*A project report on*

# EYE DETECTION AND TRACKING USING DEEP LEARNING

*Submitted in partial fulfilment for the award of the degree of*

## Bachelor of Technology in Electronics and Communication Engineering

*by*

**MARIA AUSTIN(21BEC1225)**



**SCHOOL OF ELECTRONICS AND COMMUNICATIONS ENGINEERING**

NOVEMBER, 2024

# EYE DETECTION AND TRACKING USING DEEP LEARNING

*Submitted in partial fulfillment for the award of the degree of*

## Bachelor of Technology in Electronics and Communication Engineering

*by*

**MARIA AUSTIN(21BEC1225)**



**SCHOOL OF ELECTRONICS AND COMMUNICATIONS ENGINEERING**

NOVEMBER, 2024



**DECLARATION**

I hereby declare that the thesis entitled “EYE DETECTION AND TRACKING USING DEEP LEARNING” submitted by me, for the award of the degree of Bachelor of Technology in Electronics and Communications Engineering ,Vellore Institute of Technology, Chennai is a record of bonafide work carried out by me under the supervision of Dr.BRINTHA THERESE.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: Signature of the Candidate



**School of Electronics and Communications Engineering**

CERTIFICATE

This is to certify that the report entitled **“**EYE DETECTION AND TRACKING USING DEEP LEARNING**”** is prepared and submitted by **Maria AUSTIN(21BEC1225**) to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Electronics and Communication Engineering** programme is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr./Prof.

Date:

Signature of the Internal Examiner Signature of the External Examiner

Name: Name:

Date: Date:

Approved by the Head of Department,

**B.Tech. ELECTRONICS AND COMMUNICATION ENGINEERING**

Name:

Date:

(Seal of SCOPE)

**ABSTRACT**

This project report details the inception and application of a novel ensemble model that integrates the advanced computational prowess of Stacked Long Short-Term Memory (LSTM) networks with the predictive precision of Nonlinear AutoRegressive with eXogenous inputs (NARX). This integrated model stands as a pioneering approach to stock price forecasting, specifically engineered to tackle the multifaceted and unpredictable nature of financial time series data. Traditional forecasting methodologies often struggle with the erratic behaviors of financial markets, but this ensemble model is strategically designed to decode the complexities of temporal sequences and the influence of external market variables, resulting in forecasts of exceptional accuracy.

The intricacies of the model are explored in depth, illustrating the fusion of Stacked LSTM's deep learning efficiency in identifying hidden patterns over time with NARX's sensitivity to external factors and nonlinear relationships. This combination is meticulously calibrated to adapt to the financial market's volatility, with extensive validation performed against historical data to ensure both accuracy and robustness in its predictive capabilities. The empirical analysis presented in the report confirms the model's superiority over traditional techniques, particularly its resilience in adapting to unexpected financial disturbances, positioning it as a formidable tool for understanding and anticipating market trends.

Moreover, the ensemble model's potential extends far beyond mere prediction. It heralds a transformative potential for risk management, portfolio management, and strategic investment decision-making, offering market participants a substantial edge. The report also considers the broader implications for economic research, where such advanced tools can provide deeper insights into market dynamics. Future avenues of research are outlined, focusing on the real-time application of the model to enhance its responsiveness and exploring further advancements in deep learning to solidify its standing at the forefront of financial forecasting technology. This project serves as a cornerstone for the next generation of financial analysis, promising to evolve the landscape of economic forecasting through innovative and data-driven methodologies.

**ACKNOWLEDGEMENT**

It is my pleasure to express with deep sense of gratitude to Dr. Ahadit A.B.,

Assistant Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and expert in the field of Artificial Intelligence & Machine Learning.

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan our Honorable Chancellor, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. G V Selvam Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant Vice-President, Dr. V. S. Kanchana Bhaaskaran Vice-Chancellor i/c & Pro-Vice Chancellor, VIT Chennai and Dr. P. K. Manoharan, Additional Registrar for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In a jubilant state, I express ingeniously my whole-hearted thanks to Dr. Harini S, Head of the Department, B.Tech. CSE with Specialization in Artificial Intelligence & Robotics and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculty members and staff members at Vellore Institute of Technology, Chennai who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: **Rishikesh Raj Nair**

## CONTENTS

[**CONTENTS**](#_heading=h.30j0zll) 5

[**LIST OF FIGURES**](#_heading=h.1fob9te) 9

**LIST OF TABLES** 11

**LIST OF ACRONYMS** 12

**CHAPTER 1 INTRODUCTION**

* 1. INTRODUCTION 13
  2. OVERVIEW OF ML/DL MODELS IN STOCK MARKET 14
  3. CHALLENGES PRESENT IN CURRENT MODELS 15
  4. PROBLEM STATEMENT 16
  5. OBJECTIVES 17
  6. SCOPE OF THE PROJECT 18

**CHAPTER 2 BACK GROUND**

* 1. INTRODUCTION 20
  2. LITERATURE SURVEY ON RELATED WORKS 21

**CHAPTER 3**

**METHODOLOGY**

* 1. BRIEF OVERVIEW 23
  2. NARX MODEL 24

3.3 LSTM MODEL 26

3.4 TESTING THE IMPACT OF TECHNICAL FACTORS 28

3.5 ENSEMBLE MODEL OF STACKED LSTM & NARX 30

3.6 EXPLANATION OF MODULES FOR THE ENSEMBLE MODEL 34

3.7 ALGORITHM FOR THE ENSEMBLE MODEL 36

**CHAPTER 4**

**METHODOLOGY**

* 1. LSTM MODEL 37

4.2 NARX MODEL 43

4.3 LSTM MODEL WITH ADDITIONAL PARAMETERS 47

4.4 STACKED LSTM – NARX ENSMBLE MODEL 51

**CHAPTER 5**

**RESULTS**

* 1. BRIEF INTRODUCTION TO RESULTS 57

5.2 GRAPHICAL REPRESENTATION & ANALYSIS OF RESULT 58

5.3 TABULAR REPRESENTATION & ANALYSIS OF RESULTS 75

5.4 PREDICTION INTO THE FUTURE USING ENSEMBLE MODEL (30 DAYS) 78

**CHAPTER 6**

**CONCLUSION & FUTURE WORK**…………………………………………………81

**APPENDICES**………………………………………………………………………….82

**REFERENCES**…………………………………………………………………………85

### LIST OF FIGURES

* 1. ARCHITECTURE OF NARX MODEL 25
  2. ARCHITECTURE OF LSTM MODEL 27
  3. LAYER BY LAYER ARCHITECTURE DIAGRAM OF STACKED LSTM 31

& NARX ENSMEBLE MODEL

* 1. OVERALL ARCHITECTURE DIAGRAM OF STACKED LSTM 32

& NARX ENSMEBLE MODEL

* 1. FLOW DIAGRAM OF PROPOSED SYSTEM ( STACKED LSTM 33

& NARX ENSMEBLE MODEL )

* 1. GOOGLE DATASET – LSTM RESULTS 59
  2. GOOGLE DATASET – NARX RESULTS 60
  3. GOOGLE DATASET – LSTM WITH ADDITIONAL PARAMETERS 61

RESULTS

* 1. GOOGLE DATASET – ENSEMBLE MODEL OF STACKED LSTM & NARX 62

RESULTS

* 1. NIKE DATASET – LSTM RESULTS 63
  2. NIKE DATASET – NARX RESULTS 64
  3. NIKE DATASET – LSTM WITH ADDITIONAL PARAMETERS 65

RESULTS

* 1. NIKE DATASET – ENSEMBLE MODEL OF STACKED LSTM & NARX 66

RESULTS

* 1. ITC DATASET – LSTM RESULTS 67
  2. ITC DATASET – NARX RESULTS 68
  3. ITC DATASET – LSTM WITH ADDITIONAL PARAMETERS 69

RESULTS

* 1. ITC DATASET – ENSEMBLE MODEL OF STACKED LSTM & NARX 70

RESULTS

* 1. TATA STEEL DATASET – LSTM RESULTS 71
  2. TATA STEEL DATASET – NARX RESULTS 72
  3. TATA STEEL DATASET – LSTM WITH ADDITIONAL PARAMETERS 73

RESULTS

* 1. TATA STEEL DATASET – ENSEMBLE MODEL OF STACKED LSTM 74

& NARX RESULTS

* 1. 30 DAYS FORECAST USING ENSEMBLE MODEL (GOOGLE DATASET) 78
  2. 30 DAYS FORECAST USING ENSEMBLE MODEL (NIKE DATASET) 79
  3. 30 DAYS FORECAST USING ENSEMBLE MODEL (ITC DATASET) 79
  4. 30 DAYS FORECAST USING ENSEMBLE MODEL 80

(TATA STEEL DATASET)

**LIST OF TABLES**

1.COMPARISION OF ACCURACY % OF MODELS WITH VARIOUS STOCKS 22

2.COMPARISION OF MSE OF MODELS WITH VARIOUS STOCKS 48

**LIST OF ACRONYMS**

ML Machine Learning

DL Deep Learning

MSE Mean Square Error

LSTM Long Short-Term Memory

NARX Non-linear Autoregressive Network with Exogenous Inputs

VWAP Volume-Weighted Average Price

EMA Exponential Moving Average

**Chapter 1**

**Introduction**

1.1 INTRODUCTION

The introduction to the project on "Time Series Forecasting of Stock Prices Using Advanced Deep Learning Techniques" embarks on a critical examination of the current landscape of financial forecasting, where the intricate dance of stock prices is influenced by a multitude of unpredictable factors. Traditional machine learning methodologies, once the vanguard of predictive analytics, are increasingly showing their limitations in this complex arena. They typically stumble when faced with the task of untangling the intertwined temporal dependencies and deciphering the non-linear patterns that are the hallmarks of financial time series data. Moreover, these models often lack the necessary flexibility to adapt to sudden economic shifts or geopolitical incidents, resulting in predictions that fail to stand up to the multifaceted nature of the markets.

In light of these shortcomings, this project is rooted in the premise that deep learning models, with their profound ability to process layers of abstraction, may hold the key to a new era of forecasting accuracy. Deep learning models, particularly those employing recurrent neural architectures like LSTM (Long Short-Term Memory), are inherently designed to recognize and model temporal information, which is a critical component in the analysis of financial time series. The research explores the augmentation of these models, expanding them beyond their conventional capabilities to embrace the chaotic and dynamic environment of stock trading.

The innovation at the heart of this research is the application of ensemble techniques, which amalgamate the predictive powers of various deep learning models. This approach is hypothesized to outshine the performance of regular machine learning techniques significantly. By pooling the strengths and mitigating the weaknesses of individual models, ensemble strategies aim to enhance prediction confidence and reliability, particularly under conditions of market uncertainty and when dealing with events that defy historical patterns. This project report delves into the specifics of implementing these advanced ensemble techniques, evaluating their performance against the backdrop of real-world financial data, and exploring their potential to redefine the standards of financial forecasting. The subsequent sections will detail the methodologies employed, the experiments conducted, and the insights gained from this comprehensive investigation into the realm of advanced deep learning for stock price prediction.

1.2 OVERVIEW OF ML/DL MODELS IN STOCK PRICE PREDICITON

Sophisticated techniques in stock price prediction, with the intersection of finance and technology, particularly machine learning (ML) and deep learning (DL), have been the result. In finance, these are applied to uncover the complex, non-linear patterns that traditional methods of analysis often fail to pick out.

Recurrent Neural Networks (RNNs) have emerged as a critical tool for modelling the sequential data that characterizes time series data within stock prices. RNNs, by their nature, have a structure much like the temporal nature of the financial markets, and it captures information from previous time steps to make a prediction at current time. However, their effectiveness is limited by problems such as the vanishing and exploding gradient, where the performance of weights and biases degenerates, hampering the network to learn from data points that are far apart in time. This, therefore, makes them less reliable for long-run predictions, which are equally important for financial forecasting. Long Short-Term Memory networks (LSTMs) solve this problem by using a gating mechanism to enable the flow of information. They allow capturing the important historical information while enabling the network to discard all that is unimportant—typical hallmark for catching long-term dependencies needed to be captured in great predictions. Yet, despite this capability, LSTMs do have some problems. They are complex and take huge computational resources, so, therefore, limiting the possibilities for their use. Furthermore, the extensive training time and risk of overfitting to past data make them a less than perfect solution.

Gated Recurrent Units (GRUs) are a streamlined version of LSTMs. The simple, compact architecture of GRUs decreases the computational burden because there are fewer parameters to train. Such simplicity, however, may come at the cost of model expressiveness, which would otherwise be vital for the model to be able to fully learn and represent the complexity of the data. Some frameworks of Convolutional Neural Networks (CNNs), which have been mainly designed for the purpose of image recognition, are extended to adapt them for the time series data. In the case of the stock price data, CNNs are applicable to the detection of patterns and trends due to the application of convolutional filters and prove valuable to point out important features which might be influential in affecting future prices. However, the major disadvantage of CNNs in stock prediction is the very design for which they are optimized: spatial data. This makes it less intuitive for time-series forecasting, as temporal dependencies are not captured as good as RNN-based models.

It is no surprise, then, that deep learning models, especially those devoted to sequence processing, yielded great promise for capturing the intricacies of stock market data. The interpretability is very low when compared to standard machine learning models, and they usually perform poorly due to lack of interpretability compared to deep learning models. It is of paramount importance that the reasons behind the forecasts are exposed in financial decision-making. Furthermore, the opaqueness of deep learning models may reduce ability by analysts and investors to trust and make use of the forecasts. Thirdly, the dynamic, often volatile nature of the financial market adds a further layer of complexity to the use of ML/DL models. Financial time-series data are non-stationary, where the patterns and trends of the past might not be able to predict the future. This non-stationarity is to say that models have to be continuously updated and retrained if they are to remain of value, which makes this a resource-intensive issue. In most cases, it is resource-intensive; that is why it finds an application in stock price prediction. Surely an advantage of these models in comparison with the traditional ones is their capability to process big data volumes and find delicate patterns. With the improvement in computation techniques and all sorts of more refined models being devised, there is very clear headroom for enhanced accuracy in forecasting. However, the way that lies forward should be thoughtfully considered with respect to the weaknesses of every model so that the advantages of these sophisticated ways wouldn't overshadow them.

1.3 CHALLENGES PRESENT IN CURRENT MODELS

The integration of Machine Learning (ML) and Deep Learning (DL) into stock market prediction has been a game-changer for financial analysts and investors. However, this integration is not without its challenges. These sophisticated algorithms must navigate a labyrinth of obstacles to accurately forecast market movements. This paper elucidates the primary challenges that ML and DL models face in the domain of stock market price prediction.

1. Data Integrity and Volume

Stock market prediction is contingent on high-quality data. However, financial datasets are notoriously noisy, incomplete, and often contain outliers that can lead to inaccurate model predictions. The sheer volume of data can also pose a challenge, as models need to process and learn from terabytes of information, which includes not just numerical stock prices but also textual news articles and economic reports.

2. Market Volatility and Dynamics

The financial market is inherently volatile, with prices influenced by a plethora of unpredictable factors such as political events, economic changes, and company performance. ML and DL models can struggle to adapt to these sudden shifts, as the models are typically trained on historical data that may not account for unprecedented future events.

3. Non-Stationarity of Financial Time Series

Financial time series data is non-stationary, meaning that its statistical properties change over time. This non-stationarity is a significant hurdle for ML/DL models, which assume that patterns in the data persist over time. When these patterns change, the models' predictions become less reliable.

4. Overfitting and Generalization

ML and DL models are prone to overfitting, where they perform well on training data but fail to generalize to unseen data. The complex nature of the stock market, with its intricate patterns and noise, exacerbates this issue, leading to models that capture noise as if it were a signal.

5. Model Interpretability and Trust

The "black box" nature of many ML and DL models, particularly deep neural networks, poses a challenge for analysts and investors who must understand and trust the predictions made by these models. Without a clear understanding of how predictions are derived, it's difficult to have confidence in the decisions based on these models.

6. Computational Complexity and Resource Requirements

Training ML and DL models require significant computational resources. The complexity of these models, especially deep learning architectures like recurrent neural networks (RNNs), convolutional neural networks (CNNs), and their variants, necessitates powerful hardware and can lead to long training times.

7. Real-time Data Processing

The stock market is a real-time environment where prices change by the second. ML and DL models must process data in real-time to make timely predictions, which can be a challenge given the computational demands of these models.

8. Regulatory and Ethical Considerations

The use of ML and DL in stock market prediction raises regulatory and ethical considerations. There are concerns about the potential for models to be used for market manipulation or insider trading. Regulatory bodies are still catching up with the technology, and there's a lack of clear guidelines on the ethical use of AI in financial markets.

So, the application of ML and DL models to stock market prediction offers significant potential benefits, but it is clear that there are substantial challenges that need to be addressed. Overcoming these hurdles requires not only technological advancements but also a better understanding of the financial markets' nature and behaviour. As the field progresses, it is crucial for developers and users of these models to remain cognizant of both their capabilities and their limitations. Only through a concerted effort to address these challenges can we harness the full potential of ML and DL in the realm of financial forecasting.

1.4 PROBLEM STATEMENT

In the high-stakes world of financial trading, the accurate prediction of stock prices constitutes a cornerstone of strategic investment and risk management. However, this task is fraught with complexity due to the inherently unpredictable nature of financial markets. Traditional forecasting models, while having provided a foundation for stock price analysis, are increasingly inadequate when confronted with the intricacies of modern financial time series data.

