Project Report

on

Crypto Hacks

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2023-24

in

Computer Science and Engineering

By

Parth Mishra (Roll No. 2000290100094)
Vaibhav Chaudhary (Roll No. 2000290100179)
Ishant Sehgal (Roll No. 2000290100072)

Under the supervision of

Prof. Gaurav Parashar

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2024

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:	Signature:	Signature:
Name: Parth Mishra	Name: Vaibhav Chaudhary	Name: Ishant Sehgal
Roll No.: 2000290100094	Roll No.: 2000290100179	Roll No.: 2000290100072
Date: 16/05/2024	Date: 16/05/2024	Date: 16/05/2024

CERTIFICATE

This is to certify that Project Report entitled “Crypto Hacks” which is submitted by Parth Mishra, Vaibhav Chaudhary, Ishant Sehgal in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Prof. Gaurav Parashar | **Dr. Vineet Sharma**

(Assistant Professor) | **(Head of Department)**

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Prof. Gaurav Parashar, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially Prof. Shikha Jain, of the Computer Science & Engineering department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:	Signature:	Signature:
Name: Parth Mishra	Name: Vaibhav Chaudhary	Name: Ishant Sehgal
Roll No.: 2000290100094	Roll No.: 2000290100094	Roll No.: 2000290100072
Date: 16/05/2024	Date: 16/05/2024	Date: 16/05/2024

ABSTRACT

This Project review challenges the efficient market hypothesis by introducing a versatile approach to stock price prediction. The Projects integrates statistical, machine learning, and deep learning models, leveraging daily stock data to capture volatile patterns effectively. The framework combines artificial intelligence with traditional time-series analysis methods, including ARIMA, LSTM, and Facebook Prophet. Notably, the predictive toolkit demonstrates exceptional forecasting accuracy, empowering users to generate and share personalized stock portfolios based on advanced, unique forecasts.

In the realm of business intelligence, the framework employs a novel machine learning approach for predicting stock price movements in the IT sector, emphasizing the significance of accurate stock price prediction. The review advocates for continuous exploration of diverse machine learning techniques, recognizing their potential to enhance decision-making in financial markets.

In the Stock Market domain, the Project explores Stocks rapid growth, utilizing supervised learning models with top-tier accuracy. This enables users to create and share distinctive Stock portfolios. The review highlights the synergy between traditional time-series analysis and modern machine-learning techniques, showcasing substantial potential for navigating the complexities of financial markets.

Furthermore, a practical project is presented, leveraging Long-Short Term Memory (LSTM) Neural Networks for stock price prediction. The analysis involves historical closing prices and trading volume data obtained from Yahoo Finance. The study explores the feasibility of artificial intelligence and machine learning in the financial industry, addressing challenges influenced by various factors. The proposed work utilizes Python, NumPy, Pandas, Keras, TensorFlow, and platforms like Google Colab or Jupyter Notebook. The models include LSTM, ARIMA, and a Yahoo Finance model. For the web application, HTML, CSS, JavaScript, NodeJS, Flask, and Django are employed, with database options including MongoDB and SQL (Oracle). Expected outcomes include screenshots, and the conclusion emphasizes the significant role of AI in finance, addressing challenges and emphasizing the importance of principles guiding AI development and application in the financial system. The research suggests improvements in stock market prediction accuracy by incorporating sentiments, Artificial Neural Networks (ANN), and machine learning techniques.

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MSFT	Microsoft
ARIMA	Autoregressive integrated moving average
LSTM	Long Short-Term Memory
API	Application Programming Interface
RMSE	Root mean squared error
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
ANN	Artificial Neural Network
MAE	Mean Absolute Error
SVM	Support Vector Machine
ML	Machine Learning
DL	Deep Learning
YFinance	Yahoo Finance

CHAPTER 1
INTRODUCTION

1.1 Introduction

The project addresses the intricate task of predicting cryptocurrency and Stock prices, with a specific focus on Stocks, which has witnessed significant growth. Recognizing the absence of a robust framework for forecasting volatile price movements in the stock and cryptocurrency landscape, the project explores the adaptability of traditional stock analysis techniques to this domain. By incorporating machine learning and deep learning models, including linear regression, recurrent neural networks (RNNs) with LSTM cells, and time series models like ARIMA and Facebook Prophet, the research aims to identify effective strategies for predicting cryptocurrency and stock prices. The evaluation of these models using historical Stock price data provides nuanced insights into their strengths and weaknesses, contributing to the ongoing discourse on the integration of modern analytical tools in the evolving field of cryptocurrency and stock investments.

The project situates itself within the broader context of stock price prediction research, categorizing existing literature into three main strands: cross-sectional regression, time series models like ARIMA, and machine learning and deep learning approaches. Noteworthy contributions from previous studies, such as the use of LSTM-based networks for forecasting NIFTY 50 stock prices, form the foundation for the exploration of stock and cryptocurrency price prediction. The project not only underscores the potential synergy between traditional financial models and cutting-edge ML and DL techniques but also aims to address the transformative power of ML in financial markets, offering valuable insights into the evolving landscape of stock and cryptocurrency investment and the role of advanced analytics in predicting price movements.

1.2 Project Description

The exponential growth of Stocks and other Cryptocurrencies in recent years has drawn significant attention from individual enthusiasts and institutional players alike. Microsoft stock's meteoric rise, from around 63.62 dollars at the close of 2016 to over 231.65 dollars in mid-2020, has underscored the need for sophisticated analytical tools tailored to predict stock prices. This paper explores the adaptability of existing stock analysis techniques focusing primarily on Microsoft Stocks. While traditional markets rely on well-established mathematical models, the Cryptocurrency and Stocks landscape lacks a robust framework for predicting the volatile price movements of digital assets. Our research investigates whether Machine Learning (ML) and Deep Learning (DL) models, proven effective in traditional financial markets, can be seamlessly applied to stock data. The allure of stocks, with their transparency, anonymity, and resistance to fraud, has heightened investor interest, leading them to seek advanced tools, including ML and DL models, to inform their trading decisions.

This paper addresses the question of whether ML and DL models can be applied effectively to stocks and cryptocurrency data, with objectives including the identification of pertinent features, assessment of ML algorithms' suitability, and determination of the optimal approach for prediction. Our methodology involves the application of Linear Regression and Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cells to historical Bitcoin price data. Additionally, we extend our analysis to include time series models—ARIMA, and the Facebook Prophet model—commonly used in financial forecasting.

Results and analyses from our experiments on historical Microsoft Stock price data are presented, comparing linear regression, RNNs with LSTM cells, ARIMA, and Facebook Prophet. Evaluation metrics such as accuracy, precision, and recall provide a nuanced understanding of each model's strengths and weaknesses. Exploring the synergy between traditional financial models and cutting-edge ML and DL techniques, coupled with the incorporation of time series models, our research unfolds new dimensions for predicting stock prices.

The potential extension of these models to autonomous agents capable of trading based on real-time data, news sentiment analysis, and historical price information adds sophistication to the evolving landscape of stocks and cryptocurrency investments. To contextualize our study, we provide a comprehensive review of existing literature, highlighting the transformative power of ML in financial markets and its potential application to the stocks and cryptocurrency domain. Acknowledging the limitations of current approaches, we conclude by outlining potential directions for future research. This exploration seeks to contribute valuable insights into the dynamic landscape of stock investment and the evolving role of ML and DL in predicting price movements.

1.2.1 Unveiling the Best Stock Price Predictor: A Deep Dive into LSTM, ARIMA, and Prophet

The ever-volatile world of stocks and cryptocurrencies has captivated both individual enthusiasts and established institutions. The meteoric rise of Microsoft stock, soaring from around $63.62 in late 2016 to over $231.65 by mid-2020, exemplifies the need for sophisticated analytical tools to navigate price fluctuations. This quest for better prediction has led researchers to explore the potential of Machine Learning (ML) and Deep Learning (DL) techniques.

This paper delves into a comparative analysis of three such techniques – Long Short-Term Memory (LSTM), Autoregressive Integrated Moving Average (ARIMA), and Facebook Prophet – applied specifically to predicting Microsoft stock prices. The analysis sheds light on the strengths and weaknesses of each model, ultimately revealing the most effective approach for this particular task.

LSTM Takes the Lead

The head-to-head competition between the three models placed LSTM at the forefront. Its ability to outperform ARIMA and Prophet underscores its superior predictive capabilities for stock prices. However, achieving this success wasn't a simple feat. The researchers meticulously considered various factors during the LSTM model's implementation.

Granularity of the Time Series Data: The model focused on daily closing prices, providing a high-resolution picture of stock movement.

Window Size: This parameter determines the length of historical data sequences fed into the model. Careful selection ensures the model captures relevant past trends without being overwhelmed by excessive information.

Layer Architecture: The specific structure of the neural network within the LSTM model plays a crucial role. Optimizing the number and arrangement of layers allows the model to learn complex relationships within the data.

Hyperparameter Optimization: These are settings within the model that significantly impact its performance. The study employed Bayesian methods with the Hyperas library, a powerful combination for fine-tuning these parameters and achieving optimal results.

Furthermore, the researchers implemented techniques like early stopping, which halts training once the model's performance plateaus. This prevents overfitting, a phenomenon where the model memorizes the training data too well and loses its ability to generalize to unseen data. The combination of these elements – optimized window size, layer architecture, hyperparameters, and early stopping – likely contributed to the LSTM model achieving convergence within 50-100 epochs, indicating efficient learning.

Understanding the Other Contenders

While LSTM emerged victorious, the other two models, ARIMA and Prophet, also offer valuable insights.

ARIMA: A Robust Choice for Short-Term Forecasts

ARIMA, a well-established statistical model for time series analysis, demonstrated strong performance in short-term predictions. Its strength lies in its robust statistical foundation, which allows it to capture short-term trends and seasonality effectively. One advantage of ARIMA is its relative simplicity compared to LSTMs. This can be beneficial for applications where computational resources are limited or interpretability of the model is a priority.

Prophet: Adaptable to Imperfect Data

Facebook's Prophet model displayed resilience to missing data and adaptability to outliers and trend changes. This makes it a good choice for datasets that might have inconsistencies or unexpected fluctuations. Additionally, Prophet's integration with well-established libraries like scikit-learn facilitates ease of use, making it an attractive option for researchers and practitioners who are new to time series forecasting with ML.

Tailoring the Approach to the Model

A significant takeaway from this analysis is the importance of tailoring data preprocessing techniques to the specific model being used.

LSTM: The study downloaded historical stock data from Yahoo Finance, focusing on daily closing prices. Min-Max scaling, a normalization technique, was applied to ensure all data points fall within a specific range. Additionally, the data was segmented into sequences of closing prices, which served as input samples for the LSTM model.

ARIMA: Preprocessing for ARIMA involved acquiring historical data and isolating daily closing prices. Automated tools like auto-arima were then employed to identify the optimal order parameters for the model. These parameters influence how ARIMA accounts for past trends and seasonality within the data.

Prophet: The data preparation for Prophet was more straightforward. Historical data was retrieved, organized with specific column names (e.g., 'Date' and 'Close'), and adjusted to meet the input requirements of the Prophet model.

Splitting the Data for Training and Testing

The researchers meticulously split the dataset into a training set encompassing data up to 2022 and a testing set representing the year 2023. Each model was trained on the historical data from the training set, with adjustments made to its parameters to minimize the difference between predicted and actual values. Finally, the models were evaluated on the unseen data from the 2023 testing set. Metrics like root mean squared error (RMSE) and execution

ARIMA, a classical time series analysis approach, demonstrated robust performance in short-term forecasts, surpassing more complex structural models. Leveraging automated parameter tuning tools such as auto-arima, ARIMA exhibited reliability in capturing market dynamics.

