.2020.104715

Sahu, S. K., Mokhade, A., & Bokde, N. D. (2023). An Overview of Machine Learning, Deep Learning, and Reinforcement Learning-Based Techniques in Quantitative Finance:

Recent Progress and Challenges. *Applied Sciences 2023, Vol. 13, Page 1956*, *13*(3), 1956. https://doi.org/10.3390/APP13031956

Seda Karateke, Metin Zontul, Nagihan Esendal Bozkurt, & Aslan, Z. (2021). Wavelet-ANFIS hybrid model for MODIS NDVI prediction. Journal of Applied Remote Sensing, 15(02).https://doi.org/10.1117/1.jrs.15.02451

Shen, J., & Shafiq, M. O. (2020a). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00333-6

Shen, J., & Shafiq, M. O. (2020b). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00333-6

Singh, T., Kalra, R., Mishra, S., Satakshi, & Kumar, M. (2022). An efficient real-time stock prediction exploiting incremental learning and deep learning. *Evolving Systems*. https://doi.org/10.1007/s12530-022-09481-x

Singh, V., Marcel, R. ·, Durgesh, W., Mishra, K., Amit, ·, Shikha, J., & Editors, M. (n.d.). *Advances in Intelligent Systems and Computing 841 Emerging Trends in Expert Applications and Security Proceedings of ICETEAS 2018*. http://www.springer.com/series/11156

Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., Bileschi, S., Terry, M., Nicholson, C., Gupta, S. N., Sirajuddin, S., Sculley, D., Monga, R., Corrado, G., Viégas, F. B., & Wattenberg, M. (2019). *TENSORFLOW.JS: MACHINE LEARNING FOR THE WEB AND BEYOND*.

Soni, P., Tewari, Y., & Krishnan, D. (2022). Machine Learning Approaches in Stock Price Prediction: A Systematic Review. *Journal of Physics: Conference Series*, *2161*(1). https://doi.org/10.1088/1742-6596/2161/1/012065

Staudemeyer, R. C., & Morris, E. R. (2019). *Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks*. http://arxiv.org/abs/1909.09586

Tang, Q., Shi, R., Fan, T., Ma, Y., & Huang, J. (2021). Prediction of Financial Time Series Based on LSTM Using Wavelet Transform and Singular Spectrum Analysis. *Mathematical Problems in Engineering*, *2021*. https://doi.org/10.1155/2021/9942410

Tüzen, F., Erbaş, S., & Olmuş, H. (2018). A simulation study for count data models under varying degrees of outliers and zeros. Communications in Statistics - Simulation and Computation, 49(4), 1078–1088. https://doi.org/10.1080/03610918.2018.1498886

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. *Procedia Computer Science*, *167*, 599–606. https://doi.org/10.1016/j.procs.2020.03.326

Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q., & Wang, J. (2021). A Stock Closing Price Prediction Model Based on CNN-BiSLSTM. *Complexity*, *2021*. https://doi.org/10.1155/2021/5360828

Wiederhold, G., & McCarthy, J. (2010a). Arthur Samuel: Pioneer in Machine Learning. *IBM Journal of Research and Development*, *36*(3), 329–331. https://doi.org/10.1147/rd.363.0329

Wiederhold, G., & McCarthy, J. (2010b). Arthur Samuel: Pioneer in Machine Learning. *IBM Journal of Research and Development*, *36*(3), 329–331. https://doi.org/10.1147/rd.363.0329

Williamson, B. D., Gilbert, P. B., Carone, M., & Simon, N. (2021). Nonparametric variable importance assessment using machine learning techniques. *Biometrics*, *77*(1), 9–22. https://doi.org/10.1111/biom.13392

Zhao, C., Liu, X., Zhou, J., Cen, Y., & Yao, X. (2022). GCN-based stock relations analysis for stock market prediction. *PeerJ Computer Science*, *8*. https://doi.org/10.7717/PEERJ-CS.1057