The central challenge lies in the inability of these conventional methods to unravel the intricate temporal dependencies and the nonlinear patterns that characterize financial markets. These models are limited by their linear assumptions and are often unable to adapt to the dynamic, ever-evolving landscape of global finance. Market volatility, the influence of economic indicators, and the impact of unforeseen events, such as sudden geopolitical shifts or unexpected global economic developments, further exacerbate the difficulty of making reliable predictions.

Moreover, the predictability of stock prices is undermined by the frequency of 'black swan' events that traditional forecasting tools cannot anticipate. These events can cause significant deviations in stock prices, leading to substantial financial consequences. As a result, there is a pressing need for predictive models that are not only sophisticated in their analytical capabilities but also possess a high degree of adaptability to the turbulent nature of financial markets. This project is poised to tackle these formidable challenges head-on by delving into the realm of advanced deep learning techniques. Our objective is to refine and enhance the capabilities of existing machine learning models, equipping them to capture the volatile essence of financial time series more effectively. This entails developing models that can intelligently interpret complex temporal sequences and adapt in real-time to nonlinear market trends, including those trends influenced by unpredictable global events. The problem at hand is multifaceted: it requires an improvement in the models' ability to decode complex temporal dynamics and to generalize from historical data to future states in a manner that is robust to the market's unpredictability. The research will focus on the integration and innovation of cutting-edge deep learning techniques.

The main aim to push the boundaries of what is currently achievable in stock price forecasting. By leveraging the full spectrum of deep learning methodologies, we seek to construct a framework that is not just reactive to market conditions, but one that offers a predictive insight with a higher degree of accuracy and reliability than has ever been possible. The end goal of this project is to pioneer solutions that significantly elevate the precision, reliability, and robustness of stock price forecasting models, thus providing investors and analysts with tools that can better navigate the complexities and volatilities inherent in the financial markets. In doing so, we confront questions of computational feasibility, model interpretability, and real-world applicability. The research will have to consider the ethical implications of AI in financial decision-making, ensuring that the advanced techniques developed do not only serve a functional purpose but do so within the boundaries of regulatory compliance and ethical investment practices. Ultimately, this project aims to bridge the gap between the theoretical potential of advanced deep learning models and their practical implementation in stock market forecasting, providing a beacon of insight in the often, opaque waters of financial prediction.

1.5 OBJECTIVES

In the intricate landscape of financial forecasting, the application of advanced deep learning techniques is poised to redefine the efficacy of stock price prediction models. This research initiative is directed towards the development of robust deep learning models. These models are anticipated to significantly enhance the understanding and interpretation of complex temporal dependencies inherent in financial time series data. The improvement of these models is expected to yield predictions that are not only accurate but also resilient in the face of the non-linear patterns characteristic of stock price movements.

Addressing the pervasive uncertainty within financial markets is a critical aspect of this research. The initiative involves designing adaptive models that are capable of providing reliable predictions amidst market volatility. The incorporation of real-time data and the implementation of adaptive learning mechanisms are set to improve the models' responsiveness to fluctuating market conditions. The goal is to explore the forefront of deep learning to enable these models to adjust swiftly and accurately to the dynamic nature of financial markets.

The unpredictable influence of unforeseen events on stock prices is another challenge that this research aims to mitigate. Methodologies are being developed to account for sudden market shifts triggered by unexpected global economic changes or geopolitical events. The intention is to employ advanced deep learning techniques adept at recognizing and reacting to these abrupt changes, thereby enhancing the models' robustness and reliability.

Enhancing the accuracy of predictive models remains a pivotal goal of this research. Investigating and incorporating diverse features and novel data sources to refine stock price forecasts is a key objective. A thorough evaluation of model performance, in comparison with traditional forecasting methods, is essential to quantify and elevate the accuracy of predictions.

A deeper comprehension of the complexities that govern financial markets is integral to the success of predictive models. This initiative seeks to refine machine learning models to closely mirror the complexities of financial data and market dynamics. It involves a detailed investigation into the various factors that influence stock price fluctuations, aiming to build a more complete understanding of the financial markets.

Contribution to the broader domain of stock price forecasting is a fundamental ambition of this research. It involves engaging with the research community to share innovative methodologies and insights. The aspiration is to aid in creating a more dependable and efficient suite of tools for the forecasting of stock prices, effectively bridging traditional analytical methods with cutting-edge deep learning techniques.

This research is committed to advancing the predictability and comprehensibility of stock prices in the financial sector. By pushing the boundaries of current technological capabilities and aligning them with market dynamics, the aim is to forge predictive tools that are not only advanced in their analytical prowess but also practical and responsible in their deployment within the financial industry.

1.6 SCOPE OF THE PROJECT

The scope of this research project is to architect and refine advanced deep learning models for the purpose of forecasting stock prices using historical datasets. By focusing on historical data, the project intends to extract and learn from the wealth of information that past market performance provides, eschewing real-time data such as news feeds, social media sentiment, and economic reports which can introduce volatility and noise into predictive models.

At the core of this endeavour is the development of a novel deep learning model that can effectively interpret the intricate patterns and temporal sequences inherent in historical financial data. The utilization of existing neural network architectures, such as Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and the exploration of emergent techniques in deep learning, will serve as a baseline. The project will innovate beyond these traditional frameworks, aiming to solve complex problems such as non-linear dependencies, market volatility, and the chaotic nature of financial time series. The intention is to ensure that the models developed are not only robust and accurate but also universally applicable. This universality is crucial as the project aspires to create models flexible enough to be applied across various stock exchanges and adaptable to different currencies around the world. Such a model would have to account for the idiosyncrasies of different markets, including variations in trading volumes, market size, and the economic contexts that uniquely affect each exchange.

To achieve this level of flexibility and adaptability, the project will investigate a variety of advanced machine learning techniques. This includes, but is not limited to, the application of transfer learning, where a model trained on one stock exchange can be adapted to predict stock prices on another, with minimal need for retraining. Additionally, the project will explore the use of meta-learning algorithms that can quickly adapt to new data from different financial environments, effectively learning to learn from the global financial ecosystem.

A key part of this project's scope is to provide a comparative analysis of the predictive capabilities of these newly developed models against traditional time series forecasting methods. This will involve back-testing predictions on historical data to ensure the accuracy and reliability of the models across different time periods and market conditions.Moreover, the project is dedicated to enhancing the understanding of deep learning's role in financial forecasting. By focusing on historical data, the project seeks to uncover the underlying patterns and trends that drive stock market prices over time. Through this deep dive into historical analysis, the research will contribute to a more profound understanding of how stock markets behave and evolve, providing invaluable insights into the nature of financial time series data.

The developed models will be rigorously tested for their performance and scalability, with the aim of deploying them as a reliable tool for investors and financial analysts worldwide. By accomplishing these objectives, the project will advance the field of financial forecasting, providing a new benchmark for the use of deep learning in the analysis of stock prices.In essence, the project's scope is ambitious in its technical goals and broad in its geographic and economic applicability. It aspires to break new ground in the predictive accuracy of stock prices, offering a significant contribution to the field of financial analysis and the practices of global investment strategies.

**Chapter 2**

**Back Ground**

2.1 INTRODUCTION

In the financial world of ever-changing trends, the quest for highly precise stock price predictions is tantamount to finding a needle in the haystack. Unpredictable markets are a problem, being influenced by a manifold of things such as economic indicators, market volatility, and events which catch the most seasoned analysts by surprise. However, the traditional forecasting tools using the linear approach often get defeated by navigating through the complex, non-linear pathways of financial time series data. This inadequacy has set the stage for further investigation into more innovative, changeable methods that would be able to be adequate with the fluctuation of the financial environment.

Enter into the realm of next-level deep learning methods, where the ensemble model of Stacked Long Short-Term Memory (LSTM) and Nonlinear Autoregressive Exogenous (NARX) models is all set to change the way stock prices are foretold. It is not an academic work: it is some kind of mission to master the formidable power of modern data analysis and machine learning in order to understand the complicated dance of numbers and trends defining the stock market. The project refines, extends, and advances these models so that the level of prediction accuracy can be achieved which was hitherto considered unattainable. The beauty of the Stacked LSTM model is in its ability to remember and learn from an enormous amount of historical data. This enables it to seize a temporal dependency of great significance in the financial world. A combined ensemble model with the NARX model should, therefore, be a potent team player in the arsenal of financial forecasting, charged with capturing nonlinear relationships and external influences.This applied research project does not push the technical frontier in any sense but tries to act as a bridge from the very complicated world of deep learning to practical needs in financial markets. It would be of paramount importance to fine-tune these advanced models in such a manner that they can understand even the finest details in the financial data, so as to make the market trends and events that occur without warning more adaptive and insightful. That would be a major development in itself. In the case where the financial markets are getting more interlinked and volatile, then there is an invaluable ability for predicting stock prices with more accurate outcomes and consistency. This research represents an invaluable advance in the search for more dependable predictive algorithms. With the integration of Stacked LSTM and NARX methodologies, the project avails a path for not only predicting but also accurately sailing the ebb and flow of stock prices amidst the complexities of global financial markets.

With this journey, the goal becomes very clear from the start: it is the elevation of the science of stock price forecasting to a new level, giving traders, analysts, and investors a new tool that can make a difference for them in the way they interpret market dynamics and react to it. In doing so, the promise of this study is not only underlined in the purview of the technology itself but in its potential to demystify the unpredictable nature of the stock market and make it accessible and usable to everyone through understandability.

2.2 LITERTAURE SURVEY ON RELATED WORKS

In the domain of stock market prediction, deep learning has garnered substantial interest for its capability to tackle complex temporal patterns and non-linearities. This section highlights contemporary research efforts that have leveraged advanced deep learning techniques for time series forecasting of stock prices, contributing to the growing body of knowledge in financial market prediction.

Moghar and Hamiche [1] initiated the exploration into LSTM networks, which have shown promise in grasping the temporal dependencies characteristic of volatile financial markets.

Their work underscores the advantages of LSTMs over simpler models, particularly in handling the irregular patterns often observed in stock data, and examines the influence of training epochs on model precision.Building on this, Dinesh et al. [2] conducted an extensive review of various predictive models, concluding the superiority of LSTM networks for time series analysis. Their model stands out with an impressive accuracy rate, cementing LSTM's status as a formidable tool for forecasting stock trends. Zhang [3] further affirmed the efficacy of LSTM networks, especially for extended prediction intervals. This paper sheds light on the nuances of parameter optimization within LSTM networks to bolster prediction reliability, although its performance was noted to be less stellar over shorter forecast periods.

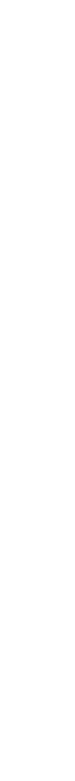
In a different approach, Dhafer AH et al. [4] employed the NARX neural network, integrating exogenous variables to enhance stock price predictions. Their methodical experimentation with network parameters and inputs exemplifies the meticulous calibration needed for neural network optimization. Alkhoshi and Belkasim [5] presented an adaptive NARX model for predicting stock market indices, utilizing time series analysis and the Levenberg-Marquardt algorithm to refine predictions, which are validated by the model’s mean squared error metrics.The work of Shen and Shafiq [6] represents a leap forward through the application of a deep learning system tailored with comprehensive feature engineering, which demonstrates superior performance in forecasting stock market trends.

In the seventh study, Yetis et al. [7] successfully employed ANNs to predict NASDAQ stock values, highlighting the network’s training with historical stock market data and its promising prediction outcomes.A novel GAN architecture was introduced by Kang Zhang et al. [8], merging LSTM and MLP to simulate stock data distribution, which underscores the expanding horizon of deep learning applications in financial forecasting. Pang et al. [9] innovated with a neural network model that integrates an embedded layer into an LSTM framework, achieving notable accuracy in predicting stock market indices and individual stock performance.

Maiti and Shetty D [10] investigated the predictability of Indian stock market prices using LSTM and GAN models, with a focus on the impact of data segmentation on forecast accuracy.Pakdaman Naeini et al. [11] compared the performance of MLP and Elman recurrent networks, with findings suggesting the MLP's superior capability in predicting stock price movements. Sonkiya's [12] exploration of BERT and GAN for stock price prediction exemplifies the fusion of sentiment analysis and deep learning, providing a comprehensive approach to forecasting.

Kalyoncu et al. [13] demonstrated the utility of LSTM models in predicting the stock values of Turkish companies, with success metrics including accuracy and error measures. Vijh et al. [14] applied Artificial Neural Networks and Random Forest techniques to predict stock closing prices, using financial data to create input variables, resulting in efficient prediction models as indicated by low RMSE and MAPE values. Lastly, Lanbouri and Said [15] explored the applicability of LSTM in High Frequency Trading environments, using technical indicators to predict short-term price movements, and validating the model's effectiveness through performance metrics.

Collectively, this survey underscores the transformative impact of advanced deep learning techniques in financial time series forecasting, marking significant strides in the prediction accuracy and reliability of stock market trends.



**Chapter 3**

**Methodology**

3.1 BRIEF OVERVIEW

In the complex arena of financial markets, accurately predicting stock prices represents a formidable challenge, underscored by the need for a deep understanding of the myriad factors that influence market dynamics. This research endeavors to construct a predictive model that not only captures the intricate temporal dependencies present in stock market data but also seeks to enhance prediction accuracy through innovative experimentation. Central to this exploration is the novel ensemble model that merges the strengths of Nonlinear Autoregressive Exogenous (NARX) and Stacked Long Short-Term Memory (LSTM) models. This fusion is designed to outperform the predictive capabilities of traditional models by leveraging LSTM's profound memory function and NARX's adeptness at modeling complex nonlinear interactions influenced by external factors.

The methodology embarks on a comparative analysis, positioning this unique ensemble model against the standard LSTM and NARX models. This comparison is not a mere academic pursuit but a rigorous evaluation aimed at understanding how these models fare in the tangible realm of stock market forecasting. A diverse portfolio of companies, including the tech behemoth Google, sportswear titan Nike, Indian conglomerate ITC, and the steel manufacturing giant Tata Steel, has been selected for this study. Spanning various industries and geographies, this selection provides a broad testing ground for the models. The data is split in an 80:20 ratio for training and testing, ensuring a solid framework for assessment.

This research also delves into the impact of technical indicators or factors such as the Volume Weighted Average Price (VWAP), 50 Day Exponential Moving Average (EMA), and 200 Day EMA. These indicators are pivotal in helping traders and analysts discern market sentiment and forecast future price movements. The study investigates the influence of these indicators on stock price predictions when applying standard, non-ensemble methods, analyzing stocks from different exchanges and currencies to impart a global perspective on stock market prediction. A distinctive aspect of this approach is its forward-looking capability, challenging the ensemble model to predict stock prices 30 days into the future. This ambitious task tests the limits of predictive accuracy and reliability, serving as critical evidence of the model's effectiveness in various scenarios.

This research extends beyond merely adding another model to the existing plethora of predictive tools; it signifies a significant advancement in the pursuit of accuracy and reliability in stock market predictions. By integrating the advantages of NARX and Stacked LSTM into a unified ensemble model and examining the role of essential technical indicators, the study aims to shed new light on financial forecasting methods. Through comprehensive testing and comparative analysis, the study endeavors to affirm the superior predictive prowess of the ensemble model, offering new insights for investors, analysts, and researchers navigating the complexities of global financial markets.

3.2 NARX MODEL

In the quest to forecast stock prices with precision, the Nonlinear AutoRegressive with eXogenous inputs (NARX) model emerges as a foundational tool. This initial foray into the complex domain of time series forecasting aimed to harness the NARX model's capabilities to untangle the intricate patterns inherent in financial markets. The approach was methodical, starting with the crucial step of determining the optimal number of lagged time steps for the model. With the decision to set this number at 10, the model was primed to incorporate a depth of historical data, ensuring that the inputs were rich with the relevant past information necessary for accurate forecasting.

This process of constructing the model began with the meticulous crafting of lagged sequences derived from the normalized closing prices of selected stocks. Such normalization ensured that the model's inputs were not only relevant but also standardized, allowing for a consistent basis of comparison across different time periods and stocks. Following this, the dataset underwent a strategic split into training and testing sets, adhering to an 80-20 partition. This careful division was pivotal, allowing for the thorough training of the model on a substantial dataset while reserving a portion of unseen data to test and validate the model's predictive prowess.

The architecture of the NARX model was deliberately designed to be straightforward yet effective. It featured a simple feedforward network architecture, beginning with an input layer tailored to accommodate the sequence of lagged steps. This was followed by a hidden layer comprised of 10 neurons, employing the 'tanh' activation function to capture and model the non-linear relationships that are characteristic of financial time series data. The architecture culminated in a single neuron with a linear activation function, dedicated to outputting the forecasted stock price value.

The training of the NARX model employed the Adam optimizer, renowned for its efficient computation and aptitude for handling large datasets, making it an ideal choice for this application. The mean squared error (MSE) served as the loss function, providing a quantitative measure of the model's forecasting accuracy. To fortify the model against the risk of overfitting—a common pitfall in machine learning—a strategy of early stopping was implemented. This approach involved monitoring the validation loss, with the training process being halted if no improvement was detected over a span of five consecutive epochs. This not only ensured the model's efficiency but also its efficacy, by retaining the optimal weights discovered during the training phase.