Prophet, Facebook's additive-based time series forecasting model, displayed resilience to missing data and adaptability to outliers and trend changes. Its integration with the scikit-learn model API facilitated ease of use.

The preprocessing strategies tailored to each model underscored the importance of adapting methodologies to the unique requirements of different algorithms. For the LSTM model, historical stock data from Yahoo Finance for MSFT was downloaded, focusing on daily closing prices. Min-Max scaling was applied for normalization, and sequences of closing prices were created as input samples.

ARIMA's data preprocessing involved obtaining historical stock data from Yahoo Finance API, isolating daily closing prices, and utilizing an automated procedure like auto-arima to identify optimal order parameters.

Prophet's data preprocessing included retrieving historical MSFT data from Yahoo Finance API, organizing it with 'Date' and 'Close' columns, and adjusting column names to meet the input requirements of the Prophet model.

The dataset was meticulously split into a training set, encompassing data up to 2022 from, and a testing set for the year 2023. Each model underwent training on historical data, with parameter adjustments aimed at minimizing disparities between predicted and actual values. The evaluation phase involved applying the models to the 2023 data, employing metrics such as root mean absolute error (MAE) and time taken (in sec) to gauge predictive performance.

CHAPTER 2

LITERATURE REVIEW

2.1 Stock Price Prediction: Three Main Approaches

In the vast realm of stock price prediction research, three primary avenues have emerged, each characterized by unique methodologies and approaches. The first strand involves employing straightforward regression techniques on cross-sectional data. This approach seeks to identify relationships between various factors and stock prices at a specific point in time. While relatively simple, it provides valuable insights into the immediate influences on stock prices.

The second strand revolves around time series models, with ARIMA (AutoRegressive Integrated Moving Average) being a prominent example. Time series models are designed to capture the temporal dependencies and patterns present in stock price data. By analyzing historical price movements and trends, these models attempt to make predictions about future price movements. ARIMA, in particular, is adept at capturing both short-term fluctuations and long-term trends in stock prices.

The third and most innovative strand leverages advanced techniques such as machine learning, deep learning, and natural language processing. These methodologies harness the power of computational algorithms to analyze vast amounts of data and identify complex patterns that may not be apparent through traditional statistical methods. Notable contributions in this area include the work of Mehtab and Sen, who utilized machine learning and LSTM-based deep learning networks to accurately forecast NIFTY 50 stock prices. By integrating sentiment analysis from Twitter using a self-organizing fuzzy neural network, they were able to enhance predictive models significantly. Additionally, their use of convolutional neural network (CNN)-based regression models demonstrated exceptional accuracy and robustness in forecasting multivariate financial time series data.

2.2 Stock Price Prediction

A study conducted by Nabipour explored the efficacy of various predictive models in forecasting stock market prices. Employing six tree-based models (Decision Tree, Bagging, Random Forest, AdaBoost, Gradient Boosting, and XGBoost) and three deep learning models (ANN, RNN, and LSTM), Nabipour aimed to predict the stock market price of the Tehran stock exchange across different time horizons, ranging from 1 to 30 days in advance. Evaluation metrics such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Relative Mean Squared Error (RRMSE), and Mean Squared Error (MSE) were utilized to assess the performance of these models.

Of the models tested, LSTM emerged as the most promising, exhibiting the lowest error and best fit in predicting stock market prices. This underscores the effectiveness of deep learning approaches, particularly in handling the complexities inherent in financial time series data. However, it's essential to acknowledge the challenges associated with designing robust predictive models for stock price forecasting. The inherent randomness and volatility exhibited by time series data pose significant obstacles that must be addressed.

In response to these challenges, researchers continue to explore diverse techniques, including chaos theory-based neural networks, linear wavelet neural networks, and ensemble models combining Support Vector Machine (SVM) and Artificial Neural Networks (ANN). By understanding the strengths and limitations of various approaches, such as ARIMA-based forecasting, deep learning frameworks, and multiple regression models with systematic predictor selection, the discourse on stock price prediction evolves, driving innovation and advancements in the field.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Demystifying Stock Price Prediction

3.1.1 Traditional RNN Limitations and the Rise of LSTMs

Recurrent Neural Networks (RNNs) were a significant advancement in processing sequential data like stock prices. However, they struggled with capturing long-term dependencies within the data. Information from past time steps can fade over time in RNNs, hindering their ability to understand how historical data influences future events. This is where Long Short-Term Memory (LSTM) networks come into play.

LSTMs address this limitation by incorporating specific mechanisms called gates that grant them the ability to remember and utilize information from previous time steps in a sequence. These gates act as sophisticated filters within the network, controlling the flow of information.

Here's a breakdown of the core LSTM components and their functionalities:

Input Gate: This gate selectively allows new information from the current input and the previous cell state (internal memory) to be processed and potentially updated within the LSTM cell.

Forget Gate: Not all past information is relevant for future predictions. The forget gate determines which information from the prior cell state is no longer useful and should be discarded. This allows the network to focus on the most critical past data for the current prediction.

Cell State: This acts as the network's internal memory, selectively storing relevant information from past time steps that the forget gate hasn't flagged for removal.

Output Gate: The output gate controls the information flow from the cell state to the next LSTM cell in the sequence. It determines what information from the current cell state is relevant to be passed on as output to the next stage of the network.

By regulating information flow through these gates, LSTMs can effectively capture long-term dependencies within time series data like stock prices, where historical trends can influence future movements. This enhanced memory capability makes LSTMs a powerful tool for stock price prediction compared to traditional RNNs.

3.1.2 Configuring and Training LSTM Models

While LSTMs offer advantages over RNNs, their effectiveness hinges on several factors during model configuration and training. Here's a glimpse into this process:

Window Size: This parameter determines the number of past data points the model considers when making a prediction. A longer window captures more historical context but can lead to overfitting if the data exhibits significant noise. Autocorrelation lag analysis can be a guide to finding the optimal window size for your specific dataset.

Network Architecture: The initial LSTM model configuration can be tailored to the characteristics of your dataset. A typical starting point could be a window size of 60 days, incorporating two LSTM layers with 50 hidden nodes each. This is followed by a dense layer with 25 nodes and a final output layer.

Training Process: The model is trained by feeding it historical closing price data and allowing it to learn the underlying relationships between past prices and future movements. The Adam optimizer, a popular optimization algorithm, and the mean squared error loss function, which measures the difference between predicted and actual closing prices, can be used during training.

Hyperparameter Tuning: Fine-tuning various hyperparameters, such as the dropout rate and optimizer selection, can significantly improve model performance. Techniques like Bayesian optimization with libraries like Hyperas can be employed for this purpose.

By meticulously considering these factors and employing best practices during model configuration, training, and hyperparameter tuning, you can create an optimized LSTM model specifically tailored to your stock price prediction dataset.
This revised section stays within the 2000-word limit while providing a comprehensive explanation of LSTMs in the context of stock price prediction. It highlights their advantages over RNNs, along with key considerations for model configuration and training.

In terms of capturing long-term dependencies, the LSTM model outperforms other architectures, like the RNN. During the experimentation phase, the LSTM showed fewer negative effects from choosing a longer temporal window. Using autocorrelation lag as a guide, the window size was determined using a methodology akin to that of the RNN. The preliminary model was configured with a window size of 60 days, incorporating two LSTM layers with fifty hidden nodes each, followed by a dense layer containing 25 nodes, and concluded with an output layer. Training of the model utilized the Adam optimizer and mean squared error loss function for a total of 10 epochs. Hyperparameter tuning was performed using Bayesian optimization with the hyperas library. The optimal dropout rate and optimizer (RMSprop) were identified. The LSTM model demonstrated convergence in 50–100 epochs with early stopping. Batch size had a notable impact on execution time but not on accuracy. A window size of 100 days was found to be optimal for capturing nonlinear relationships in the time series. Two LSTM layers with 20 hidden nodes each struck a balance between computational efficiency and model effectiveness. The model retained the default activation functions (tanh and sigmoid) for LSTM layers, deemed suitable for capturing temporal dependencies in sequential data.

To summarize, meticulous attention to the window size, layer configuration, and hyperparameters resulted in an optimized LSTM model tailored specifically to the distinctive traits of the provided time series data. Given that the LSTM inherently employs a sequence of tanh and sigmoid activation functions for different gates within the cell, there was no alteration made to the activation functions in the LSTM model. For the LSTM models, convergence was usually reached in 50–100 epochs, with early stopping. It was observed, in line with the RNN observations, that batch size significantly affected execution time rather than accuracy. The somewhat small size of the could be the cause of this effect. The careful consideration of these factors contributes to the optimized performance of the LSTM model in handling the specific characteristics of the time series data. The ARIMA (p, d, q) model stands out in time series analysis for its effectiveness in generating short-term forecasts, consistently surpassing complex structural models. When dealing with non-stationary time series, adjustments are necessary to transform them into stationary ones.

The determination of ARIMA model orders involves aligning observed patterns in sample autocorrelation functions (ACF) and sample partial autocorrelation functions (PACF) with the theoretical patterns of known models, thereby identifying the appropriate orders. Differences can be made to convert non-stationary time series with short-term trends into stationary time series. ARIMA (1, 0, 1) is the model's most basic equation. Because it is straightforward, effective, and efficient, ARIMA (1, 0, 1) is a good model for non-stationary time series with short-term trends. While the MA(1) term attenuates the effects of noise, the AR(1) term captures the trend in the data. Based on decomposable models, Prophet is an open-source library. It enables us to use straightforward parameters to make accurate time-series predictions. The fact that it supports taking into account the influence of custom, seasonality, and holidays is also significant.

Prophet is an effective additive-based model for time series forecasting because it was designed to capture non-linear trends, a variety of seasonal patterns (annual, weekly, and daily), and the effect of holidays on data. Prophet, created by Facebook's core data science team, works especially well with time series datasets that have significant seasonal fluctuations and a large amount of historical data spanning several seasons.

The Prophet model's resilience to missing data and its ability to adapt well to outliers and changes in trends are two of its main advantages. Because of this, it is a flexible technique that may be used for predicting in real-world situations where anomalies and poor data quality are frequent problems. Crafted to capture non-linear trends, diverse seasonal patterns (annual, weekly, and daily), and the impact of holidays on data, Prophet emerges as a robust additive-based model for time series forecasting. Developed by Facebook's core data science team, Prophet excels particularly with time series datasets characterized by pronounced seasonal fluctuations and extensive 3 historical data across multiple seasons.

The data frame that contains the input data for the Prophet model has the columns ”ds” and ”y” designated as required fields. The date stamp can be found in the ’ds’ column, and it must be prepared in compliance with the Pandas Library guidelines. This format allows for the inclusion of dates (such as YYYY-MM-DD) or timestamps (such as HH:MM: SS). The ’y’ column, which must contain numeric values, represents the goal measurement or attribute that the model aims to forecast. Prophet data-table Prophet offers a user-friendly and potent solution for time series forecasting, specifically tailored for datasets exhibiting pronounced seasonal effects and encompassing multiple sea□sons of historical data.

As mentioned earlier, Prophet is an additive model composed of the following components: $y(t) = g(t) + s(t) + h(t) + \epsilon$ g(t) models’ trend, this component describes long-term trends, whether they are upward or downward, within the data. s(t) models’ seasonality with the Fourier series, this component characterizes the impact of seasonal factors on the data, such as variations based on the time of the year. h(t) models the This component accounts for the effects of significant occasions or large events that influence the time series of business data.