APPENDIX A. Python codes for building the LSTM network

Below is the source code employed in the completion of the task

```
#importation of libraries

import numpy as np

import pandas as pd
import matplotlib.pyplot as plt
import math
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM

# scale the data
scaler = MinMaxScaler(feature_range=(0,1))
#transforming our data based on two input values (0 and 1) above this code
scaled_data = scaler.fit_transform(dataset)

scaled_data

#create the training dataset
#create the scaled training data set
train_data = scaled_data[0:training_data_len, :]
#split the data into x_train and y_train datasets
x_train = [] #independent training variable or training features
y_train = [] #dependent variable or target variable

for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i,0])
    if i<=61:
        print(x_train)
        print(y_train)
        print()
# x_train holds the past 60 values
# y_train dataset contains the 61st value that we want our model to predict

#convert the x_train and y_trainto numpy arrays to use them in training LSTM models
x_train, y_train =np.array(x_train), np.array(y_train)
```

```
#Reshape x_train the data because LSTM network expects the input to be 3D in the form of no. of samples, no. of time steps and number of features
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
x_train.shape

# Build the LSTM model
model = Sequential()
#Add LSTM layers to our model
model.add(LSTM(30, return_sequences=False, input_shape=(x_train.shape[1], 1)))
# model.add(LSTM(50, return_sequences=False))
# model.add(Dense(25))
model.add(Dense(1))

#Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
#An optimizer is used to improve upon the loss function while the loss function is used to measure how well the model did on the training

#Train the model
model.fit(x_train, y_train, batch_size=128, epochs=50) #fit is another name for train
#batch size is the total no. of training present in a single batch
#epoch is the no. of iterations when an entire dataset is passed forward and backward through a neural network

#Create the testing data set
#Create a new array containing scaled data from 2014 to 2157(end of the data set)
test_data = scaled_data[training_data_len - 60:, :]
#Create the datasets x_test and y_test
x_test = [] #contains the past 60 values
y_test = dataset[training_data_len:, :]#All the values we want our model to predict which are our actual test values
```

```
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])
#Convert the data to a numpy array so we can use it in the LSTM model
x_test = np.array(x_test)
#Reshape the data
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
#Get the model's predicted price values
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)

#plot the data
train = close_data_df[:training_data_len]
valid = close_data_df[training_data_len:]
valid['predictions'] = predictions
#visualise the data
plt.figure(figsize=(10,5))
plt.title('Model', fontsize=18)
plt.xlabel('Date')
plt.ylabel('Close Price', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'predictions']])
plt.legend(['Train', 'Actual close price', 'Predictions'], loc='lower right', fontsize=12)

# plot the scatter plot to show the relationship between the actual and predicted price
sns.regplot(x=valid['Close'], y=valid['predictions'],line_kws={"color": "red"}, marker='*')
from sklearn.metrics import mean_squared_error
def calculate_metrics(y_true, y_pred):
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    mape = np.mean(np.abs((y_true - y_pred) / y_true)) * 100
    r = np.corrcoef(y_true, y_pred)[0,1]
    return rmse, mape, r
```

Thesis

by Charles Adikankwu

Submission date: 26-Jul-2023 07:32PM (UTC+0300)
Submission ID: 2137158406
File name: Charles_corrected_complete_thesis.docx (724.06K)
Word count: 12696
Character count: 74307

Thesis

ORIGINALITY REPORT

6%	%	6%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"International Conference on Advanced Intelligent Systems for Sustainable Development", Springer Science and Business Media LLC, 2023 Publication	1%
2	Adamu Lawan, Muhammad Lawan, Ahmad Dikko Umar, Abba Mukhtar Bala. "Long Short Term Memory and Non-linear Autoregressive Models for Prayer Time Prediction", 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2021 Publication	<1%
3	Hum Nath Bhandari, Binod Rimal, Nawa Raj Pokhrel, Ramchandra Rimal, Keshab R. Dahal, Rajendra K.C. Khatri. "Predicting stock market index using LSTM", Machine Learning with Applications, 2022 Publication	<1%
4	Alexander Grushin, Derek D. Monner, James A. Reggia, Ajay Mishra. "Robust human action	<1%

recognition via long short-term memory", The 2013 International Joint Conference on Neural Networks (IJCNN), 2013
Publication