Over the course of 100 epochs, the model was diligently trained, with its performance rigorously evaluated against the validation set. This meticulous training regimen culminated in the application of the trained NARX model to the test set, providing a critical assessment of its predictive capabilities. The successful implementation and evaluation of this NARX model not only demonstrated its potential as a robust tool for financial forecasting but also laid a solid foundation for further exploration and innovation within the field. This endeavour paves the way for the integration of more advanced deep learning architectures, inviting subsequent experimentation and enhancement in the pursuit of unrivalled accuracy in the forecasting of financial time series.

The basic working of the NARX Model can be represented by the below given equation:

Ŷt+1 = f(𝑌𝑡 (𝑑) , 𝑋𝑇𝑡 (𝑑) ) - (1)

Where,

* T - the total number of time periods,
* Y - the variable to be forecast,
* XT - the set of N latent variables used to forecast XT = (XT(1),XT(2),...,XT(N))T,
* dY,dXT =(dXT(1),...,dXT(N))–the corresponding delays.

Also, f is a non-linear function, Yt(d) = {Yt, Yt−1, Yt−2, . . . , Yt−d+1}, XT t(d) = {XTt,XTt−1,XTt−2,...,XTt−d+1}, XTt = (XTt(1),XTt(2),...,XTt(N))T

The architecture of the NARX model can be understood from the below given diagram of the NARX model.

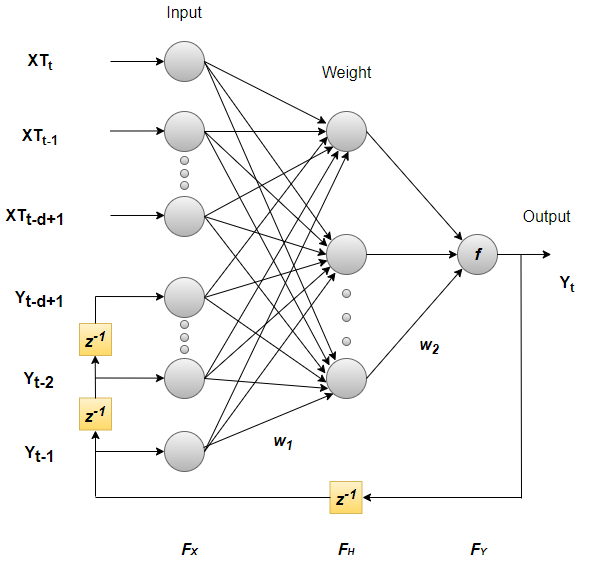


Fig.1. Architecture of NARX Model

3.3 LSTM MODEL

The exploration of advanced predictive models in the realm of stock market forecasting has led to the focused development of a Long Short-Term Memory (LSTM) network, acclaimed for its exceptional ability to capture long-term dependencies within data sequences. The intrinsic value of the LSTM model in this study is its capacity to process input data comprised of sequences with a designated number of lagged time steps, established at 10. This specific configuration was chosen to empower the LSTM network to leverage historical data points crucial for the accurate prediction of future stock prices.

To feed the LSTM model, lagged sequences were meticulously generated from the normalized closing stock price data. This normalization process ensures that the model is provided with data that reflects true market conditions, stripped of any scale-induced biases, thereby allowing the LSTM to learn from structured temporal patterns embedded within the data. A crucial step in the preparation of the data involved dividing it, allocating 80% for training purposes and reserving the remaining 20% for testing. This division is fundamental to the research methodology, as it allows for the validation of the LSTM’s predictive performance on previously unseen data, thus ensuring the model’s efficacy and generalizability across different market conditions.

The architecture of the LSTM model was deliberately constructed with a series of layers specifically designed for sequential data processing. At the forefront is an LSTM layer, adept at managing the temporal sequence of data inputs, followed by a dense layer to facilitate further data processing, culminating in an output layer tasked with generating the final forecast value. The choice of the Adam optimizer for compiling the model underscores the preference for its adaptive learning rate capabilities, which are particularly beneficial for recurrent neural networks by facilitating a more efficient convergence to the optimal set of parameters. The mean squared error (MSE) was selected as the loss function, a conventional choice for regression problems that quantifies the average squared difference between predicted values and actual outcomes, thereby providing a measure of the model’s forecasting accuracy.

To guard against the potential pitfall of overfitting, where the model might perform well on training data but poorly on new, unseen data, an early stopping mechanism was implemented. This mechanism monitored the validation loss during the training phase and halted the training process if no improvement was observed over a predetermined number of epochs. This approach ensures that the model remains robust and capable of generalizing well to new data.

The training regimen for the LSTM model spanned a specified number of epochs, with a validation dataset employed to monitor its performance rigorously throughout the training period. Upon the completion of the training phase, the LSTM model was tasked with making predictions on the test set. This step is crucial, providing valuable insights into the model’s ability to accurately forecast future stock prices, a testament to its potential utility in navigating the volatile movements characteristic of stock markets.

The deployment of the LSTM model in this research marks a pivotal evaluation of the capabilities of recurrent neural networks in the challenging and unpredictable domain of stock price forecasting. The LSTM’s design, emphasizing the processing of sequences with lagged time steps, the careful selection of the model architecture, and the strategic implementation of training and validation techniques, all contribute to a robust methodology aimed at enhancing the accuracy and reliability of stock market predictions. This meticulous approach, coupled with the LSTM’s inherent strengths in handling temporal data, underscores the model’s vital role in advancing the field of financial forecasting, offering promising avenues for future exploration and innovation.

The below given equation expresses the basic working of a LSTM Network.

it = σ(Wi · [ht−1, xt] + bi) - (2)

ft =σ(Wf ·[ht−1,xt]+bf) - (3)

ot = σ(Wo · [ht−1, xt] + bo) – (4)

Where, it represents the input gate, ft represents the forget gate, ot represents the output gate, σ represents the sigmoid function, Wx represents weight for the respective gate (x) neurons, ht−1 represents output of the previous LSTM block (at timestamp t − 1), xt represents input at the current timestamp, bx represents biases for the respective gates (x).

The diagram given below represents the architecture of a LSTM Network.

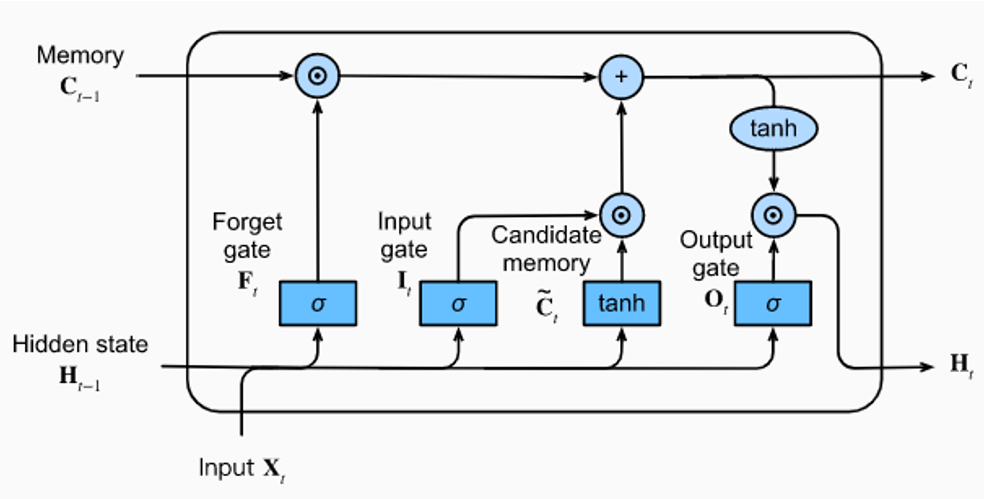


Fig.2. Architecture of LSTM Model

3.4 TESTING THE IMPACT OF TECHNICAL FACTORS

Volume-weighted Average Price (VWAP) is a trading benchmark that provides traders and investors with a glimpse into both the price and volume aspect of a stock's trading period. It's calculated by taking the cumulative sum of the product of a stock's price and its volume, divided by the total volume over a specific time frame, typically a single trading day. VWAP offers a more comprehensive understanding of market trends by integrating volume, which can indicate the strength behind price movements, making it a favoured tool for assessing the market's valuation of a stock throughout the day.

The formula of VWAP can be given as:

VWAP = Sum(V \* P) / Sum(V) – (5)

Where,

Sum = Summation of all values in the time series

V = Volume of the stock traded

P = Price of the stock traded

The 50-day Exponential Moving Average (EMA) is a technical indicator that provides insights into the short-term momentum of a stock's price. Unlike simple moving averages, EMAs give more weight to recent prices, making them more responsive to new information. The 50-day EMA tracks the average price of a stock over the past 50 days, adjusted to emphasize the most recent data. It's often used by traders to identify short-term trends, where prices above the EMA suggest bullish conditions, and those below indicate bearish trends.

The formula of 50 Day EMA can be given as:

EMAT50 = (PriceT \* 2/51) + (EMAY \* (1 – 2/51)) – (6)

Where,

EMAT50 = Value of EMA today

PriceT = Closing Price of Stock Today

EMAY = EMA value from yesterday or the previous trading day

The 200-day Exponential Moving Average (EMA) operates on the same principle as the 50-day EMA but over a longer timeframe. It reflects the average price of a stock over the past 200 days, with a bias towards more recent prices due to its exponential formula. The 200-day EMA is widely regarded as a significant indicator of long-term market trends. Crossing above this EMA is typically seen as a positive sign, suggesting long-term upward momentum, while dropping below it can signal a potential downtrend.

The formula of 50 Day EMA can be given as:

EMAT200 = (PriceT \* 2/201) + (EMAY \* (1 – 2/201)) – (7)

Where,

EMAT200 = Value of EMA today

PriceT = Closing Price of Stock Today

EMAY = EMA value from yesterday or the previous trading day

In the journey to enhance stock price forecasting through advanced deep learning techniques, a particular emphasis is placed on evaluating the role of these technical indicators within the framework of Long Short-Term Memory (LSTM) models. This phase of research, dedicated to "Testing the Effect of Technical Indicators," investigates how the integration of VWAP, 50-day EMA, and 200-day EMA into LSTM models might refine the predictive accuracy of these models. These indicators are selected for their proven ability to offer nuanced insights into market trends and investor sentiments that might escape detection through the analysis of price data alone. By embedding VWAP, along with 50-day and 200-day EMAs into LSTM models, the study embarks on an experimental quest to determine whether these indicators can significantly enhance predictive performance. The process begins with incorporating these indicators as additional features in the dataset, followed by a normalization procedure to ensure that all data types are on a level playing field, contributing equally to the learning process. This preparation is vital for transforming the data into sequences compatible with the LSTM's temporal data processing capabilities.

The crux of the experimental analysis involves training an LSTM model enriched with these technical indicators and comparing its performance to a baseline model devoid of such indicators. This comparison, grounded in the Mean Squared Error (MSE) metric, aims to quantitatively evaluate the impact of the indicators on the model's predictive accuracy. The research is exploratory, acknowledging that the addition of these indicators may not always result in enhanced forecasting accuracy but is instead aimed at empirically discerning their effect.

Should the results demonstrate that the technical indicators substantially improve forecasting accuracy, it would advocate for their inclusion in a more comprehensive ensemble model. Alternatively, should the indicators not yield significant improvement or adversely affect performance, these findings still offer critical insights into the complexities of financial time series forecasting.

This exploration is not only about enhancing model performance but also about understanding the broader implications of technical indicators in financial analytics. By capturing underlying market dynamics and investor sentiments, these indicators have the potential to reveal market trends that are not immediately visible through price analysis alone. This investigation serves as a bridge between traditional financial analysis and modern computational techniques, aiming to combine the strengths of both for superior stock price forecasting.

As this research unfolds, the detailed examination of the effectiveness of these technical indicators will shed light on their true value in the context of advanced deep learning models. This endeavour reaches beyond the technical realm to engage with the core of market psychology and the quest for patterns amidst the market's complexity. Through this comprehensive analysis, the study contributes to a deeper understanding of how deep learning, augmented with established financial metrics, can provide transformative insights into stock market forecasting.

3.5 ENSEMBLE MODEL OF STACKED LSTM & NARX

In the pursuit of advancing financial time series forecasting, the innovative approach of an ensemble model, which synergizes the Nonlinear Autoregressive with Exogenous inputs (NARX) and Stacked Long Short-Term Memory (LSTM) networks, presents a multifaceted solution to a historically complex challenge. This ensemble model is meticulously crafted to harness the distinct advantages of its constituent components, aiming to provide an unprecedented level of accuracy in stock price predictions.

The NARX network component is celebrated for its exceptional ability to model intricate nonlinear systems, especially those influenced by external factors. It excels in identifying and leveraging short-term dependencies and nonlinear patterns prevalent in financial datasets. For the purposes of this research, the NARX network is configured as a straightforward feedforward neural network. It is equipped with an input layer that processes a predetermined number of lagged observations—specifically, 10 previous time steps of normalized closing stock prices. This is followed by a dense hidden layer comprised of 50 neurons, utilizing the ReLU (Rectified Linear Unit) activation function to introduce non-linearity and enable the learning of complex data patterns. The culmination of this network is an output layer featuring a single neuron with a linear activation function, tasked with forecasting the stock price at the next time step.

Conversely, the Stacked LSTM component is engineered to capture and analyze long-term sequential dependencies intrinsic to time series data. LSTMs are renowned for their effectiveness in addressing the vanishing gradient problem, thus ensuring the preservation of information over extended sequences. The model incorporates a stacked arrangement of LSTM layers to deepen its learning capability, allowing it to extract and interpret data at multiple levels of abstraction. This model comprises three LSTM layers, each containing 50 neurons. The first two layers are configured to return full sequences, facilitating the flow of sequential output to subsequent layers. This architectural choice enables the deep network to discern a hierarchy of temporal features at varying levels of detail. The final LSTM layer aggregates the sequence into a singular vector, which is subsequently passed to a dense output layer, reflecting the predictive structure of the NARX model.

The ensemble model achieves its predictive prowess by amalgamating the forecasts from both the NARX and Stacked LSTM models through a straightforward yet efficacious averaging technique. Independently, each model forecasts the test data, which are then converted from their normalized form back to the original scale of stock prices. The ensemble's final prediction is derived by calculating the arithmetic mean of both models' forecasts at every time step, combining their insights into a singular, coherent forecast.

This strategic blending of the NARX model's acute sensitivity to immediate, nonlinear relationships and the Stacked LSTM's proficiency in navigating long-term sequential dependencies represents a comprehensive approach to stock price forecasting. By averaging the outputs, the ensemble model aims to counteract potential biases or overfitting tendencies inherent in individual models, thereby ensuring more reliable and accurate predictions. The ensemble method rests on the principle that a collective of models typically outperforms any single model by capturing a broader spectrum of patterns and relationships within the data.

Ultimately, the 'NARX-Stacked LSTM' ensemble model stands as a sophisticated hybrid framework, specifically designed to tackle the volatility and intricacies of stock market movements. It leverages the complementary strengths of the NARX and LSTM networks, positioning itself as a formidable tool in the arsenal of financial forecasting. This innovative ensemble model not only signifies a leap forward in the methodology of predicting stock prices but also exemplifies the potential of combining diverse deep learning techniques to navigate the complexities of financial markets with enhanced precision and reliability.

The layer by layer analysis diagram of the proposed model can be observed from the below diagram.

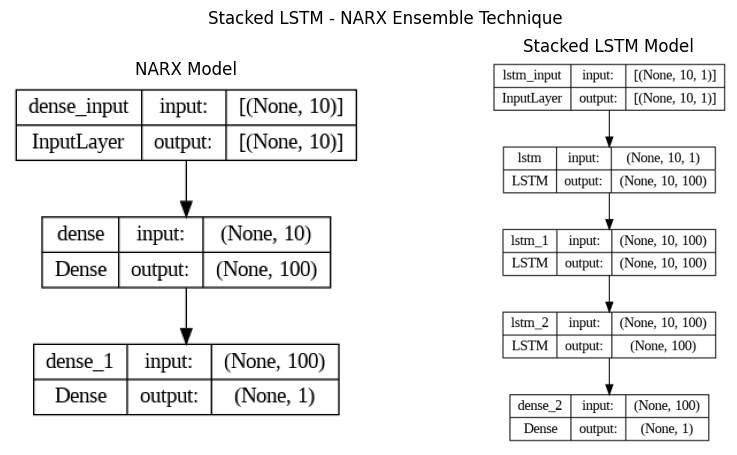


Fig.3. Layer by Layer Architecture Diagram of Stacked LSTM & NARX Ensemble Model

To analyse the Stacked LSTM and NARX Ensemble model in detail, the below given diagram would be providing detailed yet simple to understand insights on how the whole ensemble model would work in the proposed system.

This diagram represents a hybrid deep learning architecture designed for predicting future stock prices. It integrates a Nonlinear Autoregressive with Exogenous Inputs (NARX) model with a Stacked Long Short-Term Memory (LSTM) model, culminating in an ensembled output that averages the predictions from both models for enhanced forecasting accuracy.