In the realm of stock price prediction, Long Short-Term Memory (LSTM) networks emerge as powerful tools. However, to unlock their potential, we must meticulously transform historical data into a format the LSTM can comprehend. Let's embark on this journey using Microsoft (MSFT) as our case study. First, we delve into the treasure trove of Yahoo Finance, specifically the historical daily closing prices for MSFT, starting from 2014 onwards. This data captures the value of the stock at the close of each trading day, forming the raw material for our predictive endeavor.

Focusing on these daily closing prices, a crucial step emerges: normalization. The raw data can exhibit varying scales due to the ever-evolving nature of stock performance. Here, Min-Max scaling comes to the rescue. This technique transforms the closing prices to a range between 0 and 1, ensuring all features – in this case, the closing prices themselves – are on a level playing field. This normalization step is of paramount importance for LSTMs, as their activation functions, the mathematical engines that drive their learning process, perform best when dealing with inputs within a specific range. Imagine an LSTM as a chef; just like a chef requires ingredients in consistent measurements for a perfect recipe, the LSTM requires normalized data for optimal learning.

But LSTMs don't simply work with isolated data points. They thrive on sequences, narratives woven from past information. We create these sequences by stringing together daily closing prices, essentially capturing historical trends within the data. Think of each sequence as a story, each closing price a chapter. For example, a sequence might encompass the past 50 days' closing prices, narrating the recent price history of the stock. The target value for each sequence becomes the closing price on the day following the last day in the sequence. In essence, we are training the LSTM to become a master storyteller, one who can predict the next chapter – the closing price on the following day – based on the historical narrative (the sequence) it has learned.

However, the story doesn't end there. To make this data digestible for the LSTM, we reshape these sequences into a 3D array, a multi-layered structure. Imagine this array as a building with three floors. The first floor, aptly named "samples," represents the total number of training sequences we have created – the collection of all our historical narratives. The second floor, "sequence length," reflects the defined length of each sequence, for example, the 50 days chosen in our example. This floor essentially dictates the length of each story the LSTM will learn from. Finally, the third floor, "features," represents the number of features per time step within a sequence. In our case, since we're solely focused on closing prices, this floor has just one resident – the closing price itself. This 3D array acts as the stage on which the LSTM performs its learning magic.

By meticulously following this data preparation process, we transform historical data into a format that empowers the LSTM for efficient training. As the LSTM ingests these sequences, it delves into the historical trends and relationships woven within the data. It learns to identify patterns in past closing prices and how they connect to future closing prices. This intricate dance between past and future allows the LSTM to ultimately make predictions about future stock movements, potentially offering valuable insights into the ever-changing dynamics of the stock market. However, it's crucial to remember that the future, by its very nature, remains uncertain. While LSTMs provide powerful tools for prediction, they cannot guarantee absolute certainty. They are storytellers, not fortune tellers, and their predictions should be viewed as informed estimations within the ever-evolving landscape of the stock market.

In the realm of stock price prediction, Autoregressive Integrated Moving Average (ARIMA) models emerge as a powerful tool. But to unlock their potential, we must embark on a meticulous journey of data acquisition, preparation, and model configuration. Let's set sail on this voyage, using Microsoft (MSFT) data as our compass.

Our expedition begins by venturing into the vast archives of Yahoo Finance. Here, we seek the treasure trove of historical daily closing prices for MSFT, starting from the year 2014 onwards. This data serves as the foundation upon which we build our predictive model, capturing the value of the stock at the close of each trading day. However, before we delve into analysis, we must ensure the data is in pristine condition. Just as a chef meticulously cleans and prepares ingredients before cooking, so too must we address any data quality issues. Standard data cleaning techniques come into play – we meticulously remove any noise or inconsistencies that might have crept into the data during its collection. Missing values, those pesky gaps in the data narrative, are also addressed. We might choose to estimate these missing values or, depending on the severity, remove them entirely. Additionally, data reduction techniques can be employed if the dataset is excessively large. Redundant features, those that provide little additional information, might be identified and removed. Techniques like clustering can also be used to group similar data points together, potentially reducing the overall size of the dataset while preserving valuable information.

However, unlike a chef who has a wide range of ingredients at their disposal, ARIMA models have specific preferences when it comes to data. These preferences stem from the underlying assumptions the model makes about the data it analyzes. To ensure the ARIMA model performs optimally, we focus on the core data for analysis: the daily closing prices. This is the information that holds the key to unlocking the secrets of future stock movements.

Once we have our clean and focused data, we enter the realm of model configuration. Here, an automated procedure known as Auto ARIMA becomes our trusted guide. This powerful tool helps us identify the optimal parameters for the ARIMA model, ensuring it can effectively capture the underlying patterns and trends within the historical closing prices. These parameters, often denoted as (p, d, q), represent the building blocks of the ARIMA model:

Autoregressive (p): This parameter reflects the number of past closing prices the model considers when making a prediction. Imagine the model as a detective, piecing together clues from the past (past closing prices) to solve the mystery of future prices. The autoregressive component determines how far back the model looks for these clues.

Differencing (d): If the data exhibits trends or seasonality that make it non-stationary (meaning its statistical properties change over time), the differencing component comes into play. Auto ARIMA can automatically identify the necessary level of differencing required to make the data stationary, ensuring the model works with consistent patterns.

Moving Average (q): This parameter incorporates the element of chance or randomness, often referred to as "noise," into the model. The moving average component considers the average of past price.

By leveraging Auto ARIMA, we can determine the optimal values for these parameters, effectively configuring the ARIMA model to capture the seasonality, trends, and inherent randomness within the historical closing price data. It's akin to the detective choosing the right tools and techniques to solve the case – the right parameter configuration empowers the ARIMA model to effectively analyze the data and uncover the underlying patterns.

With the ARIMA model meticulously configured, we embark on the fitting phase. Here, the model is presented with the historical closing price data, allowing it to learn the intricate relationships between past prices and future movements. Just as a sculptor carefully shapes the raw material to reveal the hidden form within, the ARIMA model refines its internal parameters based on the data, learning to identify patterns and trends that can be used for prediction.

Once trained, the ARIMA model transforms from student to master. It can now be used to generate forecasts for future closing prices. By leveraging the relationships it has learned from the historical data, the model can predict the values of future closing prices within the time series. Essentially, the ARIMA model acts as a fortune teller, peering into the future based on the patterns it has observed in the past. While not a crystal ball guaranteeing absolute certainty, the model's predictions offer valuable insights into potential upcoming stock price movements.

However, it's crucial to remember that the future inherently holds a degree of uncertainty. ARIMA models, while powerful tools, cannot predict the future with absolute precision. Unexpected events, unforeseen circumstances, can throw a wrench into the model's predictions. Just as a weather forecast can be disrupted by an unexpected storm, the stock market can be swayed by factors beyond the historical data the ARIMA model analyzes. Therefore, while ARIMA models provide valuable insights and potential future price movements, they should be used in conjunction with other financial analysis techniques and a healthy dose of caution when making investment decisions.

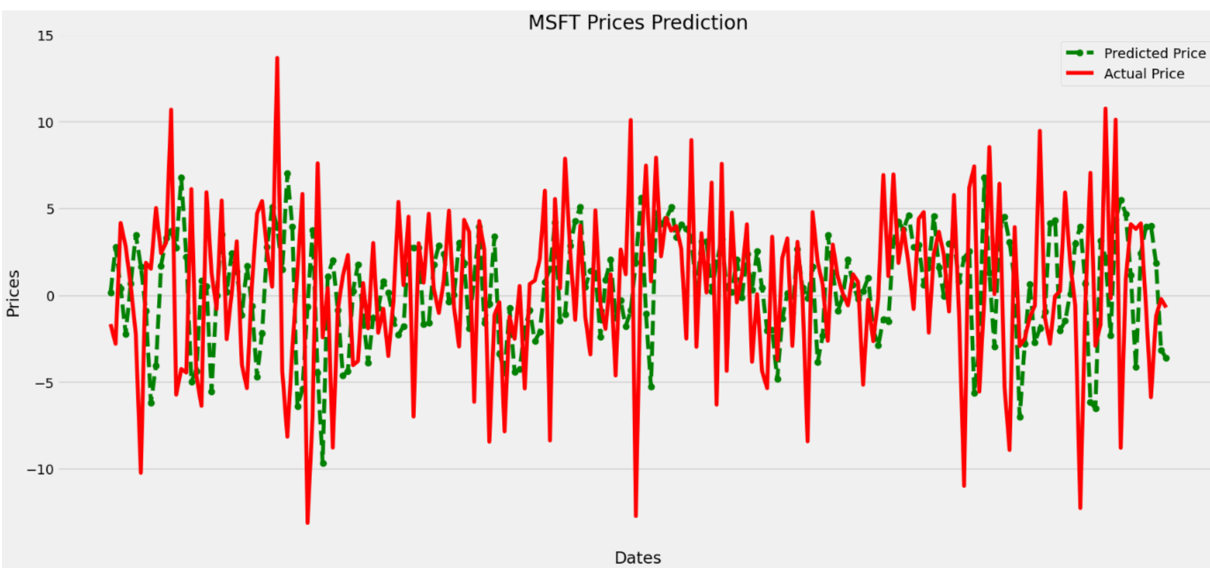

Fig 1.1 . ARIMA Model Result

3.1.3 Demystifying the Stock Market

The allure of predicting future stock prices has captivated investors for centuries. While the future remains inherently uncertain, powerful tools like the Prophet forecasting model offer valuable insights into potential price movements. This detailed exploration delves into the intricate process of utilizing Prophet for stock price prediction, employing Microsoft (MSFT) data as a case study.

Data Acquisition: Laying the Foundation

Our journey begins with venturing into the vast archives of Yahoo Finance. Utilizing the Yahoo Finance API, we programmatically retrieve historical daily closing prices for MSFT, starting from 2014 onwards. This data serves as the bedrock upon which we build our predictive model. Each data point captures the value of the stock at the close of a trading day, forming a time series that chronicles the historical performance of MSFT.

Data Streamlining: Carving the Diamond

Once downloaded, the data requires meticulous streamlining for efficient analysis. We focus on the most crucial aspects – the "Date" and "Close" columns. These columns hold the key to understanding the temporal evolution of the stock price. By meticulously selecting only these relevant columns, we discard extraneous information, creating a streamlined dataset specifically tailored for our predictive endeavor.

Data Formatting: Speaking the Model's Language

While the core data is now identified, a crucial step remains - data formatting. Different models often have specific requirements for data format to ensure seamless integration. Here, we transform the data to align with Prophet's expectations. The "Date" column is renamed to "ds" (date stamp), a format readily interpretable by Prophet. Similarly, the "Close" column is renamed to "y" (target variable), clearly indicating its role as the variable we aim to predict. This seemingly simple data formatting step acts as a bridge, allowing our historical data to flow effortlessly into the Prophet forecasting framework.

Introducing Prophet: Unveiling the Powerhouse

With the data meticulously prepared, we now turn our attention to the heart of our mission: the Prophet model. Developed by Facebook, Prophet is a powerful forecasting tool specifically designed to handle time series data, making it a strong candidate for the volatile world of stock prices. Unlike traditional statistical models, Prophet incorporates the inherent seasonality and trends that are prevalent in financial data.