5	Rafael Magdalena-Benedicto, Sonia Pérez-Díaz, Adrià Costa-Roig. "Challenges and Opportunities in Machine Learning for Geometry", Mathematics, 2023 Publication	<1 %
6	Nawa Raj Pokhrel, Keshab Raj Dahal, Ramchandra Rimal, Hum Nath Bhandari, Rajendra K.C. Khatri, Binod Rimal, William Edward Hahn. "Predicting NEPSE index price using deep learning models", Machine Learning with Applications, 2022 Publication	<1 %
7	"Proceedings of the 8th International Conference on Water Resource and Environment", Springer Science and Business Media LLC, 2023 Publication	<1 %
8	Asit Kumar Das, Debahuti Mishra, Kaberi Das, Arup Kumar Mohanty et al. "A Deep Network-Based Trade and Trend Analysis System to Observe Entry and Exit Points in the Forex Market", Mathematics, 2022 Publication	<1 %

9	Rola Houhou, Thomas Bocklitz. "Trends in artificial intelligence, machine learning, and chemometrics applied to chemical data", Analytical Science Advances, 2021 Publication	<1 %
10	"Emerging Trends in Expert Applications and Security", Springer Science and Business Media LLC, 2019 Publication	<1 %
11	"Intelligent Data Engineering and Automated Learning – IDEAL 2018", Springer Science and Business Media LLC, 2018 Publication	<1 %
12	Cheng Zhao, Ping Hu, Xiaohui Liu, Xuefeng Lan, Haiming Zhang. "Stock Market Analysis Using Time Series Relational Models for Stock Price Prediction", Mathematics, 2023 Publication	<1 %
13	Chyan-Long Jan. "Detection of Financial Statement Fraud Using Deep Learning for Sustainable Development of Capital Markets under Information Asymmetry", Sustainability, 2021 Publication	<1 %
14	Sepp Hochreiter, Jürgen Schmidhuber. "Long Short-Term Memory", Neural Computation, 1997 Publication	<1 %

15	"Communication Networks and Service Management in the Era of Artificial Intelligence and Machine Learning", Wiley, 2021 Publication	<1%
16	Sayan Mukhopadhyay, Pratip Samanta. "Chapter 5 Deep Learning and Neural Networks", Springer Science and Business Media LLC, 2023 Publication	<1%
17	Archit Kapoor, Varun Rastogi, Nirbhay Kashyap. "Forecasting Daily Close Prices of Stock Indices using LSTM", 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020 Publication	<1%
18	Md. Ashraful Islam, Md. Rana Sikder, Sayed Mohammed Ishtiaq, Abdus Sattar. "Stock market prediction of Bangladesh using multivariate long short-term memory with sentiment identification", International Journal of Electrical and Computer Engineering (IJECE), 2023 Publication	<1%
19	Jaimin Shah, Darsh Vaidya, Manan Shah. "A Comprehensive Review on Multiple Hybrid Deep Learning Approaches for Stock	<1%

Prediction", Intelligent Systems with Applications, 2022
Publication

20	Shigao Huang, Jie Yang, Simon Fong, Qi Zhao. "Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges", Cancer Letters, 2020 Publication	<1 %
21	Maan Singh Rathore, S.P. Harsha. "An attention-based stacked BiLSTM framework for predicting remaining useful life of rolling bearings", Applied Soft Computing, 2022 Publication	<1 %
22	Omnia Kelany, Sherin Aly, Mohamed A. Ismail. "Deep Learning Model for Financial Time Series Prediction", 2020 14th International Conference on Innovations in Information Technology (IIT), 2020 Publication	<1 %
23	P. Chinthan, Rashmi Mishra, B. Prakash, B. Saleena. "Chapter 7 Stock Market Prediction Using Machine Learning Techniques: A Comparative Study", Springer Science and Business Media LLC, 2023 Publication	<1 %
24	"Data Analytics and Management", Springer Science and Business Media LLC, 2021 Publication	<1 %