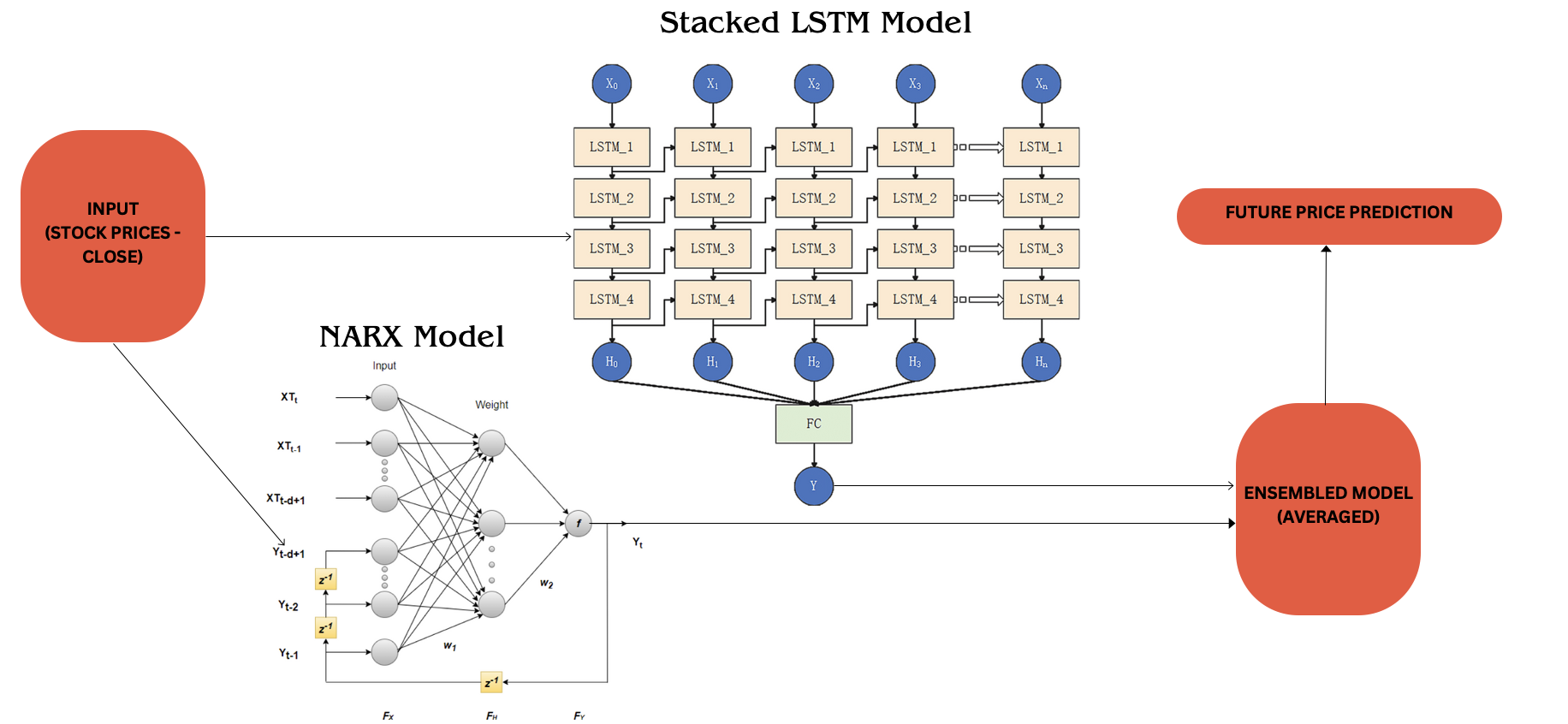


Fig.4. Overall Architecture Diagram of Stacked LSTM & NARX Ensemble Model

To analyse the whole program including the model and all the required pre-processing and to get the efficient and accurate result from the program, the below diagram gives the overall flow of how the proposed system works when implemented, which could be understood on a wider aspect of interpretation.

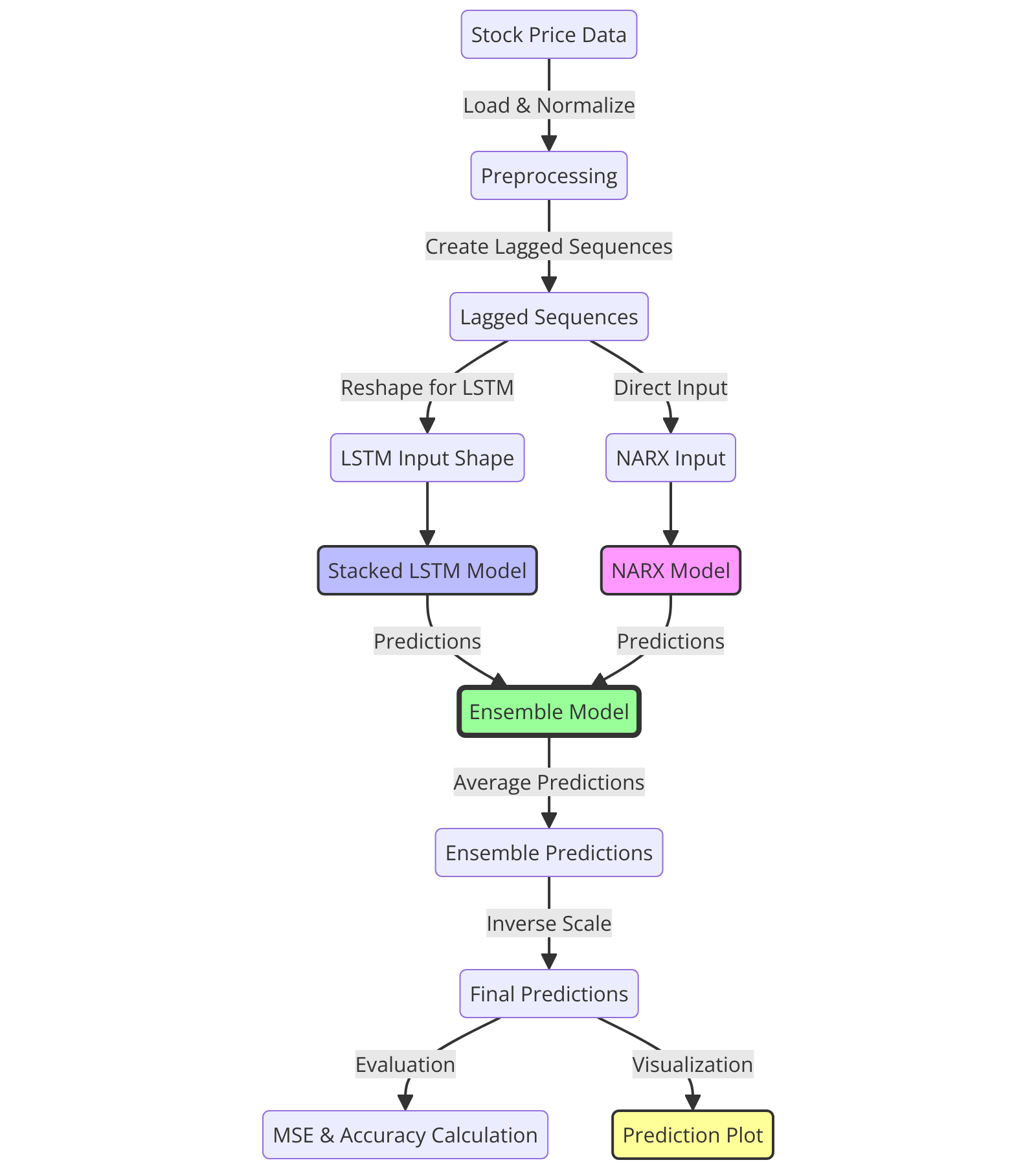


Fig.5. Flow Diagram of the Proposed System (Stacked LSTM & NARX Ensemble Model)

3.6 EXPLANATION OF MODULES FOR THE ENSEMBLE MODEL

The methodology of this project report is centered around a meticulous and methodical process that begins with data acquisition and exploration and concludes with future price forecasting, encompassing a variety of sophisticated techniques and models. Each module of this process is crucial for the development of a robust forecasting model.

1. DATA ACQUISITION AND EXPLORATION

The foundation of any data-driven project is the procurement of quality data. In this study, historical stock price data are retrieved from a comprehensive CSV file containing the historical stock data of the given company. Initial exploration of this data is conducted using Pandas, a powerful data manipulation library in Python, which facilitates the examination of the data structure. Key features, such as 'Date' for chronological order and 'Close' for the final price of the stock at the end of the trading day, are identified, setting the stage for further analysis.

1. DATA CLEANING AND TRANSFORMATION

Data integrity is paramount; thus, any missing data points are addressed through well-established strategies to ensure completeness. This stage also involves feature engineering, where relevant features or indicators are derived to potentially enhance the predictive capacity of the models. These may include technical indicators or derived statistical features that could provide additional insights into stock price movements.

1. TEMPORAL DATA REPRESENTATION

The dimension of time is pivotal in stock price data. Utilizing Pandas' datetime capabilities, the 'Date' column is set as the index, ensuring chronological integrity of the data. Visual exploration follows, with the creation of initial time-series plots to visualize stock price trends and identify possible patterns that may inform the predictive models.

1. DATA NORMALIZATION AND SCALING

To facilitate model stability and convergence, the 'Close' prices are normalized using Min-Max scaling. This process adjusts the data to a common scale without distorting differences in the ranges of values. The Scikit-Learn library provides robust tools for this normalization, ensuring seamless and efficient data scaling operations.

1. SEQUENCE GENERATION FOR LSTM MODELS

For LSTM models, which are adept at handling sequential data, lagged sequences are generated. A custom function is developed to create these sequences with a specified number of lagged time steps, which are essential for capturing the temporal dependencies within the data. The generated sequences are then formulated into input-output pairs, a crucial step for training the LSTM models.

1. TRAIN-TEST SPLIT AND VALIDATION

The dataset is divided into training and testing sets with an 80-20 temporal splitting, considering the time-series nature of the data. A portion of the training data is set aside as validation data to monitor and enhance the generalization capabilities of the models during the training phase.

1. NARX MODEL IMPLEMENTATION

The NARX model is implemented using a Sequential modeling approach, featuring a Dense input layer with 100 units to process the lagged inputs and a Dense output layer for predictions. The ReLU activation function is chosen for its ability to introduce non-linearity, which is key for modeling complex patterns.

1. STACKED LSTM ARCHITECTURE

A Stacked LSTM model is constructed with a layered structure, consisting of three LSTM layers to increase the representational power of the network. Each LSTM layer has 50 neurons, and the model utilizes Mean Squared Error as the loss function, conducive to training stability and performance.

1. MODEL TRAINING AND EVALUATION

The models are trained iteratively over a number of epochs, with performance metrics monitored for both the training and validation sets. Early stopping is implemented to prevent overfitting, halting the training process when the validation loss no longer shows improvement.

1. ENSEMBLE MODEL CONSTRUCTION

An ensemble model is constructed by fusing the predictions from both the NARX and Stacked LSTM models through ensemble averaging. The performance of the ensemble model is then quantified using accuracy percentage and MSE to evaluate its predictive strength.

1. VISUALIZING PREDICTIONS AND MODEL COMPARISON

To present the results in an accessible manner, a time-series plot is created, comparing the actual stock prices with the ensemble model’s predictions. This visual comparison provides a platform for extracting insights and discussing the observed model behaviours.

1. FUTURE PRICE FORECASTING

For future price forecasting, the initial input setup is prepared using the last 'n\_steps' from the normalized dataset. The models then iteratively predict stock prices for a specified number of future days, with the NARX and Stacked LSTM models both contributing to this forecast. The predicted values, initially in a normalized scale, are inversely transformed back to the original stock price scale to provide a meaningful interpretation. This 30-day forecast aims to demonstrate the model's reliability and accuracy under future, uncertain market conditions.

3.7 ALGORITHM FOR THE ENSEMBLE MODEL

Step 1: Import necessary libraries

Step 2: Load and preprocess data in .csv format

Step 3: Create sequences

Step 4: Normalize data

Step 5: Split data into training and testing sets

Step 6: Reshape data

Step 7: Define and train the NARX model

Step 8: Predict with the NARX model

Step 9: Define and train the Stacked LSTM model

Step 10: Predict with the Stacked LSTM model

Step 11: Ensemble prediction

Step 12: Calculate MSE

Step 13: Plot the results

Step 14: Forecast future prices (30 Days into the future)

Note: For the testing of the other algorithms (LSTM, NARX & LSTM with additional parameters) only the steps that correspond to the their respective models are carried out as the pre-processing and the result generation are of the same process for testing all algorithms.

**Chapter 4**

**Implementation**

4.1 LSTM MODEL

The implementation process for stock price prediction using an LSTM model involves several stages, each carefully executed to ensure the model learns effectively from historical data and makes accurate future predictions.

1. DATA PREPARATION

# Assuming 'dataset.csv' is your dataset file

data = pd.read\_csv("dataset.csv")

data = data[["Close"]] # Use only the 'Close' column

data = data.dropna()

The initial stage is data preparation, where 'dataset.csv' is read into a pandas DataFrame with a focus on the 'Close' price column. This column is crucial as it represents the final price at which the stock settles at the end of a trading day. To guarantee the consistency of data, any missing values are identified and removed. This step is essential because LSTM models, which excel at capturing temporal dependencies, require complete sequences without any gaps.

1. DATA NORMALIZATION

# Normalize the data

scaler = MinMaxScaler(feature\_range=(0, 1))

scaled\_data = scaler.fit\_transform(data)

Following data preparation, the next stage is data normalization. This involves scaling the 'Close' prices within a range between 0 and 1 using the MinMaxScaler. Normalization is a critical step in the modeling process as it helps the LSTM network to learn more efficiently by providing data within a scale that is known to work well with the network's activation functions.

1. SEQUENCE CREATION

# Create sequences and labels for training

def create\_sequences(data, seq\_length):

sequences, labels = [], []

for i in range(len(data) - seq\_length):

seq = data[i : i + seq\_length]

label = data[i + seq\_length]

sequences.append(seq)

labels.append(label)

return np.array(sequences), np.array(labels)

sequence\_length = 50 # Adjust this based on your preference

X, y = create\_sequences(scaled\_data, sequence\_length)

The normalized data is then organized into sequences that serve as the input for the LSTM model. These sequences are created to have a specific length, and each one is paired with a corresponding label that the model will learn to predict. This pairing of sequences and labels is what allows the LSTM to understand the patterns in the stock price movements over time.

1. TRAIN-TEST SPLIT

# Split the data into training and testing sets

train\_size = int(len(X) \* 0.8)

X\_train, X\_test = X[:train\_size], X[train\_size:]

y\_train, y\_test = y[:train\_size], y[train\_size:]

After sequence creation, the data is divided into a training set and a test set, with the training set typically constituting 80% of the data. The training set is used to train the model, while the test set is reserved for evaluating the model's performance later on.

1. RESHAPING INPUT DATA FOR LSTM

# Reshape input for LSTM

X\_train = np.reshape(X\_train, (X\_train.shape[0], X\_train.shape[1], 1))

X\_test = np.reshape(X\_test, (X\_test.shape[0], X\_test.shape[1], 1))

Prior to model training, the input data must be reshaped to match the LSTM's requirements. The LSTM expects the input to be a three-dimensional array where the dimensions represent the number of samples, the number of time steps, and the number of features, respectively.

1. BUILDING LSTM MODEL

# Build a LSTM model

model = Sequential()

model.add(LSTM(units=50, input\_shape=(X\_train.shape[1], 1))) # Single LSTM layer

model.add(Dense(units=1)) # Output layer

The LSTM model is built with a simple yet effective architecture consisting of an LSTM layer followed by a dense layer. The LSTM layer is configured with 50 units, which allows the model to capture the complexity of the stock price sequence data. The dense layer, on the other hand, culminates the model's architecture by providing the output prediction for the 'Close' price.

1. COMPILING & TRAINING THE MODEL

model.compile(optimizer="adam", loss="mean\_squared\_error")

# Train the model

model.fit(

X\_train, y\_train, epochs=20, batch\_size=32, verbose=1

)

The model is compiled using the 'adam' optimizer, known for its efficiency, and the mean squared error (MSE) loss function, which is appropriate for regression tasks like stock price prediction. The model is then trained on the training set for a predefined number of epochs, with the progress being displayed during training due to the verbosity setting.

1. PREDICTIONS & VISUAL PLOTTING

# Predictions

predictions = model.predict(X\_test)

predictions = scaler.inverse\_transform(

predictions

) # Inverse transform to original scale

y\_test\_rescaled = scaler.inverse\_transform(

y\_test.reshape(-1, 1)

)

# Visualization

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

plt.plot(predictions, label="Predicted Close Price", color="blue")

plt.plot(y\_test\_rescaled, label="Real Close Price", color="red")

plt.title("Stock Price Prediction - Simplified Model")

plt.xlabel("Time")

plt.ylabel("Close Price")

plt.legend()

Next, predictions are made on the test set. These predictions, initially on the same scale as the normalized data, are then inversely transformed back to the original stock price scale. This allows for a direct comparison between the model's predictions and the actual 'Close' prices. A graphical plot is then presented, showcasing the predicted 'Close' prices against the real 'Close' prices, giving a visual appraisal of the model's predictive accuracy.