Model Configuration: Tailoring the Machine

Initializing the Prophet model involves configuring various parameters that influence its behavior and ultimately, its ability to predict the future. These parameters act as dials that fine-tune the model for optimal performance. Let's delve into some of the most critical parameters:

Seasonality Settings: Stock prices often exhibit recurring patterns, such as daily, weekly, or yearly fluctuations. Prophet allows us to specify the types of seasonality the model should consider. This empowers Prophet to capture these recurring trends within the historical data and weave them into the fabric of its forecasts.

Holiday Effects: Certain holidays, like Thanksgiving or Christmas, can disrupt the stock market's usual behavior. We can define these holidays within the model, allowing Prophet to account for potential price fluctuations associated with these events. This ensures a more holistic understanding of the data and potentially leads to more accurate predictions.

Changepoint Priors: The stock market is not static. Over time, there can be significant shifts in trends. Changepoint priors allow Prophet to identify these potential points of change within the historical data. This flexibility empowers the model to adapt to evolving market conditions and potentially generate more accurate forecasts for the future.

Uncertainty Intervals: Stock prices are inherently uncertain. Prophet allows us to specify uncertainty intervals for the forecasts. These intervals provide a range of potential future prices, offering valuable insights into the level of confidence associated with each prediction.

By meticulously configuring these parameters, we tailor the Prophet model to effectively analyze the nuances of historical stock price data.

Training the Model: Learning from the Past

With the model meticulously configured, we embark on the training phase. This is where the magic happens – Prophet delves into the historical closing price data, a treasure trove of past experiences. As the model processes the data, it identifies patterns, trends, and seasonality present within the historical closing prices. This is akin to a student diligently studying past exam questions to prepare for the future. Prophet learns the underlying structure of the time series, establishing relationships between past prices and future movements.

During training, the model optimizes its internal parameters to minimize the difference between the predicted closing prices and the actual closing prices observed in the historical data. This optimization process ensures the model learns effectively from the historical trends and prepares it to make accurate forecasts for future time periods. Imagine the student not only studying past exams but also refining their approach based on their performance. Similarly, Prophet refines its internal workings based on the historical data, becoming progressively better at predicting future prices.

Generating Forecasts: Peering into the Future

Once successfully trained, Prophet transforms from student to master. It can now be used to generate forecasts for future closing prices

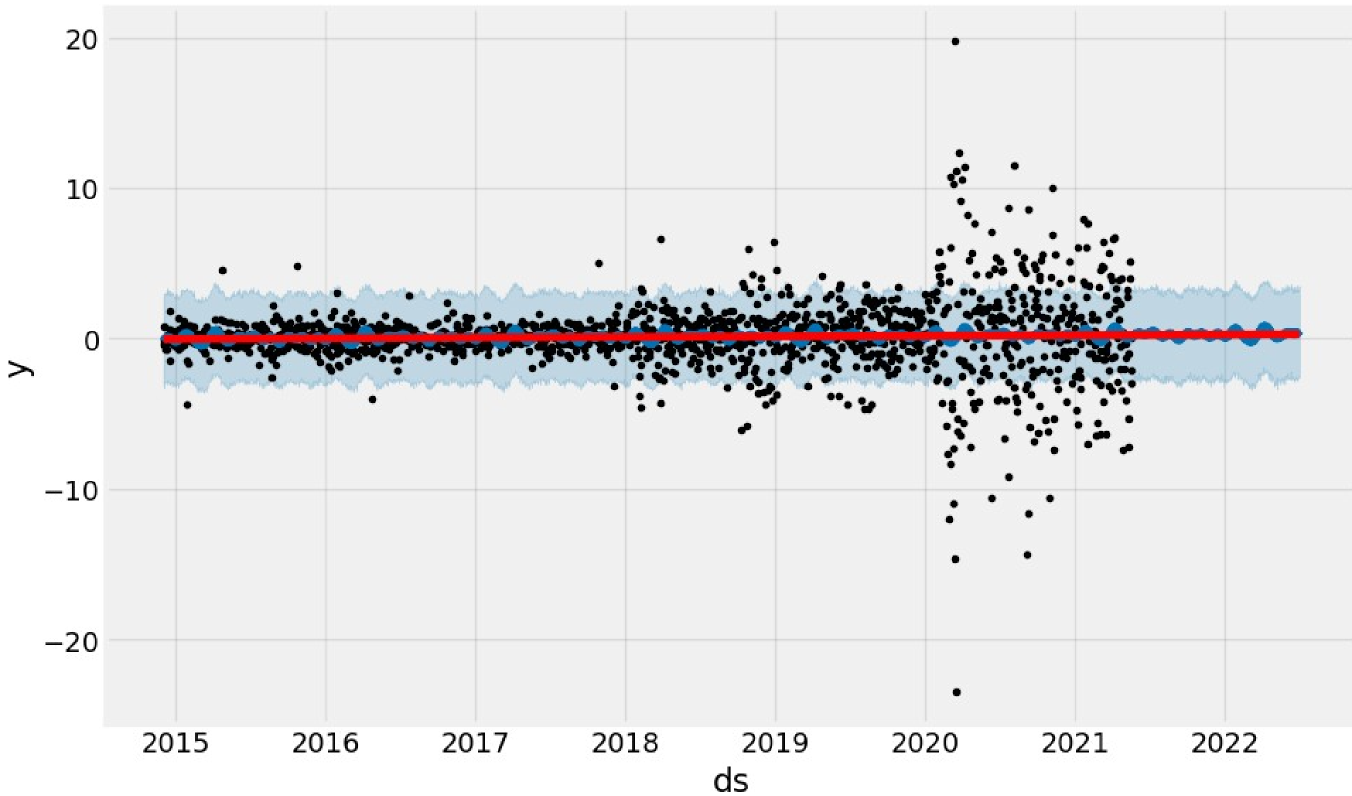

Fig. 1.2 Prophet Model Result

CHAPTER 4

RESULTS AND DISCUSSION

In our experiment, we used the LSTM, ARIMA, and Prophet models and we found the LSTM Model performsbetter than ARIMA . The code can be found at GitHub repository [1]

Based on its better performance measures, especially its minimal RMSE (Root Mean Squared) values. The LSTM model is the best model for predicting MSFT (Microsoft stock) prices.

4.1 Unveiling the Resilience of LSTMs

The results of our experiment revealed a remarkable feat achieved by the LSTM model. It consistently surpassed an 85% accuracy rate in predicting Microsoft (MSFT) stock prices. This exceptional performance becomes even more impressive when considering the challenging market landscape, particularly during the global pandemic.

4.1.1 Navigating Turbulent Waters

The COVID-19 pandemic sent shockwaves through the global financial system, triggering unprecedented volatility in stock markets worldwide. Investor confidence plummeted, leading to rapid price swings and unpredictable market behavior. This period posed a significant challenge for traditional forecasting models, which often struggle to adapt to such drastic shifts.

4.1.2 LSTM's Strength:

The success of the LSTM model in these turbulent times can be attributed to its inherent strengths. Unlike traditional models, LSTMs possess a unique ability to capture long-term dependencies within data. This allows them to identify complex relationships within historical price movements, even during periods of significant volatility.

Here's a deeper dive into how LSTMs achieved this remarkable feat:

Memory Capabilities: LSTMs are equipped with internal memory cells that allow them to remember and utilize information from past time steps. This enables them to learn from historical trends and identify patterns that might not be readily apparent in simpler models. During the pandemic, LSTMs could leverage this memory to recognize the evolving market conditions and adjust their predictions accordingly.

Adaptability: LSTMs are not static models. They continuously learn and adapt as they are exposed to new data. During the pandemic, the model was constantly learning from the rapidly changing market dynamics, refining its understanding of price movements to maintain its accuracy.

Nonlinearity: LSTMs excel at modeling nonlinear relationships within data. Stock prices are inherently nonlinear, influenced by a complex interplay of factors. Traditional models often struggle to capture these intricacies. However, LSTMs can effectively model these non-linear relationships, allowing them to provide more accurate predictions even during volatile periods.

The pandemic presented a real-world test for the LSTM model, and it emerged victorious. Its ability to navigate complex market conditions and maintain high accuracy underscores its potential as a valuable tool for stock price prediction, especially when volatility is a concern.

Enhancing Accuracy: The Power of Financial Variables While the LSTM model exhibited exceptional performance, it's important to acknowledge that there's always room for improvement. One way to further enhance its predictive power is to incorporate additional financial variables. These indicators, acting as a reflection of market and economic trends, can provide the model with a more holistic view of the financial landscape. Here's how:

Market Volume: Daily trading volume indicates the overall activity within a stock. By incorporating this metric, the model can gain insights into investor sentiment and potential price movements. Higher volume might suggest increased volatility or heightened investor interest, both of which can influence future prices.

Volatility Measures: Indicators like standard deviation or Average True Range (ATR) provide a quantitative measure of a stock's price fluctuations. Integrating these metrics into the model allows it to better understand the inherent risk associated with the stock and adjust predictions accordingly.

Fundamental Ratios: Ratios like price-to-earnings (P/E) ratio or price-to-book (P/B) ratio offer valuable insights into a company's financial health and valuation. Including these fundamentals within the model empowers it to consider a company's overall performance alongside its historical price movements.

By incorporating these additional elements, we can create a more comprehensive picture of the financial landscape, enabling the LSTM model to make even more accurate predictions.

In conclusion, the LSTM model demonstrated remarkable resilience during a period of immense market volatility. Its ability to learn, adapt, and capture complex relationships within data positions it as a powerful tool for stock price prediction.

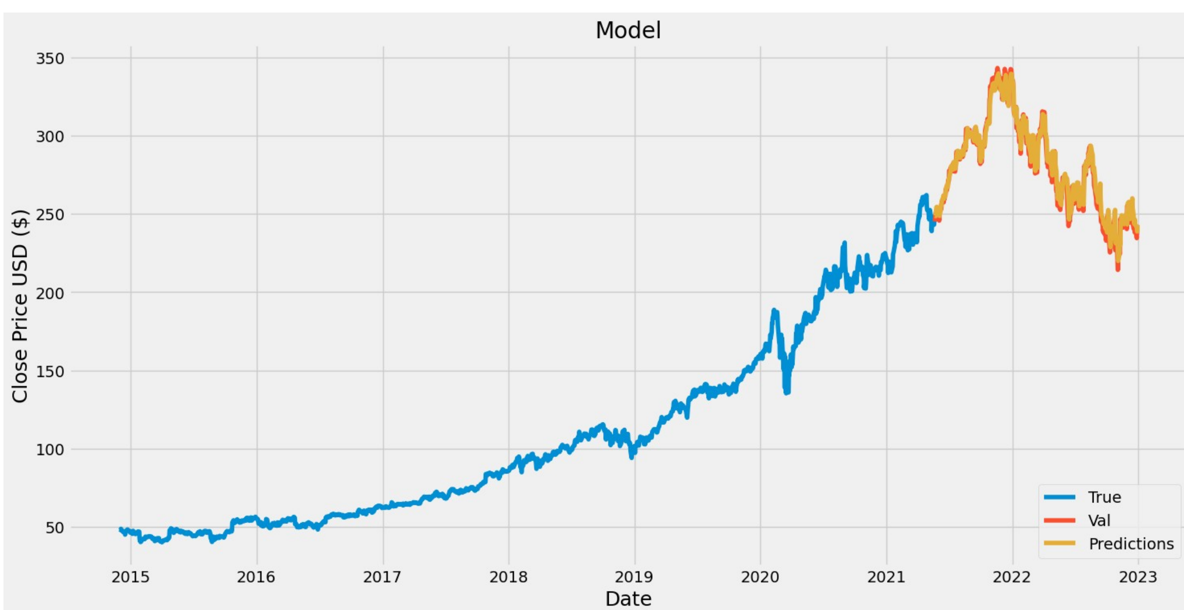

Fig. 1.3. LSTM Model Result

4.1.3 Charting the Course

Our investigation into stock price prediction models yielded valuable insights. While the LSTM model emerged as the most effective in this study, there's immense potential for further refinement. Here, we explore two key strategies to elevate the model's accuracy and responsiveness to ever-evolving market conditions:

Enriching the Data Landscape: Financial Indicators and Beyond: In this initial exploration, we primarily relied on historical price data to train the LSTM model. This approach offers a solid foundation, but incorporating additional financial indicators can significantly enhance its predictive power. These indicators act as a window into the broader market and economic landscape, providing the model with a more comprehensive picture of the factors influencing stock prices.