25	"Human Interaction, Emerging Technologies and Future Applications IV", Springer Science and Business Media LLC, 2021 Publication	<1 %
26	Anood Shakeel, Bijendra Bishow Maskey, Sagar Shrestha, Sajjan Parajuli, Younsu Jung, Gyoujin Cho. "Towards Digital Twin Implementation in Roll-To-Roll Gravure Printed Electronics: Overlay Printing Registration Error Prediction Based on Printing Process Parameters", Nanomaterials, 2023 Publication	<1 %
27	KS Rekha, MK Sabu. "A cooperative deep learning model for stock market prediction using deep autoencoder and sentiment analysis", PeerJ Computer Science, 2022 Publication	<1 %
28	Makumbonori Bristone, Rajesh Prasad, Adamu Ali Abubakar. "CPPCNDL: Crude oil price prediction using complex network and deep learning algorithms", Petroleum, 2019 Publication	<1 %
29	Muhammad Junaid, Sajid Ali, Fatma Eid, Shaker El-Sappagh, Tamer Abuhmed. "Explainable machine learning models based on multimodal time-series data for the early	<1 %

detection of Parkinson’s disease", Computer Methods and Programs in Biomedicine, 2023
Publication

30 Poornachandra Sarang. "Artificial Neural Networks with TensorFlow 2", Springer Science and Business Media LLC, 2021 <1%
Publication

31 Ritika Chopra, Gagan Deep Sharma. "Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda", Journal of Risk and Financial Management, 2021 <1%
Publication

32 Shilpa Gite, Hrituja Khatavkar, Ketan Kotecha, Shilpi Srivastava, Priyam Maheshwari, Neerav Pandey. "Explainable stock prices prediction from financial news articles using sentiment analysis", PeerJ Computer Science, 2021 <1%
Publication

33 T. T. Dhivyaprabha, P. Subashini, M. Krishnaveni, N. Santhi, Ramesh Sivanpillai, G. Jayashree. "A novel synergistic fibroblast optimization based Kalman estimation model for forecasting time-series data", Evolving Systems, 2018 <1%
Publication

34 Xiaowei Huang, Gaojie Jin, Wenjie Ruan. "Chapter 10 Deep Learning", Springer Science <1%

and Business Media LLC, 2023
Publication

35	"Book reports", Computers & Mathematics with Applications, 2002 Publication	<1%
36	"Integrated Uncertainty in Knowledge Modelling and Decision Making", Springer Science and Business Media LLC, 2020 Publication	<1%
37	"Intelligent Computing Theories and Application", Springer Science and Business Media LLC, 2021 Publication	<1%
38	"Proceedings of Third International Conference on Intelligent Computing, Information and Control Systems", Springer Science and Business Media LLC, 2022 Publication	<1%
39	Akhter Mohiuddin Rather. "LSTM-based Deep Learning Model for Stock Prediction and Predictive Optimization Model", EURO Journal on Decision Processes, 2021 Publication	<1%
40	Antonella Guzzo, Mikel Joaristi, Antonino Rullo, Edoardo Serra. "A multi-perspective approach for the analysis of complex business processes behavior", Expert Systems with Applications, 2021	<1%

Publication

41	Chinedu I. Ossai, Nilmini Wickramasinghe. "Intelligent decision support with machine learning for efficient management of mechanical ventilation in the intensive care unit – A critical overview", International Journal of Medical Informatics, 2021 Publication	<1 %
42	Chuanjun Zhao, Meiling Wu, Jingfeng Liu, Zening Duan, Jie li, Lihua Shen, Xuekui Shangguan, Donghang Liu, Yanjie Wang. "Progress and Prospects of Data-Driven Stock Price Forecasting Research", International Journal of Cognitive Computing in Engineering, 2023 Publication	<1 %
43	Divyansh Divyansh, Abdullah Moonis, Animesh Raghav, Aditya Walture, Siddharth Singh Chouhan, Uday Pratap Singh, Akash Saxena. "Stock Closing Price Prediction of Indian Tech Companies using Deep Neural Networks", Research Square Platform LLC, 2023 Publication	<1 %
44	Ekaterina Vitalevna Chumakova, Dmitry Gennadievich Korneev, Tatiana Alexandrovna Chernova, Mikhail Samuilovich Gasparian et al. "Comparison of the Application of FNN and	<1 %