1. EVALUATION USING PERFORMANCE METRICS

# Calculate MSE Assuming 'predictions' and 'y\_test\_rescaled' are your predicted and actual values respectively

import numpy as np

mse = np.mean(np.square(predictions - y\_test\_rescaled))

print(f"Mean Squared Error: {mse}")

import numpy as np

def calculate\_custom\_accuracy(y\_true, y\_pred, threshold=0.05):

"""

Calculate the percentage of predictions that are within a specified threshold.

:param y\_true: The actual values.

:param y\_pred: The predicted values.

:param threshold: The threshold for considering predictions as accurate (default is 6%).

:return: The accuracy percentage.

"""

# Ensure y\_true and y\_pred are numpy arrays

y\_true = np.array(y\_true)

y\_pred = np.array(y\_pred)

# Calculate the absolute percentage error

percentage\_errors = np.abs((y\_true - y\_pred) / y\_true)

# Determine which errors are below the threshold

accurate\_predictions = percentage\_errors < threshold

# Calculate the accuracy percentage

accuracy\_percentage = np.mean(accurate\_predictions) \* 100

return accuracy\_percentage

# Assuming 'predictions' and 'y\_test' have been defined in your previous code

# Flatten the arrays to ensure they are 1D

y\_test\_flattened = y\_test\_rescaled.flatten()

predictions\_flattened = predictions.flatten()

# Calculate accuracy with a threshold of 5%

accuracy\_percentage = calculate\_custom\_accuracy(

y\_test\_flattened, predictions\_flattened, threshold=0.05

)

print(f"Accuracy Percentage (within 5% error): {accuracy\_percentage}%")

The model's MSE is evaluated on the test set, providing a clear indication of the model's prediction error on unseen data. The MSE is a critical measure as it reflects how closely the model's predictions match the actual stock prices. The final step in the evaluation process involves calculating the model's prediction accuracy. An accuracy function is defined to compute the percentage of predictions that fall within a specified threshold of the actual prices. For this implementation, a threshold of 5% is used, recognizing the inherent fluctuations in stock prices. The accuracy percentage thus obtained offers a gauge of the model's reliability in making predictions that are within an acceptable range of the true values.

4.2 NARX MODEL

1. DATA PREPARATION

# Load the stock price dataset

df = pd.read\_csv("datset.csv")

df["Date"] = pd.to\_datetime(df["Date"])

df = df.set\_index("Date")

df = df.dropna()

To begin with, the task of predicting stock prices using a NARX model involves a sequence of structured steps. The initial stage is to prepare the data adequately. This involves loading the dataset into a pandas dataframe and converting the 'Date' column into a datetime format for better manipulation and indexing. Ensuring that the data is free from missing values is essential for maintaining the integrity and reliability of the predictive model.

1. DATA NORMALIZATION

# Normalize the 'Close' prices

scaler = MinMaxScaler(feature\_range=(0, 1))

df["Close\_Normalized"] = scaler.fit\_transform(df["Close"].values.reshape(-1, 1))

Once the data is prepared, the next crucial step is to normalize the 'Close' prices. By using a MinMaxScaler, the 'Close' prices are scaled between 0 and 1, creating a new column in the dataframe to hold these normalized values. The purpose of this scaling is to stabilize the model's training process and improve the predictive performance.

1. SEQUENCE CREATION

# Function to create lagged sequences for time series data

def create\_sequences(data, n\_steps):

X, y = [], []

for i in range(len(data) - n\_steps):

X.append(data[i : (i + n\_steps)])

y.append(data[i + n\_steps])

return np.array(X), np.array(y)

# Define the number of lagged time steps

n\_steps = 10

X, y = create\_sequences(df["Close\_Normalized"].values, n\_steps)

The creation of input-output sequences is pivotal in modelling time series data. A specially designed function, `create\_sequences`, constructs these sequences, taking into account a predefined number of time steps or lags. These sequences are what the model will learn from, understanding the dependencies and patterns inherent in historical price changes.

1. DATA SPLITTING

# Split the data into training and testing sets

train\_size = int(len(X) \* 0.8)

X\_train, X\_test = X[:train\_size], X[train\_size:]

y\_train, y\_test = y[:train\_size], y[train\_size:]

After constructing the sequences, the data is split into training and testing sets, adhering to an 80-20 split. This separation allows the model to be trained on a majority of the data while being validated and tested on unseen data, ensuring that the model's performance is not just a reflection of overfitting to the training set but also indicative of its ability to generalize.

1. BUILDING NARX MODEL

# Define the NARX model

narx\_model = Sequential()

narx\_model.add(Dense(10, activation="tanh", input\_dim=n\_steps))

narx\_model.add(Dense(1))

narx\_model.compile(optimizer="adam", loss="mse")

Building the architecture of the NARX model is the next step. A simple but effective feedforward neural network with a sequence of layers is set up, starting with a dense layer featuring 'tanh' activation to map non-linear relationships. The model is compiled using the 'adam' optimizer, with mean squared error as the loss function to provide a quantitative measure of the model's prediction error during training.

1. EARLY STOPPING

# Define early stopping to prevent overfitting

early\_stopping = EarlyStopping(

monitor="val\_loss", patience=5, restore\_best\_weights=True

)

To avoid the common pitfall of overfitting, early stopping is employed. It monitors the validation loss and halts the training process if no improvement is observed over a certain number of epochs. This technique ensures that the model retains the most effective weights and does not memorize the training data

1. MODEL TRAINING

# Train the NARX model

narx\_model.fit(

X\_train,

y\_train,

epochs=100,

validation\_data=(X\_test, y\_test),

callbacks=[early\_stopping],

verbose=2,

)

Training the model is a delicate process where the model learns to predict future stock prices based on the provided sequences. The number of epochs and the batch size dictate the training's intensity and duration. Once the model is trained, predictions are made on the test set to evaluate its performance.

1. PREDICTIONS & VISUAL PLOTTING

# Make predictions on the test set using the NARX model

y\_pred\_narx = narx\_model.predict(X\_test)

# Inverse transform the normalized predictions to the original scale

y\_pred\_narx\_inv = scaler.inverse\_transform(y\_pred\_narx)

y\_test\_inv = scaler.inverse\_transform(y\_test.reshape(-1, 1))

# Plot the real vs predicted values for the NARX model

plt.figure(figsize=(10, 6))

plt.plot(

df.index[train\_size + n\_steps :], y\_test\_inv, label="Real Prices", color="blue"

)

plt.plot(

df.index[train\_size + n\_steps :],

y\_pred\_narx\_inv,

label="NARX Predictions",

color="red",

linestyle="dashed",

)

plt.title("Stock Price Prediction using NARX Model")

plt.xlabel("Date")

plt.ylabel("Close Price")

plt.legend()

plt.show()

The values have been predicted on the test set. A visual comparison of the model's predictions and the actual stock prices is presented in a graph. The visual representation allows for an intuitive understanding of how well the model's predictions align with reality.

1. EVALUATION USING PERFORMANCE METRICS

from sklearn.metrics import mean\_squared\_error

import numpy as np

# Function to calculate accuracy

def calculate\_accuracy(real, predicted, threshold\_percent=5):

threshold = threshold\_percent / 100.0

diff = np.abs((real - predicted) / real)

correct = np.less\_equal(diff, threshold)

accuracy = np.mean(correct) \* 100

return accuracy

# Calculate and print the Mean Squared Error (MSE)

mse\_narx = mean\_squared\_error(y\_test\_inv, y\_pred\_narx\_inv)

print(f"NARX Model Mean Squared Error (MSE): {mse\_narx}")

# Calculate accuracy using the provided function

accuracy\_narx = calculate\_accuracy(y\_test\_inv, y\_pred\_narx\_inv)

print(f"NARX Model Accuracy: {accuracy\_narx}%")

Assessing the model's performance involves calculating the Mean Squared Error (MSE) on the test data, reflecting the average prediction error across all predictions. Additionally, a visual comparison of the model's predictions and the actual stock prices is presented in a graph. The visual representation allows for an intuitive understanding of how well the model's predictions align with reality. Lastly, to quantitatively determine the model's predictive accuracy, a custom accuracy calculation is implemented. Predictions within a 5% range of the real stock prices are considered accurate. This bespoke threshold accommodates the volatile nature of stock prices, acknowledging the model's effectiveness in capturing the overall trend rather than penalizing slight deviations.

4.3 LSTM MODEL WITH ADDITIONAL PARAMETERS

1. DATA LOADING & FEATURE CALCULATION

# Load your dataset (replace 'your\_data.csv' with your actual file)

data = pd.read\_csv("TS.csv")

columns = ["Close", "Volume"]

data = data[columns]

data = data.dropna()

# Calculate additional features (EMA, VWAP)

data["200EMA"] = data["Close"].ewm(span=200, adjust=False).mean()

data["50EMA"] = data["Close"].ewm(span=50, adjust=False).mean()

data["VWAP"] = (data["Close"] \* data["Volume"]).cumsum() / data["Volume"].cumsum()

The exploration begins with the retrieval of the dataset, particularly focusing on the 'Close' and 'Volume' columns to underpin our prediction model. The ingenuity of our approach is further enriched by the introduction of calculated features like the 200-day and 50-day Exponential Moving Averages (EMAs) and the Volume Weighted Average Price (VWAP). These indicators are crafted to tap into the underlying momentum and weighted price trends, infusing our model with a depth of market insight.

1. DATA NORMALIZATION

# Normalize the data

scaler = MinMaxScaler(feature\_range=(0, 1))

scaled\_data = scaler.fit\_transform(data)

With the application of MinMaxScaler, our dataset undergoes a transformation, normalizing our features into a uniform scale that enhances the model's ability to learn effectively. The data metamorphoses into a series of inputs and labels, courtesy of a custom sequence creation function tailored to encapsulate the temporal dynamics essential for LSTM processing.

1. SEQUENCE CREATION

# Create sequences and labels for training

def create\_sequences(data, seq\_length):

sequences, labels = [], []

for i in range(len(data) - seq\_length):

seq = data[i : i + seq\_length]

label = data[i + seq\_length]

sequences.append(seq)

labels.append(label)

return np.array(sequences), np.array(labels)

sequence\_length = 50

X, y = create\_sequences(scaled\_data, sequence\_length)

The construction of input-output sequences is critical in modelling time series data. A specially developed function, 'create\_sequences', creates these sequences while accounting for a preset amount of time steps or lags. The model will learn from these sequences to comprehend the dependencies and patterns inherent in historical price fluctuations.

1. DATA SPLITTING

# Split the data into training and testing sets

train\_size = int(len(X) \* 0.8)

X\_train, X\_test = X[:train\_size], X[train\_size:]

y\_train, y\_test = y[:train\_size], y[train\_size:]

The data is partitioned conscientiously, reserving 80% for the crucible of training and setting aside 20% to test the model's prowess.

1. BUILDING LSTM MODEL

# Build an improved LSTM model

model = Sequential()

model.add(

LSTM(

units=100,

return\_sequences=True,

input\_shape=(X\_train.shape[1], X\_train.shape[2]),

)

)

)

model.add(Dropout(0.2)) # Add dropout for regularization

model.add(LSTM(units=100, return\_sequences=True))

model.add(Dropout(0.2))

model.add(LSTM(units=100))

model.add(Dropout(0.2))

model.add(Dense(units=scaled\_data.shape[1]))

The architecture of our LSTM model is a symphony of complexity and refinement, with a trio of LSTM layers each comprising 100 units. The layers are interlaced with Dropout to combat overfitting, ensuring that our model remains robust and generalizable.

1. MODEL COMPILATION & TRAINING

# Use a lower learning rate

model.compile(optimizer=Adam(learning\_rate=0.001), loss="mean\_squared\_error")

# Train the model with more epochs

model.fit(X\_train, y\_train, epochs=100, batch\_size=64, validation\_data=(X\_test, y\_test))

The model's compilation is marked by the precision of the Adam optimizer, deliberately calibrated with a lower learning rate to fine-tune our convergence to the stock market's rhythm. Training spans across 100 epochs, with validation intermissions using the test set as a sounding board for the model's accuracy.

1. PREDICTIONS & VISUAL PLOTTING

# Plot a line graph between Predicted and Real Price

plt.figure(figsize=(12, 6))

plt.plot(predictions[:, 0], label="Predicted Close Price")

plt.plot(y\_test[:, 0], label="Real Close Price")

plt.title("Predicted vs Real Close Price")

plt.xlabel("Time")

plt.ylabel("Close Price")

plt.legend()

plt.show()

The values have been projected for the test set. A graph is used to visually compare the model's predictions to real stock prices. The visual depiction provides an intuitive grasp of how well the model's predictions correspond to reality.

1. EVALUATION USING PERFORMANCE METRICS

# Calculate MSE for the improved model

mse = np.mean(np.square(predictions[:, 0] - y\_test[:, 0]))

print(f"Mean Squared Error: {mse}")

# Define a tolerance for how close the prediction needs to be to be considered "correct"

tolerance = 0.05

diff = np.abs(predictions - y\_test)

# Calculate the percentage difference

pct\_diff = diff / y\_test

# Calculate accuracy

accuracy = np.mean(pct\_diff < tolerance)

print(f"Accuracy: {accuracy \* 100:.2f}%")

To evaluate the model's performance, calculate the Mean Squared Error (MSE) on the test data, which is the average prediction error over all predictions. A graph compares the model's predictions to real stock prices. The visual depiction provides an intuitive grasp of how well the model's predictions correspond to reality. Finally, to quantify the model's predicted performance, a unique accuracy calculation is used. Predictions made within a 5% range of actual stock values are considered accurate. This custom threshold accounts for the volatility nature of stock prices, recognizing the model's ability in capturing the broad trend rather than penalizing minor deviations.

4.4 STACKED LSTM – NARX ENSMBLE MODEL

1. DATA LOADING & CLEANING

# Load the stock price dataset

df = pd.read\_csv("dataset.csv")

df["Date"] = pd.to\_datetime(df["Date"])

df = df.set\_index("Date")

df = df.dropna()

The code starts by loading a stock price dataset into a pandas DataFrame, converting the 'Date' column to datetime format, and setting it as the DataFrame index. This establishes a time series data structure that is essential for temporal analysis. It performs a drop operation for any NA values, ensuring the data integrity for the analysis.

1. DATA NORMALIZATION

# Normalize the 'Close' prices

scaler = MinMaxScaler(feature\_range=(0, 1))

df["Close\_Normalized"] = scaler.fit\_transform(df["Close"].values.reshape(-1, 1))

The 'Close' prices are normalized between 0 and 1 using `MinMaxScaler`. This step is crucial for neural network models to ensure all input features have a similar scale, contributing to more stable and faster convergence during training.

1. SEQUENCE CREATION

# Define the number of lagged time steps

n\_steps = 10

# Function to create lagged sequences for time series data

def create\_sequences(data, n\_steps):

X, y = [], []

for i in range(len(data) - n\_steps):

X.append(data[i : (i + n\_steps)])

y.append(data[i + n\_steps])

return np.array(X), np.array(y)

X, y = create\_sequences(df["Close\_Normalized"].values, n\_steps)

A custom function constructs sequences from the normalized data, with each sequence consisting of `n\_steps` lagged values of 'Close' prices. This process is pivotal for time series forecasting, where the goal is to predict future stock prices based on a window of past observations.

1. DATA SPLITTING

# Split the data into training and testing sets

train\_size = int(len(X) \* 0.8)

X\_train, X\_test = X[:train\_size], X[train\_size:]

y\_train, y\_test = y[:train\_size], y[train\_size:]

X\_train\_reshaped = X\_train.reshape((X\_train.shape[0], X\_train.shape[1], 1))

X\_test\_reshaped = X\_test.reshape((X\_test.shape[0], X\_test.shape[1], 1))

As usual, the data is partitioned conscientiously, reserving 80% for the crucible of training and setting aside 20% to test the model's prowess.

1. MODEL BUILDING AND COMPILING

# Adjusted learning rate

learning\_rate = 0.001

# Define and compile the NARX model

narx\_model = Sequential()

narx\_model.add(Dense(100, activation="tanh", input\_dim=n\_steps)) # Increased units

narx\_model.add(Dense(1))

narx\_model.compile(optimizer=Adam(learning\_rate=learning\_rate), loss="mse")

# Define and compile the Stacked LSTM model

stacked\_lstm\_model = Sequential()

stacked\_lstm\_model.add(

LSTM(100, return\_sequences=True, input\_shape=(n\_steps, 1))

) # Increased units

stacked\_lstm\_model.add(LSTM(100, return\_sequences=True)) # Increased units

stacked\_lstm\_model.add(LSTM(100)) # Increased units

stacked\_lstm\_model.add(Dense(1))

stacked\_lstm\_model.compile(optimizer=Adam(learning\_rate=learning\_rate), loss="mse")

# Early stopping to prevent overfitting

early\_stopping = EarlyStopping(

monitor="val\_loss", patience=10, restore\_best\_weights=True

)

The NARX configuration includes a Dense layer with 100 units and a 'tanh' activation, followed by a Dense output layer. This Stacked LSTM is built specifically for time series forecasting, with three LSTM layers each containing 100 units. The first two LSTM layers return sequences to feed into the subsequent LSTM layer, enabling the model to learn from the sequence's temporal structure. The final Dense layer outputs the forecasted value. This stacked architecture allows the model to capture complex temporal dependencies in the data. Like the NARX model, it's compiled using the Adam optimizer and MSE loss.