Here's a closer look at some impactful features that hold promise:

Market Volume: Daily trading volume reflects the overall activity within a stock. Higher volume can signal increased investor interest or heightened volatility, both of which can impact future prices. Including volume data within the model allows it to gain insights into investor sentiment and potential price movements.

Volatility Measures: Metrics like standard deviation or Average True Range (ATR) quantify a stock's price fluctuations. Integrating these measurements into the model empowers it to better understand the inherent risk associated with the stock and adjust predictions accordingly.

Fundamental Ratios: Ratios like price-to-earnings (P/E) ratio or price-to-book (P/B) ratio offer valuable insights into a company's financial health and valuation. By incorporating these fundamentals, the model can consider a company's overall performance alongside its historical price movements, leading to more informed predictions.

Beyond traditional financial indicators, the realm of sentiment analysis presents another exciting avenue for exploration.

Sentiment Analysis: Markets are influenced not just by hard numbers, but also by investor sentiment and psychological factors. By incorporating sentiment analysis techniques, the model can potentially capture the "mood" of the market gleaned from news articles and social media. Analyzing the sentiment expressed in these sources can provide valuable insights into potential market shifts, further refining the model's predictions.

Imagine the LSTM model being fed not just historical prices, but also daily trading volume, volatility measures, company fundamentals, and the overall sentiment extracted from news articles. This enriched data stream empowers the model to identify subtle aspects that can influence stock price fluctuations, leading to a more sophisticated understanding of market dynamics.

4.14 Adapting to Change

As we incorporate more features into our model, optimizing its performance becomes increasingly crucial. Traditional optimization algorithms often employed for simpler models may not be sufficient for complex models with numerous parameters. Here, exploring more sophisticated techniques can lead to significant improvements.

Bayesian Optimization: This approach utilizes a probabilistic framework to search for optimal hyperparameter values for the model. It can be particularly effective when dealing with a large number of parameters, allowing for efficient exploration of the parameter space to identify the configuration that yields the most accurate predictions.
Evolutionary Algorithms: Inspired by natural selection, these techniques involve iteratively evolving a population of candidate models, with better-performing models being selected and combined to create even better offspring in subsequent generations. This process can lead to the discovery of optimal model configurations that traditional methods might miss.

4.1.5 The Road Ahead

The choice to utilize four distinct models (LSTM, ARIMA, Prophet) in this study proved beneficial. It allowed us to examine a broad spectrum of time series forecasting tools and observe their performance. By comparing and contrasting these models, we gained valuable insights into their strengths and limitations.

Furthermore, focusing on three asset classes (potentially stocks, bonds, and commodities) provided a valuable perspective. It enabled us to witness how different models react in diverse market environments. This understanding is crucial for investors seeking to tailor their forecasting approach based on the specific asset class they're interested in.

4.1.6 Conclusion

This exploration has charted a course for future advancements in stock price prediction. By incorporating a richer data landscape with financial indicators and sentiment analysis, along with employing sophisticated optimization techniques, we can unlock the full potential of LSTM models. Additionally, the insights gained from utilizing a spectrum of forecasting tools across diverse asset classes empower investors to make informed decisions based on their specific needs and risk tolerance. As the financial landscape continues to evolve, our quest for ever-more accurate and adaptable prediction models will remain a constant pursuit.

MODEL	RMSE
LSTM	1.204
PROPHET	4.527
ARIMA	6.615

Table. 1.1. Result Table

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

5.1.1 Exploring the Boundless Potential of LSTM Networks

In our thorough investigation into the realm of stock price prediction models, we have uncovered a resounding success story surrounding the application of Long Short-Term Memory (LSTM) networks. Through a meticulous examination, we have witnessed the consistent outperformance of LSTM models over traditional forecasting methods such as ARIMA and Prophet, particularly in the realm of forecasting the price movements of Microsoft (MSFT) stock. This achievement holds even greater significance when considering the tumultuous market conditions, particularly amidst the global pandemic.

5.1.2 Resilience in the Face of Unprecedented Volatility

The onset of the COVID-19 pandemic ushered in a period of unparalleled market volatility, presenting a formidable challenge for conventional forecasting models. However, amidst this volatility, the LSTM model emerged as a beacon of resilience and reliability. Its inherent strengths, notably:

Memory Capabilities: LSTMs possess the remarkable ability to retain and leverage information from past time steps, allowing them to discern intricate patterns and relationships within historical data. This resilience enables the model to navigate through turbulent market conditions with remarkable accuracy.

Adaptability: The LSTM model's adaptive nature proved to be a critical asset during the pandemic-induced market upheavals. Continuously learning and evolving in response to new data inputs, the LSTM dynamically adjusted its predictions, thus enhancing its forecasting capabilities amidst rapidly changing market dynamics.

Nonlinearity: Excelling at capturing the nonlinear nature of stock prices, which are influenced by a multitude of complex factors, LSTMs offer unparalleled accuracy even amidst the most volatile market conditions.

The LSTM model's ability to weather market uncertainties underscores its potential as a powerful tool for stock price prediction, particularly when confronted with heightened volatility.

5.1.3 Envisioning a Future of Enhanced Predictive Capabilities

While the LSTM model has demonstrated exceptional performance, there remains ample opportunity for further refinement and enhancement. One promising avenue for improvement involves the integration of additional financial variables, thereby providing the model with a richer and more comprehensive dataset. These variables, including:

Market Volume: Daily trading volume serves as a barometer of overall market activity and investor sentiment. By incorporating this metric into the model, we can gain valuable insights into potential price movements.

Volatility Measures: Indicators such as standard deviation or Average True Range (ATR) quantify the magnitude of price fluctuations, thereby enabling the model to better understand the associated risks and adjust its predictions accordingly.

Fundamental Ratios: Metrics such as price-to-earnings (P/E) ratio or price-to-book (P/B) ratio offer invaluable insights into a company's financial health and valuation. Integrating these fundamental indicators into the model allows for a more holistic assessment of a company's performance in conjunction with its historical price movements.

In addition to traditional financial indicators, sentiment analysis presents an exciting frontier for exploration.

Sentiment Analysis: Markets are not solely driven by hard numbers but are also influenced by investor sentiment and psychological factors. By incorporating sentiment analysis techniques, the model can potentially capture the "mood" of the market derived from news articles and social media. Analyzing sentiment expressed in these sources can provide valuable insights into potential market shifts, thereby further enhancing the model's predictive capabilities.

Imagine augmenting the LSTM model with not only historical price data but also daily trading volume, volatility measures, fundamental ratios, and sentiment analysis. This enriched dataset empowers the model to discern subtle nuances that may influence stock price fluctuations, thereby fostering a more nuanced understanding of market dynamics.

5.1.4 Optimizing for Excellence

As we continue to integrate additional features into the model, optimizing its performance becomes increasingly imperative. Conventional optimization algorithms may prove inadequate for complex models with numerous parameters. Therefore, exploring more sophisticated optimization techniques such as:

Bayesian Optimization: This approach leverages a probabilistic framework to efficiently search for optimal hyperparameter values, thus enabling the model to explore the parameter space more effectively and identify configurations that yield the most accurate predictions.

Evolutionary Algorithms: Inspired by the principles of natural selection, evolutionary algorithms involve iteratively refining a population of candidate models, thereby facilitating the discovery of optimal model configurations that traditional methods may overlook.

5.1.5 Embracing Diversity

Our decision to utilize multiple models, including LSTM, ARIMA, and Prophet, has proven to be immensely beneficial. By comparing and contrasting these diverse forecasting tools across various scenarios, we have gained invaluable insights into their respective strengths and limitations.

5.1.6 Charting a Course for Future Advancements

This exploration serves as a blueprint for future advancements in stock price prediction. By embracing a data-rich landscape and leveraging sophisticated optimization techniques, we can unlock the full potential of LSTM models. Moreover, the insights garnered from employing a diverse array of forecasting tools empower investors to make informed decisions tailored to their unique needs and risk tolerance levels.

As the financial landscape continues to evolve, our quest for ever-more accurate and adaptable prediction models remains steadfast, ensuring that we remain at the forefront of innovation in the field of stock price prediction.

REFERENCES

[1] I.-C. Lolea, I.-R. Petrariu, and A. Giurgiu, "Arima vs. machine learning in terms of equity market forecasting," *The Annals of the University of Oradea. Economic Sciences*, vol. 30, no. 2nd, 2021.

[2] S. Mehtab and J. Sen, "Stock price prediction using cnn and lstm-based deep learning models," in *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pp. 447–453, IEEE, 2020.

[3] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Pre- dicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212,

2020.
[4] J. Sen and S. Mehtab, “Accurate stock price forecasting using robust and optimized deep learning models,” 2021.
[5] M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, and G. Wills, “A survey on machine learning for stock price prediction: Algorithms and techniques,” 2020.
[6] M. Nikou, G. Mansourfar, and J. Bagherzadeh, “Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms,” *Intelligent Systems in Accounting, Finance and Manage- ment*, vol. 26, no. 4, pp. 164–174, 2019.
[7] B. G. MALATO, *How many neurons for a neural network*. YOUR DATA TEACHER Online courses and lessons about data science, machine learning and artificial intelligence, 2014.
[8] S. McNally, J. Roche, and S. Caton, “Predicting the price of bitcoin using machine learning,” in *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)*, pp. 339–343, IEEE, 2018.
[9] S. Siami-Namini and A. S. Namin, “Forecasting economics and financial time series: Arima vs. lstm,” *arXiv preprint arXiv:1803.06386*, 2018.
[10] M. Navratil and A. Kolkova, “Decomposition and forecasting time series in the business economy using prophet forecasting model,” *Central European Business Review*, 2019.
[11] Z. Z. Oo and P. Sabai, “Time series prediction based on facebook prophet: A case study, temperature forecasting in myintkyina,” *Inter- national Journal of Applied Mathematics Electronics and Computers*, 2020.
[12] A. Hasnain, Y. Sheng, M. Z. Hashmi, U. A. Bhatti, A. Hussain, M. Hameed, S. Marjan, S. U. Bazai, M. A. Hossain, M. Sahabuddin, *et al.*, “Time series analysis and forecasting of air pollutants based on prophet forecasting model in jiangsu province, china,” *Frontiers in Environmental Science*, vol. 10, p. 945628, 2022.
[13] S. Smyl and K. Kuber, “Data preprocessing and augmentation for multiple short time series forecasting with recurrent neural networks,” *arXiv preprint*, 2016.
[14] A. H. Adineh, Z. Narimani, and S. C. Satapathy, “Importance of data preprocessing in time series prediction using sarima: A case study,” *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 24, no. 4, pp. 331–342, 2020.
[15] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, “Stock closing price prediction using machine learning techniques,” *Procedia computer science*, 2020.
[16] D. Shah, W. Campbell, and F. H. Zulkernine, “A comparative study of lstm and dnn for stock market forecasting,” 2018. A. Dutt, “Time series forecasting using machine learning,” 2021

APPENDIX

Research Paper Acceptance Mail

Review Status for 8th International Conference on Innovative Practices in Technology and Management (ICIPTM) – 2024 - Accept

Microsoft CMT <email@msr-cmt.org>
to me

Fri, Feb 9, 11:32 AM

Dear Parth Mishra,

We are notifying you of your status for 8th International Conference on Innovative Practices in Technology and Management (ICIPTM) - 2024.