LSTM Based on the Use of Modules of Artificial Neural Networks in Generating an Individual Knowledge Testing Trajectory", Journal Européen des Systèmes Automatisés, 2023

Publication

45	Guide to OCR for Arabic Scripts, 2012. Publication	<1 %
46	Ibtissam Medarhri, Mohamed Hosni, Najib Nouisser, Fatma Chakroun, Khalid Najib. "Predicting Stock Market Price Movement using Machine Learning Techniques", 2022 8th International Conference on Optimization and Applications (ICOA), 2022 Publication	<1 %
47	Imad Sassi, Samir Anter, Abdelkrim Bekkhoucha. "A graph-based big data optimization approach using hidden Markov model and constraint satisfaction problem", Journal of Big Data, 2021 Publication	<1 %
48	Jingjun Guo, Weiyi Kang, Yubing Wang. "Option pricing under sub-mixed fractional Brownian motion based on time-varying implied volatility using intelligent algorithms", Soft Computing, 2023 Publication	<1 %

49 Muhammad Raheel Raza, Walayat Hussain, Jose Maria Merigo. "Cloud Sentiment Accuracy Comparison using RNN, LSTM and GRU", 2021 Innovations in Intelligent Systems and Applications Conference (ASYU), 2021 <1%
Publication

50 Naga Pranava Shashank V, Vimal S Ramanan, Altrin Lloyd Hudson D, A Sheik Abdullah. "An Efficient Decision Support System for Stock Market Prediction using Technical Indicators as Features by Machine Learning Approaches", 2023 4th International Conference for Emerging Technology (INCET), 2023 <1%
Publication

51 Noella Nazareth, Yeruva Yenkata Ramana Reddy. "Financial applications of machine learning: a literature review", Expert Systems with Applications, 2023 <1%
Publication

52 Santanu Pattanayak. "Chapter 4 Natural Language Processing", Springer Science and Business Media LLC, 2023 <1%
Publication

53 Steve D. Yang, Zulfikhar A. Ali, Hyuna Kwon, Bryan M. Wong. "Predicting Complex Erosion Profiles in Steam Distribution Headers with Convolutional and Recurrent Neural <1%

Networks", Industrial & Engineering Chemistry Research, 2022
Publication

54	Yang Yue, Jin-Hai Li, Li-Feng Fan, Li-Li Zhang, Peng-Fei Zhao, Qiao Zhou, Nan Wang, Zhong-Yi Wang, Lan Huang, Xue-Hui Dong. "Prediction of maize growth stages based on deep learning", Computers and Electronics in Agriculture, 2020 Publication	<1 %
55	"Computer Networks and Inventive Communication Technologies", Springer Science and Business Media LLC, 2023 Publication	<1 %
56	"Innovative Engineering with AI Applications", Wiley, 2023 Publication	<1 %
57	B V Vishwas, Ashish Patel. "Chapter 7 Bleeding-Edge Techniques for Multivariate Time Series", Springer Science and Business Media LLC, 2020 Publication	<1 %
58	Daniel A. Spielman. "Smoothed analysis of algorithms", Journal of the ACM, 5/1/2004 Publication	<1 %
59	Santosh Kumar Sahu, Anil Mokhade, Neeraj Dhanraj Bokde. "An Overview of Machine Learning, Deep Learning, and Reinforcement	<1 %

Learning-Based Techniques in Quantitative Finance: Recent Progress and Challenges", Applied Sciences, 2023
Publication

60	V Kishore Ayyadevara. "Chapter 10 Recurrent Neural Network", Springer Science and Business Media LLC, 2018 Publication	<1 %
61	Zeynep Cipiloglu Yildiz, Selim Baha Yildiz. " A portfolio construction framework using - based stock markets forecasting ", International Journal of Finance & Economics, 2020 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		