1. MODEL TRAINING

# Adjusted batch size

batch\_size = 32

# Train the NARX model

narx\_model.fit(

X\_train,

y\_train,

epochs=200,

batch\_size=batch\_size,

validation\_data=(X\_test, y\_test),

callbacks=[early\_stopping],

verbose=2,

)

# Train the Stacked LSTM model

stacked\_lstm\_model.fit(

X\_train\_reshaped,

y\_train,

epochs=200,

batch\_size=batch\_size,

validation\_data=(X\_test\_reshaped, y\_test),

callbacks=[early\_stopping],

verbose=2,

)

Both models are trained with their respective datasets, applying early stopping based on validation performance to mitigate overfitting.

1. PREDICTIONS & VISUAL PLOTTING

# Make predictions

y\_pred\_narx = narx\_model.predict(X\_test)

y\_pred\_stacked\_lstm = stacked\_lstm\_model.predict(X\_test\_reshaped)

# Inverse transform predictions to original scale

y\_pred\_narx\_inv = scaler.inverse\_transform(y\_pred\_narx)

y\_pred\_stacked\_lstm\_inv = scaler.inverse\_transform(y\_pred\_stacked\_lstm)

y\_test\_inv = scaler.inverse\_transform(y\_test.reshape(-1, 1))

# Ensemble predictions

ensemble\_predictions = (y\_pred\_narx\_inv + y\_pred\_stacked\_lstm\_inv) / 2

# Calculate MSE for the ensemble

mse\_ensemble = mean\_squared\_error(y\_test\_inv, ensemble\_predictions)

print(f"Ensemble Mean Squared Error (MSE): {mse\_ensemble}")

# Plot real vs predicted values

plt.figure(figsize=(10, 6))

plt.plot(

df.index[train\_size + n\_steps :], y\_test\_inv, label="Real Prices", color="blue"

)

plt.plot(

df.index[train\_size + n\_steps :],

ensemble\_predictions,

label="Ensemble Predictions",

color="green",

linestyle="dashed",

)

plt.title("Improved Stock Price Prediction using Ensemble Model")

plt.xlabel("Date")

plt.ylabel("Close Price")

plt.legend()

plt.show()

After training, both models make predictions on the test dataset. Predictions are then scaled back to the original price range using the inverse of the MinMaxScaler. Predictions from both models are averaged to form ensemble predictions. This method aims to combine the predictive strengths of both models to improve overall accuracy. A plot visualizes the actual stock prices against the ensemble predictions, providing an intuitive comparison of the model's performance over time.

1. EVALUATION USING PERFORMANCE METRICS

from sklearn.metrics import mean\_squared\_error

# Function to calculate accuracy

def calculate\_accuracy(real, predicted, threshold\_percent=5):

threshold = threshold\_percent / 100.0

diff = np.abs((real - predicted) / real)

correct = np.less\_equal(diff, threshold)

accuracy = np.mean(correct) \* 100

return accuracy

# Calculate MSE for the ensemble

mse\_ensemble = mean\_squared\_error(y\_test\_inv, ensemble\_predictions)

print(f"Ensemble Mean Squared Error (MSE): {mse\_ensemble}")

# Calculate accuracy using the provided function

accuracy\_ensemble = calculate\_accuracy(y\_test\_inv, ensemble\_predictions)

print(f"Ensemble Model Accuracy: {accuracy\_ensemble}%")

To assess the efficacy of the model, the Mean Squared Error (MSE) metric is computed on the testing dataset to determine the average error across all predictions. The model's forecasted values are graphically represented alongside the actual stock prices, offering a clear visual comparison of the model's predictive accuracy against the actual market trends. In addition to the MSE, a specialized accuracy metric is implemented. This metric considers predictions falling within a 5% margin of the true stock values as correct, acknowledging the inherent fluctuations in stock prices and allowing for a slight margin of error, thus highlighting the model's capability to track the general direction of stock movements rather than exact values.

**Chapter 5**

**Results**

5.1 BRIEF INTRODUCTION TO RESULTS

In the sphere of stock market forecasting, this project report delves into a comprehensive analysis, investigating the predictive performance of various modelling techniques across a selection of stocks. The assets under study are divided by currency, with two represented in US dollars—Google and Nike—and two in Indian Rupees—ITC and Tata Steel. This diverse selection is intentional, allowing for a robust test of the models' effectiveness across different economic environments and currency valuations.

The methodological approach of this study encompasses four distinct experimental variations of forecasting models. The first is the Long Short-Term Memory (LSTM) model, known for its proficiency in capturing long-term dependencies in time series data. The second is the Nonlinear Autoregressive with Exogenous inputs (NARX) model, an architecture tailored to map complex non-linear relationships influenced by external factors. The third variation extends the LSTM model by integrating technical indicators, a method aiming to enrich the model with nuanced market insights. Lastly, the Ensemble model represents the culmination of this research, merging the Stacked LSTM and NARX models to potentially boost the accuracy and consistency of stock price predictions.

The results of these models are meticulously evaluated based on two key metrics: the percentage of accuracy and the Mean Square Error (MSE). Accuracy percentage provides a straightforward indicator of the model's performance in predicting the correct stock price movements, while MSE offers a measure of the model's prediction error magnitude. These metrics are essential for not only quantifying the models' performances but also for comparing the effectiveness of different modelling approaches.

The findings of this study are presented through a combination of graphical and tabular representations. Graphs offer an immediate visual interpretation of the models' predictive capabilities over time, allowing for a dynamic comparison between the predicted and actual stock prices. Tables complement these visual aids by providing a detailed numerical breakdown of performance metrics, facilitating a precise and analytical assessment of each model's forecasting prowess.

This section, 'Brief Introduction to Results’, sets the stage for a deeper discussion that follows, unveiling the strengths and weaknesses of each model and their implications for the field of financial forecasting. It also provides a critical foundation for future research directions, guiding subsequent developments in stock price prediction modelling.

5.2 GRAPHICAL REPRESENTATION & ANALYSIS OF RESULTS

In the quest to refine stock price forecasting, the project delves into a critical comparative analysis through "Graphical Representation & Analysis of Results." This section is dedicated to a visual comparison of various models like the LSTM, NARX, and LSTM enhanced with additional technical indicators against the innovative Ensemble Model that combines Stacked LSTM and NARX models. These models are meticulously evaluated using the datasets of prominent companies such as Google, Nike, ITC, and Tata Steel.

The graphical illustrations are pivotal in this section, offering a visual interpretation of the predictive effectiveness of each model on the test sets. The prediction graphs serve as a visual testimony to the models' proficiency in estimating future stock prices. In these graphs, the actual closing prices are plotted alongside the predicted values, providing an immediate visual cue to the models' accuracy. The X-axis of these graphs represents the timeline over which the predictions were made, while the Y-axis depicts the closing prices, allowing for a direct comparison between predicted and actual market behavior.

The Ensemble Model, a sophisticated hybrid of Stacked LSTM and NARX models, is expected to showcase a higher degree of correlation with the actual stock prices, indicating the strength of its predictive analytics. The graphical results for the Ensemble Model are particularly significant, as they encapsulate the cumulative predictive insights gleaned from the individual strengths of both the Stacked LSTM and NARX methodologies. The graphical representation is not merely about plotting lines on a chart; it is a narrative that unfolds the story of each model's predictive journey. It highlights the areas where the models align closely with the actual stock prices and where they diverge, offering an at-a-glance understanding of each model's performance nuances. Such visual depictions are essential for identifying the models' responses to market volatility and their ability to capture the subtle cues that dictate stock price movements.

Moreover, these graphs are instrumental in discerning the added value brought by the Ensemble Model. By overlaying the predictions from the individual and combined models, the graphs facilitate a clear, comparative understanding of the incremental benefits offered by the ensemble approach. Observing the Ensemble Model’s graph will reveal its effectiveness in smoothing out anomalies and providing a consistent predictive output, which is crucial for stakeholders who rely on accurate forecasts for decision-making.

This section, replete with detailed graphs, not only serves as a visual affirmation of the numerical findings but also provides an intuitive understanding of the models' predictive behaviors. The comparison of the Ensemble Model with its constituent models in graphical form is a testament to the power of visual communication in demonstrating complex data relationships in a format that is accessible and engaging for all readers. As the project report progresses, these graphical representations will offer a compelling visual dialogue about the advancements made in the field of financial forecasting through deep learning.

The plots in this section have been plotted with the Close Price on the X-axis and the Time on the Y-Axis The graphical representation of results and its analysis are displayed as below.

1. GOOGLE DATASET

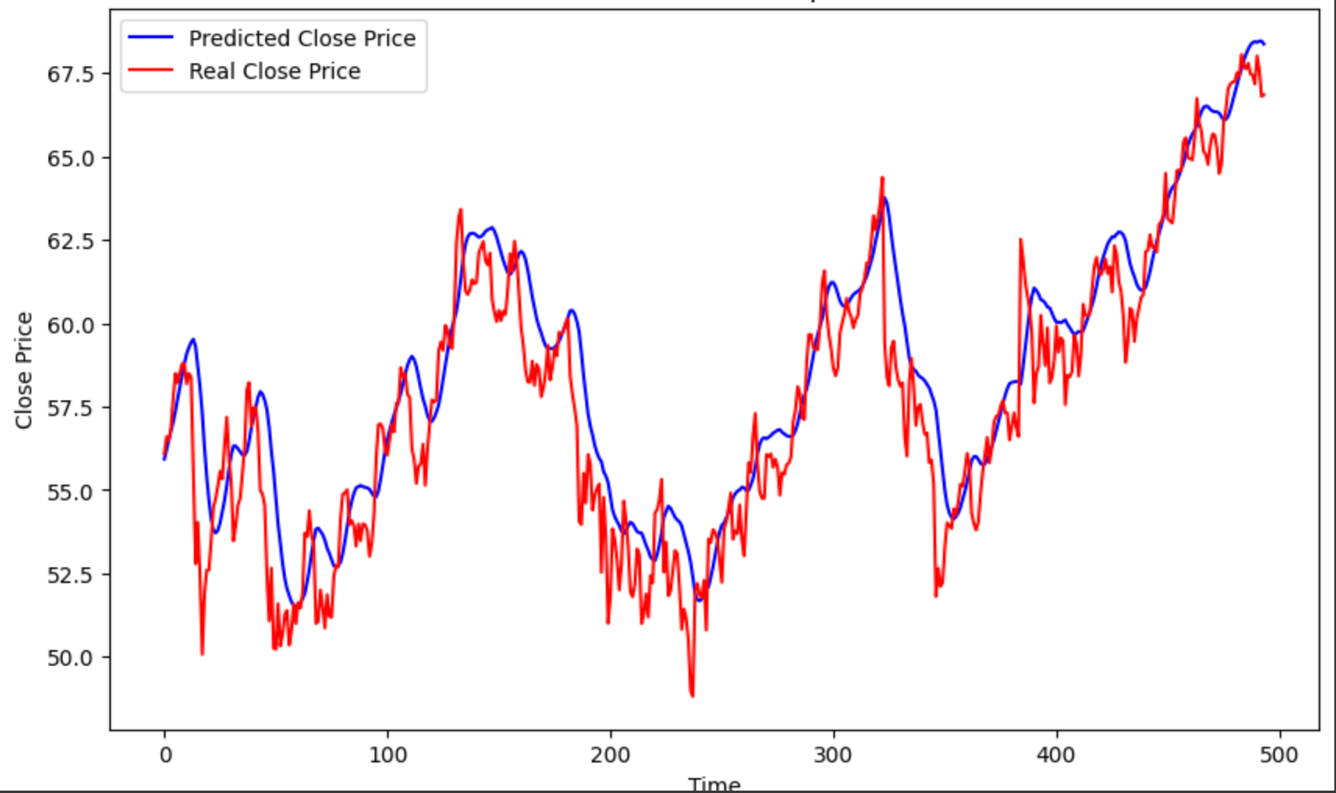


Fig.6. GOOGLE Dataset – LSTM Results

The above provided graph illustrates the predictions made by an LSTM model on Google's stock prices, juxtaposed with the actual market data. The blue line denotes the LSTM's predictions, while the red line shows the actual closing prices.

From the graph, it appears that the LSTM model has a moderate grasp on the overall direction in which Google's stock prices are heading. The predicted trends generally seem to shadow the real price movements, suggesting that the model has captured some of the broader trends in the historical data. However, there is a noticeable discrepancy between the predicted and actual values, especially during periods of quick price changes or market volatility. The model seems to lag in capturing sharp rises and drops in stock prices, potentially indicating a delay in the model's response to new market information. It also appears to smooth out the predictions, which could be useful in reducing the noise of daily price fluctuations, but this smoothing effect may lead to a less accurate reflection of the market's rapid movements. While the LSTM model has demonstrated an ability to follow the stock's price trajectory to some extent, the prediction line doesn't consistently match the actual closing prices. This moderate performance suggests that while there is potential in using LSTMs for stock price prediction, there's substantial room for improvement in model tuning or possibly integrating additional data sources to better capture the complexities of the stock market.

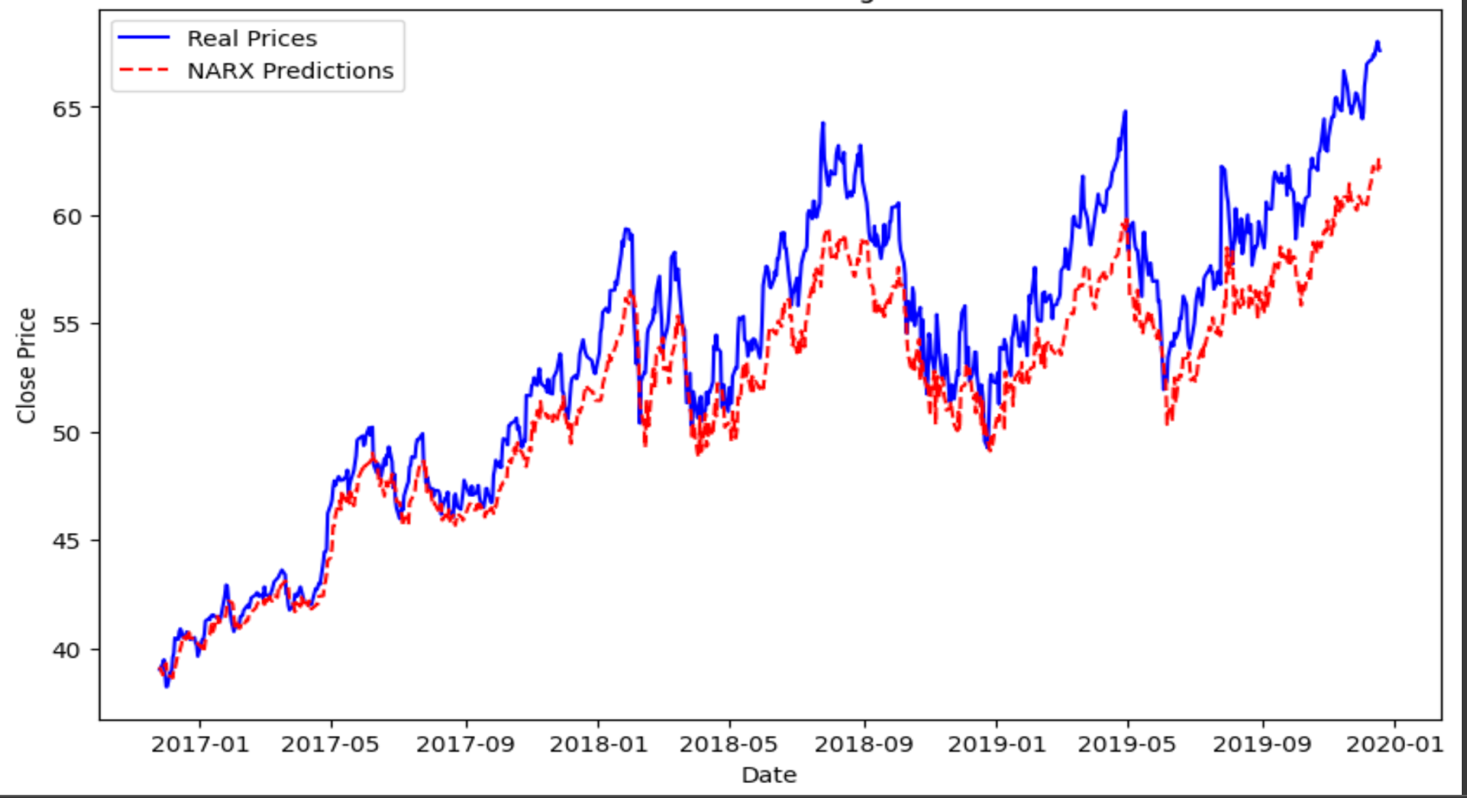


Fig.7. GOOGLE Dataset –NARX Results

The graphical representation presented above shows the performance of a NARX (Nonlinear Autoregressive with Exogenous inputs) model in predicting Google's stock prices, with the actual prices depicted by the blue line and the NARX model's predictions represented by the dashed red line.