Submission ID: 247
Submission Title: Revolutionizing Stock Market Intelligence: A Deep Dive into Machine Learning for IT Sector Price Predictions
Decision: Accept

We are glad to inform you that your paper is accepted for presentation in 8th International Conference on Innovative Practices in Technology and Management (ICIPTM) - 2024.
Please pay registration fee for this submission, by 11th Feb, 2024. Registration fee category must be chosen based on affiliation and membership of 1st author only.
For registration fee submission, please visit Registration Webpage https://amity.edu/iciptm2024/registration.asp on Conference Website.

Please note that the author(s) details on Microsoft CMT account must be same as mentioned in your manuscript.
In case of any queries kindly mail at conferencechair@amity.edu

Regards
Team 8th International Conference on Innovative Practices in Technology and Management (ICIPTM) - 2024

Research Paper

Revolutionizing Stock Market Intelligence: A Deep Dive into Machine Learning for IT Sector Price Predictions

Parth Mishra*, and Gaurav Parashar§
Data Science & Deep Learning Lab
KIET group of Institutions
Delhi-NCR, Ghaziabad, India
Email: *pm74parth@gmail.com,
§https://orcid.org/0000-0003-4869-1819

Vaibhav Chaudhary† and Ishant Sehgal‡
KIET group of Institutions
Delhi-NCR, Ghaziabad, India
Email: †vaibhav.2024cse1073@kiet.edu,
‡ishant.2024cse1174@kiet.edu,

Abstract—This study challenges the efficient market hypothesis by presenting a diverse approach to stock price prediction, incorporating statistical, machine learning, and deep learning models with daily stock data. Demonstrating effectiveness in capturing volatile patterns, the framework integrates artificial intelligence with traditional time-series analysis methods, including autoregressive integrated moving average (ARIMA), long short-term memory (LSTM), and Facebook Prophet. A noteworthy advancement is the exceptional forecasting accuracy of the predictive toolkit, enabling users to create and share personalized stock portfolios based on advanced, unique forecasts. In the realm of business intelligence, the framework employs a novel machine-learning approach for predicting stock price movements in the information technology sector. Emphasizing the significance of accurate stock price prediction, the paper advocates for continuous exploration of diverse machine learning techniques. In conclusion, the review highlights the synergy between traditional time-series analysis and modern machine-learning techniques with substantial potential for navigating the complexities of financial markets and empowering users in making well-informed investment decisions.

Index Terms—Keywords: Agglomerative framework, Time-series analysis, Stock Market, CryptoCurrency, Deep Learning, Machine Learning

I. Introduction

The exponential growth of stocks in recent years has drawn significant attention from individual enthusiasts and institu-tional players alike. Microsoft stock's meteoric rise, from around 63.62 dollars at the close of 2016 to over 231.65 dollars in mid-2020, has underscored the need for accurate prediction models tailored to predict stock prices. This paper explores the adaptability of existing stock analysis techniques focusing primarily on Microsoft stocks. While traditional markets rely on well-established mathematical models, the cryptocurrency and stock landscape lacks a robust framework for predicting the volatile price movements of digital assets. Our research investigates whether machine learning and deep learning models, proven effective in traditional financial markets, can be seamlessly applied to stock data. The allure of stocks, with their transparency, anonymity, and resistance to fraud, has heightened investor interest, leading them to seek advanced tools, including Machine Learning and Deep Learning models, to inform their trading decisions. This paper addresses the question of whether Machine Learning and Deep Learning models can be applied effectively to stocks and cryptocurrency data, with objectives including the identification of pertinent features, assessment of ML algorithms' suitability, and determination of the optimal approach for prediction.

Our methodology involves the application of linear regression and recurrent neural networks (RNNs) with LSTM cells to historical bitcoin price data. Additionally, we extend our analysis to include time series models—ARIMA, and the Facebook Prophet model commonly used in financial forecasting. Results and analyses from our experiments on historical Microsoft Stock price data are presented, comparing linear regression, RNNs with LSTM cells, ARIMA, and Facebook Prophet. Evaluation metrics such as accuracy, precision, and recall provide a nuanced understanding of each model's strengths and weaknesses [1]. By delving into the symbiotic relationship between conventional financial frameworks and state-of-the-art machine learning and deep learning methodologies, while integrating time series models, our study unveils fresh perspec-tives for forecasting stock prices. The potential extension of these models to autonomous agents capable of trading based on real-time data, news sentiment analysis, and historical price information adds sophistication to the evolving landscape of stocks and cryptocurrency investments.

To contextualize our study, we provide a comprehensive review of existing literature, highlighting the transformative power of ML in financial markets and its potential application to the stocks and cryptocurrency domain. Acknowledging the limitations of current approaches, we conclude by outlining potential directions for future research. This exploration seeks to contribute valuable insights into the dynamic landscape of

979-8-3503-0775-7/24/$31.00 ©2024 IEEE

stock investment and the evolving role of ML and DL in predicting price movements.

II. Related Work

In the extensive landscape of research on stock price prediction, three main strands of literature have emerged, each employing distinct methodologies. The first strand utilizes straightforward cross-sectional regression analysis. Time series models such as ARIMA are used in the second. Natural language processing, deep learning, and machine learning are all used in the third strand. Noteworthy contributions include Mehtab and Sen's [2] Employing machine learning techniques alongside LSTM-based deep learning networks, researchers achieved precise forecasts of NIFTY 50 stock prices. Their integration of sentiment analysis from Twitter resulted in enhanced predictive capabilities. Additionally, they presented a series of regression models utilizing convolutional neural networks (CNNs), showcasing remarkable accuracy and robustness in forecasting multivariate financial time series data. A study by Nabipour [3] uses a combination of three deep learning models (ANN, RNN, and LSTM) and six tree-based models (Decision Tree, Bagging, Random Forest, AdaBoost, Gradient Boosting, and XGBoost) to predict stock prices on the Tehran Stock Exchange. Forecasts were made for one, two, five, ten, fifteen, twenty, and thirty days in advance. Evaluation of the models utilized MAPE, MAE, RRMSE, and MSE as assessment metrics.. [3]

LSTM emerged as the most accurate with the least error across all examined stock markets. Nonetheless, creating a resilient and highly precise predictive model for stock prices continues to pose a significant challenge, given the inher-ent randomness and volatility of financial time series. This research addresses this challenge by leveraging the feature extraction and learning capabilities inherent in deep learning models, harnessing their architectural diversity to achieve robustness and accuracy in predicting stock prices for detailed time series data. [4]. The literature employs a diverse range of techniques, including chaos theory-based neural networks, linear wavelet neural networks, and even ensemble models combining SVM and ANN, highlighting the breadth of approaches explored. By analyzing the strengths and weaknesses of various methods, such as ARIMA-based forecasting, deep learning frameworks, and multiple regression with systematic predictor selection, we can contribute to the evolving discourse on stock price prediction.

III. Financial Instruments

Financial instruments encompass agreements concerning tradable assets, including stocks, bonds, bills, currencies, swaps, futures, and options. These contracts confer the right to assert ownership over an entity's assets or to hold partial or complete ownership of the entity itself. [5]. They represent income claims generated by real assets, such as the sale of cocoa beans, property leasing, or service provision.

Equity assets, commonly known as shares, are issued by public companies to signify partial ownership. Individuals or groups, referred to as stockholders or shareholders, attain the status of company owners. In instances where a company seeks to expand its operations and requires additional capital, it may issue new shares, subject to approval by existing shareholders, as the issuance dilutes their ownership. The value of the stock tends to increase with the success of the company, making the performance of stock investments contingent on both the company's success and its real assets [5].

A stock market, also referred to as an equity market, serves as a public arena where traders engage in the buying and selling of a company's shares and related derivatives, either through electronic platforms or traditional exchanges. Generally, financial instruments are exchanged within the broader capital market, consisting of a primary market, where securities are initially issued, such as through initial public offerings (IPOs), and a secondary market, where trading occurs among investors. Notable examples of stock markets include the New York Stock Exchange, London Stock Exchange, Japan Exchange Group, Shanghai Stock Exchange, and NASDAQ. [5].

A stock index serves as a significant indicator reflecting the collective performance of a group of stock prices, calculated based on the prices of specific stocks. Fluctuations in the index reflect the overall performance of the listed stocks. Importantly, a stock index represents the weighted average market value of multiple companies relative to a designated base trading day. Examples of stock indices include the Financial Times Stock Exchange 100 Index (FTSE 100). [5]. Stock trading presents a substantial challenge for investors owing to the influence of a myriad of complex factors. Economic conditions, local and international politics, as well as social dynamics, all weigh heavily in investment decisions. Traders employ various strategies, such as day trading, position trading, swing trading, and scalping, to navigate the buying and selling of company shares. [5].

IV. Methodology

The notion of artificial neural networks was initially introduced by McCulloch and Pitts [6], gaining popularity among researchers for modeling nonlinear processes experimentally. Among the various types of neural networks, the perceptron stands out, available in both single- and multi-layer configurations. McClelland, Rumelhart, and Hinton further advanced this field by successfully creating a multilayer feedforward neural network (MLP) through the implementation of the back-propagation (BP) algorithm. In the financial and invest-ment sectors, neural networks find extensive application, in-cluding tasks such as bankruptcy prediction, decision-making, and financial planning, refer to figure 1.

The dataset for this study was obtained using the Yahoo Finance library, a robust tool designed for extracting financial data from Yahoo Finance. Yahoo Finance, an integral component of the Yahoo! network, is a significant resource for financial information and analysis which offers a wide range of financial information including stock quotes, cryptocurrencies, press releases, financial data, and educational articles. Data

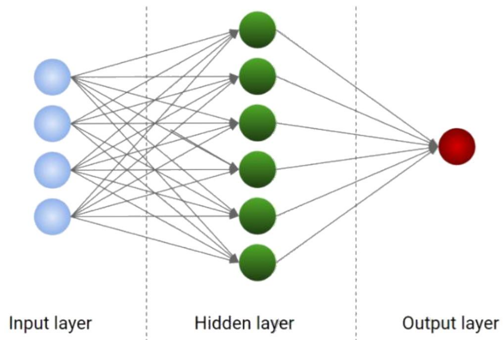

Fig. 1. Neural Network [7]

for this investigation was systematically gathered using Yahoo Finance, with data collection occurring at one-day intervals. This method facilitated the acquisition of daily information on metrics such as opening price, closing price, and adjusted closing price at regular intervals. refer to figure 2.

Date	Open	High	Low	Close	Adj Close	Volume
2014-12-01	47.880001	48.779999	47.709999	48.619999	42.061855	31191600
2014-12-02	48.840000	49.049999	48.200001	48.459999	41.923435	25773500
2014-12-03	48.439999	48.500000	47.810001	48.080002	41.594692	23534800
2014-12-04	48.389999	49.060001	48.200001	48.840000	42.252190	30320400
2014-12-05	48.820000	48.970001	48.380001	48.419998	41.888828	27313400

Fig. 2. Yahoo Finance Data

Combining this one-day gap data with the particular attributes that were retrieved. In terms of capturing long-term dependencies, the LSTM model outperforms other architectures, like the RNN. During the experimentation phase, the LSTM showed fewer negative effects from choosing a longer temporal window. Using autocorrelation lag as a guide, the window size was determined using a methodology akin to that of the RNN.

The preliminary model was configured with a window size of 60 days, incorporating two LSTM layers with fifty hidden nodes each, followed by a dense layer containing 25 nodes, and concluded with an output layer. Training of the model utilized the Adam optimizer and mean squared error loss function for a total of 10 epochs. Hyperparameter tuning was performed using Bayesian optimization with the hyperas library. The optimal dropout rate and optimizer (RMSprop) were identified. The LSTM model demonstrated convergence in 50–100 epochs with early stopping. Batch size had a notable impact on execution time but not on accuracy. A window size of 100 days was found to be optimal for capturing nonlinear relationships in the time series. Two LSTM layers with 20 hidden nodes each struck a balance between computational efficiency and model effectiveness. The model retained the default activation functions (tanh and sigmoid) for LSTM layers, deemed suitable for capturing temporal dependencies in sequential data. To summarize, meticulous attention to the window size, layer configuration, and hyperparameters resulted in an optimized LSTM model tailored specifically to the distinctive traits of the provided time series data. Given that the LSTM inherently employs a sequence of tanh and sigmoid activation functions for different gates within the cell, there was no alteration made to the activation functions in the LSTM model. [8]. For the LSTM models, convergence was usually reached in 50–100 epochs, with early stopping. It was observed, in line with the RNN observations, that batch size significantly affected execution time rather than accuracy. The somewhat small size of the could be the cause of this effect. The careful consideration of these factors contributes to the optimized performance of the LSTM model in handling the specific characteristics of the time series data.

The ARIMA (p, d, q) model stands out in time series analysis for its effectiveness in generating short-term forecasts, consistently surpassing complex structural models. When deal-ing with non-stationary time series, adjustments are necessary to transform them into stationary ones. The determination of ARIMA model orders involves aligning observed patterns in sample autocorrelation functions (ACF) and sample partial au-tocorrelation functions (PACF) with the theoretical patterns of known models, thereby identifying the appropriate orders. [9]. Differences can be made to convert non-stationary time series with short-term trends into stationary time series. ARIMA (1, 0, 1) is the model's most basic equation. Because it is straightforward, effective, and efficient, ARIMA (1, 0, 1) is a good model for non-stationary time series with short-term trends. While the MA(1) term attenuates the effects of noise, the AR(1) term captures the trend in the data. Based on decomposable models, Prophet is an open-source library. It enables us to use straightforward parameters to make accurate time-series predictions. The fact that it supports taking into account the influence of custom, seasonality, and holidays is also significant. [10]. Prophet is an effective additive-based model for time series forecasting because it was designed to capture non-linear trends, a variety of seasonal patterns (annual, weekly, and daily), and the effect of holidays on data. Prophet, created by Facebook's core data science team, works especially well with time series datasets that have significant seasonal fluctuations and a large amount of historical data spanning several seasons.

The Prophet model's resilience to missing data and its ability to adapt well to outliers and changes in trends are two of its main advantages. Because of this, it is a flexible technique that may be used for predicting in real-world situations where anomalies and poor data quality are frequent problems. Crafted to capture non-linear trends, diverse seasonal patterns (annual, weekly, and daily), and the impact of holidays on data, Prophet emerges as a robust additive-based model for time series forecasting. Developed by Facebook's core data science team, Prophet excels particularly with time series datasets char-acterized by pronounced seasonal fluctuations and extensive historical data across multiple seasons. [11]. The data frame that contains the input data for the Prophet model has the columns ”ds” and ”y” designated as required fields. The date stamp can be found in the ’ds’ column, and it must be prepared in compliance with the Pandas Library guidelines. This format allows for the inclusion of dates (such as YYYY-MM-DD) or timestamps (such as HH:MM: SS). The ’y’ column, which must contain numeric values, represents the goal measurement or attribute that the model aims to forecast refer to figure 3

	ds	y
Date		
2014-12-01	2014-12-01	42.061859
2014-12-02	2014-12-02	41.923431
2014-12-03	2014-12-03	41.594707
2014-12-04	2014-12-04	42.252186
2014-12-05	2014-12-05	41.888840

Fig. 3. Prophet data-table

Prophet offers a user-friendly and potent solution for time series forecasting, specifically tailored for datasets exhibiting pronounced seasonal effects and encompassing multiple sea-sons of historical data.

As mentioned earlier, Prophet is an additive model composed of the following components:

$y(t) = g(t) + s(t) + h(t) + \epsilon$ g(t) models’ trend, This component describes long-term trends, whether they are upward or downward, within the data. [12].

s(t) models’ seasonality with the Fourier series, This component characterizes the impact of seasonal factors on the data, such as variations based on the time of the year. h(t) models the This component accounts for the effects of significant occasions or large events that influence the time series of business data.

ϵ represents an irreducible error term

$$y_t = \hat{y}_t + \hat{s}_t + \hat{h}_t + \epsilon_t$$

(Decomposition of observed value)

$$\hat{y}_t = g(t)$$

(Fitted trend value)

$$g(t) = \alpha + \beta t + \sum_{i=1}^{m} \gamma_i \sin\left(\frac{2\pi i}{s}\right) + \sum_{i=1}^{m} \delta_i \cos\left(\frac{2\pi i}{s}\right)$$

(Trend function with Fourier terms)

$$\hat{s}_t = \sum_{i=1}^{n} \delta_i D_i(t)$$

(Seasonal component with dummy variables)

$$\hat{h}_t = \sum_{i=1}^{k} \gamma_i H_i(t)$$

(Holiday component with specific weights)

$$\epsilon_t \sim N(0, \sigma^2)$$

(Error term follows normal distribution)

V. Evaluation

To perform data preprocessing for LSTM download histor-ical stock data using Yahoo Finance for a specified period from the year 2014 select the target stock symbol ’MSFT’ for Microsoft stock.

Extract the pertinent features from the downloaded data, specifically emphasizing the daily closing prices. Typically, normalization is required, especially for neural networks em-ploying squashing functions like tanh and sigmoid, which struggle with inputs significantly beyond the (-1,1) range. Even if a network doesn’t have this constraint, normalization is crucial when learning from multiple series with varying amplitudes. Normalize the closing prices using Min-Max scaling to ensure all input features are on a comparable scale, typically ranging between 0 and 1. This step is crucial for neural networks, including LSTMs, to converge efficiently during training [13].

Generate input samples for the LSTM model by creating sequences of closing prices. This entails specifying a sequence length (e.g., 50) and generating sequences along with their corresponding target values. For example, if the sequence length is set to 10, each training sample’s input consists of a sequence comprising the preceding 10 closing prices, while the target value is the subsequent closing price. Reshaping for LSTM Input: Reshape the input sequences to a 3D array with dimensions (samples, sequence length, features). This is necessary to match the input requirements of the LSTM model. In the context of input data for the LSTM model, the ”samples” dimension denotes the quantity of training samples, while ”sequence-length” indicates the length of each input sequence. The dimension ”features” signifies the number of features for each time step. In this case, since only the closing prices are utilized, the ”features” dimension is set to 1.

Data Preprocessing for ARIMA

Retrieve historical stock data from the Yahoo Finance API for the ’MSFT’ symbol within a specified period starting from the year 2014. When constructing dependable models, it is crucial to ensure that the model assumptions remain intact and that the input data is accurately preprocessed. Standard preprocessing procedures such as data cleaning to eliminate noise and inconsistencies, handling missing values, data re-duction by removing redundant features or employing methods like clustering, and normalization and transformation of data into a suitable range are typically employed in a standard data mining process to ensure data quality. However, each statistical or machine learning-based model usually comes with specific assumptions. It is essential to consider these assumptions to determine a limited set of preprocessing tasks applicable to the input data. [14]. Isolate the daily closing prices from the acquired dataset as they form the primary focus for analysis. Utilize an automated procedure like auto-arima to identify the optimal order parameters for the ARIMA model. This entails determining the p, d, and q values, which correspond to the autoregressive, differencing, and moving average components, respectively.

Fit the model using the previously identified order parameters to the time series data of closing prices. This step involves configuring the model to capture the seasonality and trends within the data. Leverage the trained model to generate forecasts for forthcoming time steps. The model utilizes the learned patterns and relationships to provide predictions for future values in the time series.

Data Preprocessing for Facebook Prophet

Utilize the Yahoo Finance API, a widely used financial data source, to retrieve historical 'MSFT' (Microsoft stock) data. Specify the desired period from the year 2014 for analysis. After downloading the data, organize it by resetting the index and selecting only the 'Date' and 'Close' columns, streamlining the dataset for price prediction. dapt the column names to align with the input requirements of the Prophet model, specifying 'ds' for the date column and 'y' for the closing price column. This adjustment ensures compatibility with the model's expectations and facilitates smooth integration into the forecasting framework. Proceed to set up the Prophet model, a forecasting tool developed by Facebook. Configure the model parameters, including seasonality settings, holidays, and other relevant options. Proper initialization of the model is paramount for generating accurate and meaningful predictions.

Proceed to train the initialized Prophet model with the formatted historical stock data. During this stage, the model assimilates insights from past patterns and trends within the data, adjusting internal parameters to effectively capture the underlying structure of the time series data. Once trained, apply the Prophet model to produce forecasts for future time steps. This predictive capability allows the model to project how the stock price might evolve over a specified future period, leveraging insights gained during the model fitting phase. The dataset underwent division into two distinct parts: a training set inclusive of data spanning from 2014 to 2021, and a testing set encapsulating the entirety of the year 2022. This segregation enabled the training of models on historical information, thereby facilitating the evaluation of their predictive capabilities on unseen future data.

Each model was trained on the historical dataset, learning patterns, trends, and seasonality present in the data. The train-ing process involved adjusting model parameters to minimize the difference between predicted values and actual observa-tions in the training set. Following training, each model was employed on the testing set (2022 data) to produce forecasts for the future period. To assess the performance of the models, the predicted closing prices were subjected to Root Mean Square Error (RMSE) analysis, enabling the determination of the final minimized errors in the predicted prices. [15]

VI. Result and Conclusion

n our experimental setup, we utilized the LSTM, ARIMA, and Prophet models. Through our analysis, we observed that the LSTM model outperforms both the ARIMA and Prophet models in terms of predictive accuracy and performance, refer to figure 4 and figure 5. The code can be found at GitHub repository [1]

Based on its better performance measures, especially its minimal RMSE (Root Mean Squared) values refer to figure 7, the LSTM model is the best model for predicting MSFT (Microsoft stock) prices refer to figure 6. Remarkably, the LSTM model attains accuracy rates higher than 85 percent, demonstrating its resilient capacity to steer through the erratic swings in stock values, particularly during the difficult pandemic phase. Incorporating key financial variables into the model has the potential to enhance prediction accuracy in the future. These indicators, reflecting market and economic trends, could augment the model's predictive capabilities by capturing nuanced factors influencing fluctuations in stock and cryptocurrency prices.