In assessing the NARX model's predictive capability, it's clear that the model can follow the general trend of the stock's price movement over the observed period. While the NARX predictions seem to grasp the overall upward and downward trends in the market, there is a noticeable variation between the predicted and the actual prices, particularly during periods of high volatility and significant price changes. The NARX model appears to lag in real-time responsiveness, which could suggest limitations in how the model processes sudden shifts in stock price trends. Additionally, while the model captures the broader swings in price, it seems to struggle with the finer points of the market's day-to-day fluctuations, leading to a series of overestimations and underestimations as compared to the actual closing prices. Despite these discrepancies, the model does exhibit a degree of predictive power, indicating that it has learned to some extent the complex dynamics that drive Google's stock prices. However, the moderate fit shown by the NARX model underscores the challenges in stock market prediction and highlights potential areas for model improvement, such as parameter tuning or the integration of additional market features that could enable a more refined analysis of stock price movements.



Fig.8. GOOGLE Dataset – LSTM With Additional Parameters Results

The graph above, depicts a comparison between the actual closing prices of Google's stock and the predicted prices obtained from an LSTM model that has been enhanced with additional technical indicators: Volume Weighted Average Price (VWAP), 50-Day Exponential Moving Average (EMA), and 200-Day EMA. Upon examination of the graph, the incorporation of these technical indicators seems to have provided the LSTM model with additional layers of data, allowing for a more nuanced prediction. The orange line, representing the model's predictions, follows the blue line, which illustrates the real prices, suggesting a decent level of accuracy. The model captures the general upward and downward trends in Google's stock prices and adheres closely to the actual price trajectory, especially in the stable market phases.

However, there are periods where the prediction does not perfectly align with the real prices, particularly in more volatile market conditions, where the stock exhibits sharper rises or falls. This indicates that while the inclusion of VWAP, 50 Day EMA, and 200 Day EMA has equipped the LSTM model to better understand and predict the stock's behavior, there is still room for improvement in the model's ability to respond to more rapid market changes. The overall impression from the graph is that the LSTM model with additional technical indicators holds significant promise for stock price prediction, capturing both the rhythm and the pulse of Google's stock movements with a degree of finesse. The model seems to interpret the flow of the market with an informed perspective, thanks to the depth of information provided by the technical indicators. Nonetheless, the subtle nuances of stock price forecasting, particularly in forecasting peak and trough patterns, pose a continuing challenge, one that could potentially be met with further refinement of the model or by incorporating additional market sentiment analysis.

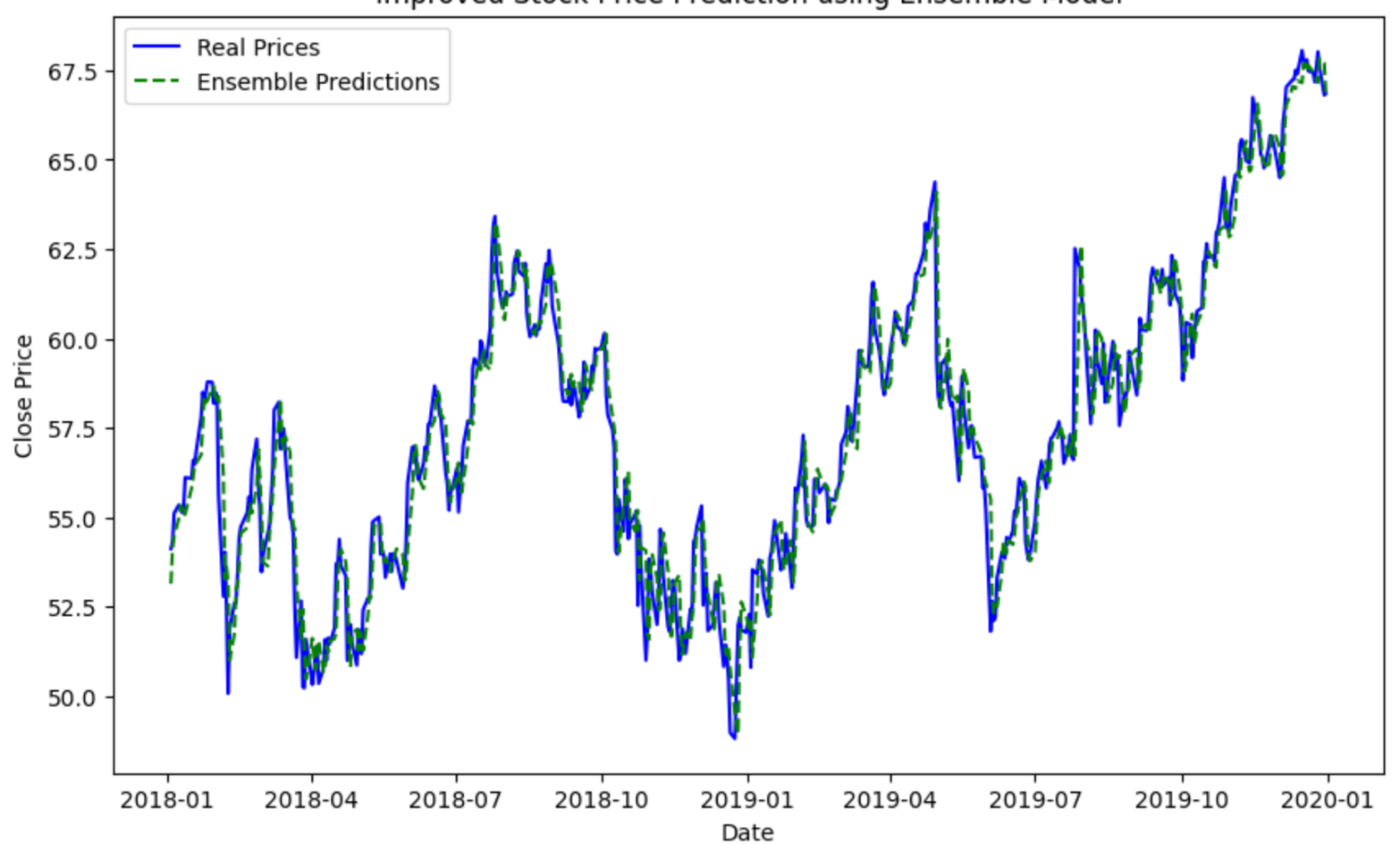
****

Fig.9. GOOGLE Dataset –Ensemble Model of Stacked LSTM & NARX Results

The graph provided showcases the performance of an Ensemble Model, which combines the strengths of Stacked LSTM and NARX models, in predicting the stock prices of Google. The solid blue line represents the actual stock prices, while the dashed green line indicates the predictions made by the Ensemble Model. Reviewing the graph, it's noticeable that the Ensemble Model closely shadows the real stock prices, suggesting a well-tuned synergy between the LSTM's ability to capture long-term dependencies and the NARX model's skill in modeling nonlinear relationships. The model seems to have a firm grasp on the overall trends, rising and falling in concert with the actual prices, and even navigating through the volatility with a notable degree of fidelity.

Like any model, it is not without its slight imperfections. There are moments where the predicted values do not entirely coincide with the actual figures, particularly at points where the market experiences sharper fluctuations. These instances highlight the challenging nature of perfectly modeling stock market behaviors, which are influenced by a complex web of unpredictable variables. Nevertheless, the graph reflects the Ensemble Model's robust predictive capabilities, capable of understanding and anticipating the movement of Google's stock prices with a significant level of accuracy. This suggests that the combined power of Stacked LSTM and NARX models could be a valuable asset for analysts and investors looking to gain insights into future market trends. The graph does not just chart a series of price points; it narrates the intricate dance between prediction and reality, showing a model that learns from the past to inform the future

1. NIKE DATASET



Fig.10. NIKE Dataset – LSTM Results

The graph under scrutiny provides a visual comparison between the actual market closing prices of Nike's stock and the forecasted figures derived from an LSTM model. In this depiction, the LSTM's predictions are represented by the blue line, while the red line mirrors the real-world closing prices of the stock. A careful examination of the graph suggests that the LSTM model captures the general trajectory of Nike's stock prices with moderate success. It reflects an understanding of the overarching movements in the historical price data, with the model's predicted trends often aligning with the actual price path. However, the model's grasp on the stock's behaviour is not without its challenges. There are evident gaps between the predicted and actual values, particularly during times when the stock price experiences rapid shifts or heightened volatility. These discrepancies hint at a potential hesitancy in the model's ability to adapt swiftly to new developments in the market, such as sudden spikes or dips in the stock price. The model's tendency to produce smoothed predictions is a double-edged sword; on one hand, it filters out the daily price noise, providing a cleaner overall trend line. On the other hand, this smoothing can sometimes oversimplify the complex, real-time fluctuations that characterize the stock market's behaviour.

The LSTM model's performance with the Nike dataset indicates a proficiency in tracking the stock's price trend, albeit with a noticeable margin for error. This level of performance underscores the possibilities and limitations of using LSTM models for stock price predictions. It serves as a reminder that there is ample scope for refining the model—perhaps by fine-tuning its parameters or integrating diverse datasets—to achieve a more precise understanding of the nuances in stock price movements.

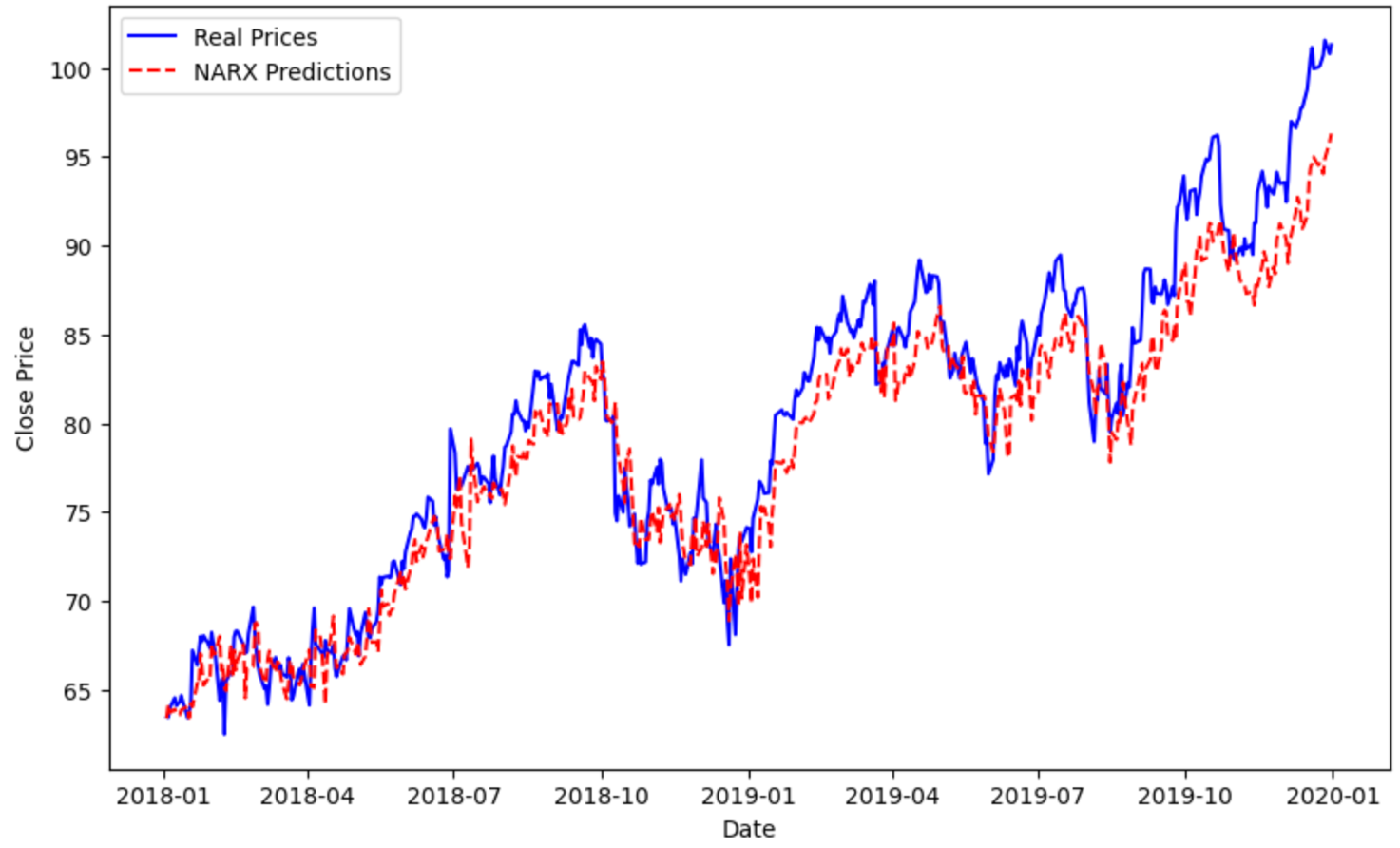
****

Fig.11. NIKE Dataset – NARX Results

The graph in focus paints a picture of the NARX model's attempt at capturing Nike's stock price movements, with the actual stock trajectories plotted in blue and the NARX model's forecast in dashed red. The NARX model, tuned to tease out the complex patterns in stock data, shows an adeptness at tracing the overarching ebb and flow of Nike's stock prices. It captures the stock’s general pattern, ascending and descending in tandem with the market's actual performance. Yet, the spaces between the red forecast line and the true blue of market prices reveal the model’s struggles—particularly evident during tumultuous market periods where stock prices take sharp turns. Here, the model's predictions play catch-up, reflecting a certain inertia when the market quickly changes course.

This behaviour is indicative of a model that, while it can catch the larger waves of market movement, may miss the subtleties of daily stock price vibrations. The instances of over and underestimation scattered throughout the graph suggest that the NARX model, though competent, has room to grow in sophistication. The potential to refine its parameters or to weave in a broader spectrum of market indicators is clear, with the goal of crafting a model that not only follows the market's rhythm but also anticipates its intricate dance steps with greater precision.

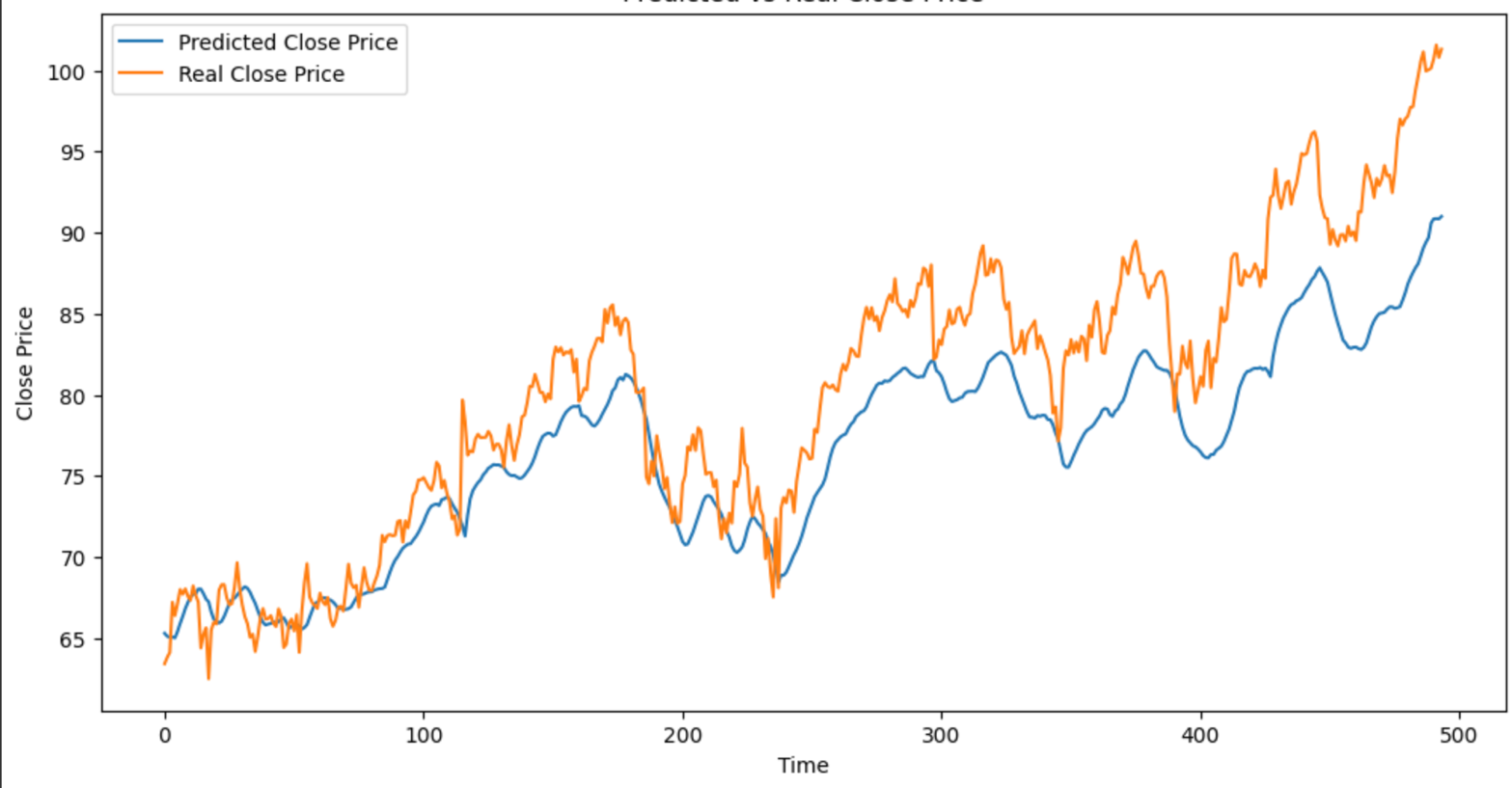
****

Fig.12. NIKE Dataset – LSTM With Additional Parameters Results

The chart presented offers insight into the predictive analysis of Nike's stock prices, carried out through an LSTM model that has been enriched with additional technical indicators, namely the Volume Weighted Average Price (VWAP), and the 50 and 200-Day Exponential Moving Averages (EMAs). In this depiction, the model's forecasts are traced in blue, while the actual closing prices are marked in orange.