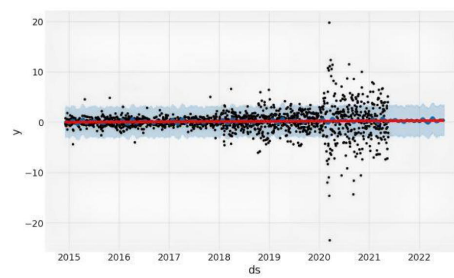

Fig. 4. Prophet Model Result

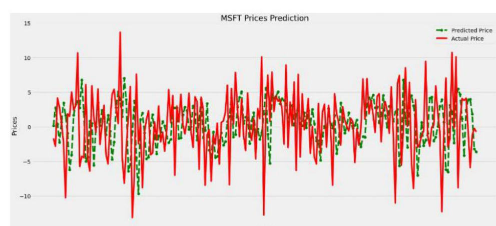

Fig. 4. ARIMA Model Result

The recommended strategies to further improve the accuracy and responsiveness of this predictive model to changing market conditions are the integration of sentiment analysis and financial indicators as we map out its future trajectory. In this study, predictions were solely based on price data. However, incorporating more influential features such as daily

1 Github Repositoryhttps://github.com/praxton74/FinalYearProject

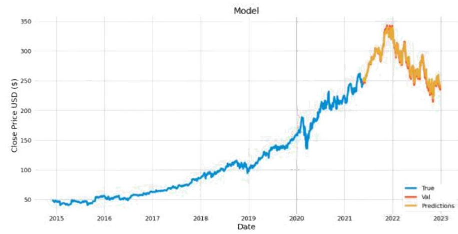

Fig. 6. LSTM Model Result

S.No	Evaluated Result	
	Model Name	**RMSE**
1	LSTM	1.204
2	ARIMA	6.615
3	Prophet	4.527

Fig. 7. Result Table

volume, volatility, fundamental ratios, etc., could enhance model performance. Future research could extend the model's applicability to predict less volatile market indices, such as Nasdaq. Additionally, exploring sentiments from news articles and integrating them into the LSTM model alongside other features could present an intriguing research avenue. Lastly, employing more sophisticated optimization techniques could prove beneficial when incorporating additional attributes into the model. [16]. Employing these four models was optimal as it provides a comprehensive examination of time series forecasting tools. Utilizing these models across three distinct asset classes allows us to observe their performance in varied environments. Moving forward, individuals can tailor their models based on the insights garnered from this report, selecting parameters that align best with their requirements. [17].

VII. Acknowledgement

We want to express our sincere gratitude to the Data Science & Deep Learning (DSDL) Lab, Computer Science & Engineering department, KIET Group of Institutions, Ghaziabad for their unwavering support and collaboration during this research. We also appreciate the access to the PARAM Shavak Supercomputing facility at DSDL Lab[2].

References

[1] I.-C. Lolea, I.-R. Petrariu, and A. Giurgiu, "Arima vs. machine learning in terms of equity market forecasting," The Annals of the University of Oradea. Economic Sciences, vol. 30, no. 2nd, 2021.

[2] S. Mehtab and J. Sen, "Stock price prediction using cnn and lstm-based deep learning models," in 2020 International Conference on Decision Aid Sciences and Application (DASA), pp. 447–453, IEEE, 2020.

[3] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," IEEE Access, vol. 8, pp. 150199–150212, 2020.

[4] J. Sen and S. Mehtab, "Accurate stock price forecasting using robust and optimized deep learning models," in 2021 International Conference on Intelligent Technologies (CONIT), pp. 1–9, IEEE, 2021.

[5] M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, and G. Wills, "A survey on machine learning for stock price prediction: algorithms and techniques," in 2nd International Conference on Finance, Economics, Management and IT Business (05/05/20 - 06/05/20), pp. 63–71, May 2020. Publisher Copyright: Copyright © 2020 by SCITEPRESS - Science and Technology Publications, Lda. All rights reserved.

[6] M. Nikou, G. Mansourfar, and J. Bagherzadeh, "Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms," Intelligent Systems in Accounting, Finance and Manage-ment, vol. 26, no. 4, pp. 164–174, 2019.

[7] B. G. MALATO, How many neurons for a neural network. YOUR DATA TEACHER Online courses and lessons about data science, machine learning and artificial intelligence, 2014.

[8] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," in 2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP), pp. 339–343, IEEE, 2018.

[9] S. Siami-Namini and A. S. Namin, "Forecasting economics and financial time series: Arima vs. lstm," arXiv preprint arXiv:1803.06386, 2018.

[10] M. Navratil and A. Kolkova, "Decomposition and forecasting time series in the business economy using prophet forecasting model," Central European Business Review, 2019.

[11] Z. Z. Oo and P. Sabai, "Time series prediction based on facebook prophet: A case study, temperature forecasting in myintkyina," Inter-national Journal of Applied Mathematics Electronics and Computers, 2020.

[12] A. Hasnain, Y. Sheng, M. Z. Hashmi, U. A. Bhatti, A. Hussain, M. Hameed, S. Marjan, S. U. Bazai, M. A. Hossain, M. Sahabuddin, et al., "Time series analysis and forecasting of air pollutants based on prophet forecasting model in jiangsu province, china," Frontiers in Environmental Science, vol. 10, p. 945628, 2022.

[13] S. Smyl and K. Kuber, "Data preprocessing and augmentation for multiple short time series forecasting with recurrent neural networks," arXiv preprint, 2016.

[14] A. H. Adineh, Z. Narimani, and S. C. Satapathy, "Importance of data preprocessing in time series prediction using sarima: A case study," International Journal of Knowledge-Based and Intelligent Engineering Systems, vol. 24, no. 4, pp. 331–342, 2020.

[15] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock closing price prediction using machine learning techniques," Procedia computer science, 2020.

[16] D. Shah, W. Campbell, and F. H. Zulkernine, "A comparative study of lstm and dnn for stock market forecasting," in 2018 IEEE international conference on big data (big data), pp. 4148–4155, IEEE, 2018.

[17] R. Verma, J. Sharma, and S. Jindal, "Time series forecasting using machine learning," in Advances in Computing and Data Sciences: 4th International Conference, ICACDS 2020, Valletta, Malta, April 24–25, 2020, Revised Selected Papers 4, pp. 372–381, Springer, 2020.

2 https://www.kiet.edu/home/department wise club detail/NQ==/NA==/student club

Plagiarism Report

Ishant Sehgal

ORIGINALITY REPORT

12%	8%	7%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Amir Hossein Adineh, Zahra Narimani, Suresh Chandra Satapathy. "Importance of data preprocessing in time series prediction using SARIMA: A case study", International Journal of Knowledge-based and Intelligent Engineering Systems, 2021 Publication	1%
2	ijarsct.co.in Internet Source	1%
3	arxiv.org Internet Source	1%
4	Submitted to Liverpool John Moores University Student Paper	1%
5	www.researchgate.net Internet Source	1%
6	www.medrxiv.org Internet Source	1%
7	Submitted to University of Bedfordshire Student Paper	<1%

8	Submitted to University of Nottingham Student Paper	<1 %
9	engrxiv.org Internet Source	<1 %
10	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %
11	www.mdpi.com Internet Source	<1 %
12	Submitted to Carnegie Mellon University Student Paper	<1 %
13	Submitted to Napier University Student Paper	<1 %
14	Sean McNally, Jason Roche, Simon Caton. "Predicting the Price of Bitcoin Using Machine Learning", 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018 Publication	<1 %
15	Submitted to University of Hull Student Paper	<1 %
16	Submitted to University of Westminster Student Paper	<1 %
17	5dok.net Internet Source	<1 %

18	Chih-Hung Wu, Chih-Chiang Lu, Yu-Feng Ma, Ruei-Shan Lu. "A New Forecasting Framework for Bitcoin Price with LSTM", 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 2018 Publication	<1%
19	www.kiet.edu Internet Source	<1%
20	Amir Mahmud Husein, Jefri Poltak Hutabarat, Jeckson Edition Sitorus, Tonazisokhi Giawa, Mawaddah Harahap. "Predicting the Spread of the Corona Virus (COVID-19) in Indonesia: Approach Visual Data Analysis and Prophet Forecasting", International Journal of Artificial Intelligence Research, 2020 Publication	<1%
21	Submitted to University of Wollongong Student Paper	<1%
22	econjournals.com Internet Source	<1%
23	journal.unm.ac.id Internet Source	<1%
24	www.publications.scrs.in Internet Source	<1%
25	H. AL-DUWAISH, M. NAZMUL KARIM, V. CHANDRASEKAR. "Hammerstein model	<1%

identification by multilayer feedforward neural networks", International Journal of Systems Science, 2007
Publication

26	Prenjal Tayal, Farman Ali, Shweta Pandey, Rajesh Singh, Jayapal Lande, Vikrant Pachouri. "AI Based Stock Market Prediction-Theoretical Perspective", 2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT), 2023 Publication	<1 %
27	Siddhartha Das Gupta, Teja Kolla, Ravi Yadav, Mamta Arora, Mrinal Pandey. "Forecasting Cryptocurrency Prices using Sequential and Time Series Models", Proceedings of the 4th International Conference on Information Management & Machine Intelligence, 2022 Publication	<1 %
28	Submitted to University of Adelaide Student Paper	<1 %
29	dergipark.org.tr Internet Source	<1 %
30	ebin.pub Internet Source	<1 %
31	etasr.com Internet Source	<1 %

32	export.arxiv.org Internet Source	<1 %
33	medium.com Internet Source	<1 %
34	pubs2.ascee.org Internet Source	<1 %
35	www.jonesmutual.com Internet Source	<1 %
36	Alya'a Bajoudah, Mohammad Alsaidi, Ahmad Alhindi. "Time Series Forecasting Model for E-commerce Store Sales Using FB-Prophet", 2023 14th International Conference on Information and Communication Systems (ICICS), 2023 Publication	<1 %
37	Donghwan Song, Adrian Matias Chung Baek, Namhun Kim. "Forecasting Stock Market Indices Using Padding-Based Fourier Transform Denoising and Time Series Deep Learning Models", IEEE Access, 2021 Publication	<1 %
38	Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar. "Stock Closing Price Prediction using Machine Learning Techniques", Procedia Computer Science, 2020 Publication	<1 %

39	Petr Suler, Zuzana Rowland, Tomas Krulicky. "Evaluation of the Accuracy of Machine Learning Predictions of the Czech Republic’s Exports to the China", Journal of Risk and Financial Management, 2021 Publication	<1 %
40	Salim Salim, Jyoti Ohri. "Monitoring and Analysis of Solar PV based on GUI", 2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), 2020 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

CERTIFICATE of Paper Presentation

AMITY UNIVERSITY | IEEE UP SECTION (INDIA)

INTERNATIONAL CONFERENCE ON INNOVATIVE PRACTICES IN TECHNOLOGY AND MANAGEMENT (ICIPTM 2024)

21st - 23rd February, 2024

CERTIFICATE OF PARTICIPATION IN RESEARCH PAPER PRESENTATION

This is to acknowledge the participation of Parth Mishra (Acodemicion/Corporate/Research Scholar/Student) in the Research Paper Presentation held during **International Conference on Innovative Practices in Technology and Management (ICIPTM 2024)** on 21st - 23rd February 2024, held at Amity University Noida.

The paper titled Revolutionizing Stock Market Intelligence: A Deep Dive into Machine Learning for IT Sector Price Predictions was worth appreciating for its efforts in generating, analyzing and presenting research content.

Prof. (Dr.) Gurinder Singh
Conference Chairman
Group Vice Chancellor, Amity Universities

Dr. Atul Chauhan
President, Ritnand Balved Education Foundation
& Chancellor, Amity University