A glance at the graph reveals that the LSTM model, supplemented with these indicators, has achieved a certain resonance with the actual price movements of Nike's stock. The model's forecasts, which ride alongside the market's actual closing prices, reflect an understanding of the stock's rhythm, capturing the general trends and shifts in value over time. However, the model's predictions are not without their variations from reality. At several points, particularly where the stock displays more pronounced fluctuations, the model's trajectory diverts from the actual prices, suggesting a certain limitation in its predictive accuracy. These departures from the actual closing prices indicate that while the model benefits from the integration of VWAP and EMA indicators, which provide a richer context for the stock's potential movements, it still faces challenges in fully encapsulating the complexity of market behaviours. The nuanced dance of stock prices, influenced by a myriad of factors beyond historical data, requires a level of adaptability and foresight that the current model has yet to fully master.

In summary, while the LSTM model equipped with additional technical indicators demonstrates an approach to forecasting, as evidenced by its ability to generally track the trajectory of Nike's stock prices, it also highlights the intricate balance between data-driven predictions and the unpredictable nature of the stock market. There is a palpable opportunity for refinement, suggesting that future iterations of the model could benefit from further tuning and perhaps the inclusion of even more diverse datasets or alternative analytical indicators.

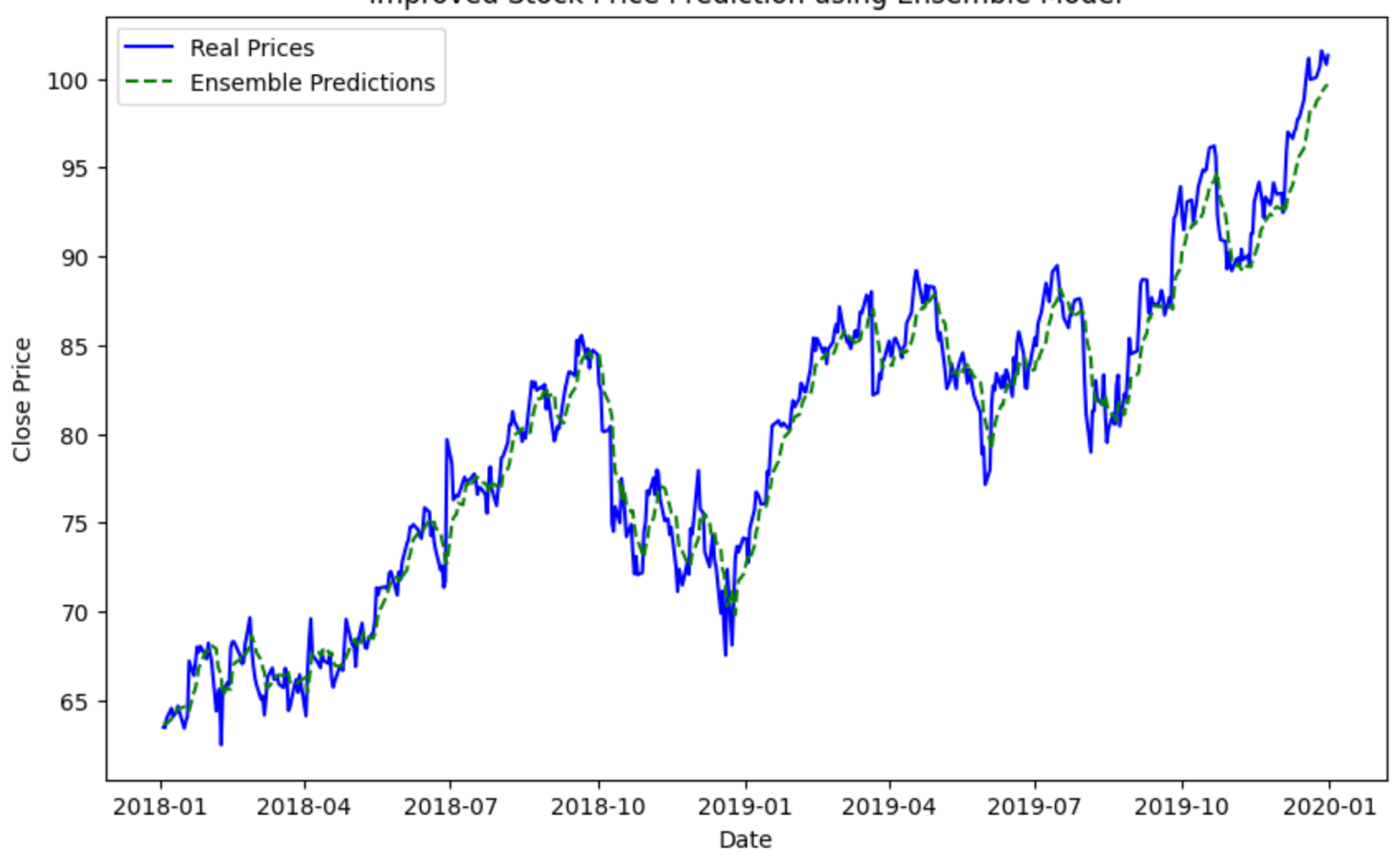
****

Fig.13. NIKE Dataset –Ensemble Model of Stacked LSTM & NARX Results

The provided graph offers a visual assessment of an Ensemble Model's proficiency in forecasting Nike's stock prices. This model is a strategic blend of Stacked LSTM and NARX models, leveraging their collective strengths. The actual stock prices are plotted as a solid blue line, while the Ensemble Model's forecasts are depicted as a dashed green line. Upon examining the graph, it's apparent that the Ensemble Model adeptly mirrors the actual price movements of Nike's stock. This alignment is indicative of the harmonious interplay between the LSTM model's knack for deciphering long-term sequential data and the NARX model's capability to unravel complex, nonlinear patterns. The Ensemble Model rides alongside the market's tide, capturing the rises and dips of the stock prices with commendable accuracy, and exhibiting resilience even when faced with market volatility.

Yet, no model is infallible, and the Ensemble Model is no exception. It occasionally deviates slighlty from the actual stock prices, especially during moments of abrupt market changes. These discrepancies underscore the inherent complexities of the stock market, a domain rife with unpredictable elements that challenge even the most sophisticated of predictive models. Despite these variances, the Ensemble Model's performance on the Nike dataset is strikingly solid, demonstrating a deep understanding and a credible anticipation of stock price trajectories. This aptitude for prediction posits the Ensemble Model as a significant analytical tool for those seeking to navigate the future currents of the stock market. The narrative woven by the graph is not just about the predicted versus actual prices; it's about the model's journey through a landscape of data, learning from historical patterns to forecast what lies ahead on the market's horizon.

1. ITC DATASET

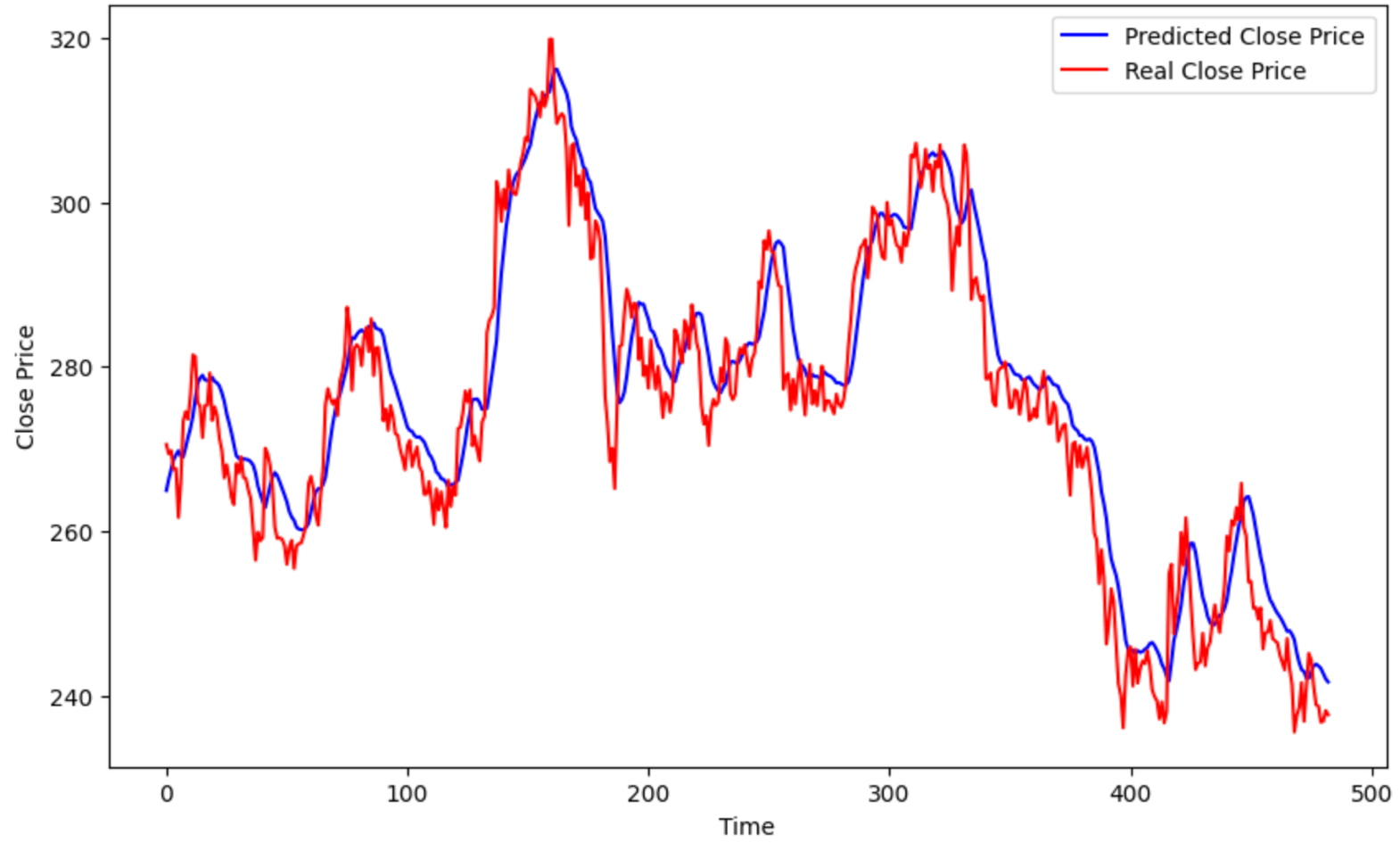


Fig.14. ITC Dataset – LSTM Results

The graph presents an analysis of ITC's stock price predictions made by an LSTM model, portrayed by the blue line, alongside the actual closing prices indicated in red. The narrative told by the blue line speaks to the LSTM model's earnest attempt to trace the complex movements of ITC's stock prices. The model, with its roots deeply embedded in learning from past data, appears to follow the general trend of the stock's journey through the marketplace. It rises and falls, echoing the real stock's movements, revealing an understanding of the broader strokes of the stock's behaviour. Despite its earnest efforts, the model's rendition of the stock's path shows moments of discordance with the actual prices. The model's rendition occasionally strays from the true trajectory, particularly at points where the stock takes a sudden dive or leap, illustrating the difficulty in capturing the market's sometimes abrupt and sharp twists and turns. These are the moments that remind us of the unpredictable heartbeat of the market, where countless unseen factors can lead to unexpected outcomes.

The LSTM model's performance, as seen on the graph, underscores the delicate balance between data-driven foresight and the inherent unpredictability of the stock market. It reflects the model's potential in providing guidance on the probable direction of stock prices while also highlighting the need for continuous learning and adaptation to fully harness the rhythm of the financial markets. The blue line is not just a series of predictions; it's a testament to the model's ongoing quest to decode the enigmatic patterns that govern the ebb and flow of stock prices.

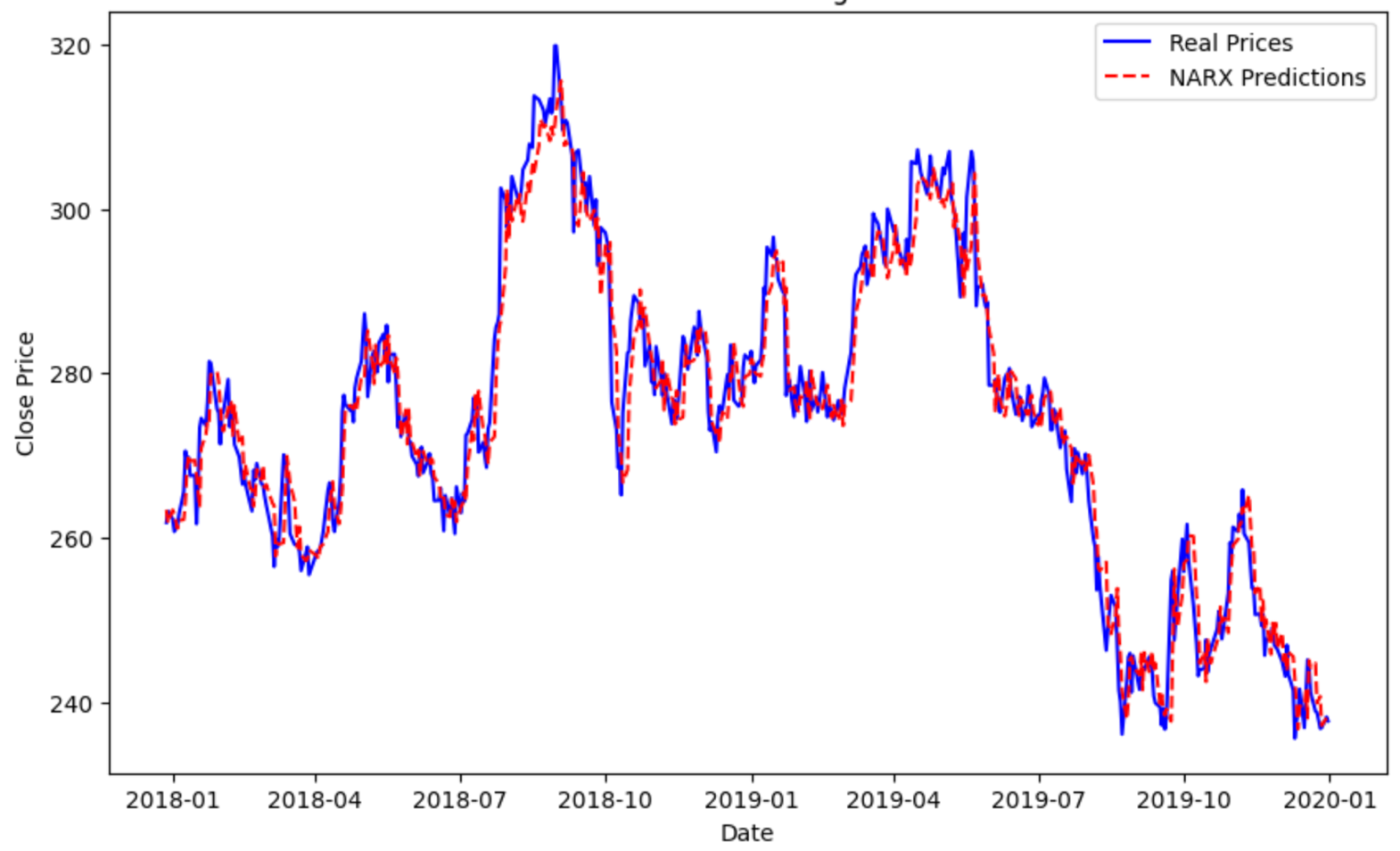


Fig.15. ITC Dataset – NARX Results

The chart presents the outcomes of employing a NARX model to forecast the stock prices of ITC, illustrated with the real stock prices in solid blue and the NARX model's predictions in dashed red. Upon scrutiny, it's evident that the NARX model has managed to capture the ebb and flow of ITC's stock prices with a certain degree of acumen. The predictions undulate closely with the actual price line, reflecting the model's capability to discern the stock's overall price trajectory. Yet, this dance between prediction and reality isn't without its missteps. The model occasionally stumbles, evidenced by the red dashes that diverge from the blue reality, especially in instances where the stock undergoes swift and sharp price changes.

These moments of divergence, where the model's foresight doesn't quite align with the stock's actual performance, underscore the unpredictable nature of financial markets, filled with intricacies that a single model may not fully grasp. While the NARX model has showcased its potential in tracing the general market trend, the stock's sudden peaks and valleys pose a challenge that the model has yet to fully surmount. In essence, the NARX model's rendition of ITC's stock price movements paints a picture of a predictive tool that, while insightful, acknowledges the complexity of financial forecasting. It's a tool that has learned much from historical data, yet recognizes the journey ahead, filled with opportunities for refinement and learning in the quest for a deeper understanding of the market's capricious nature.

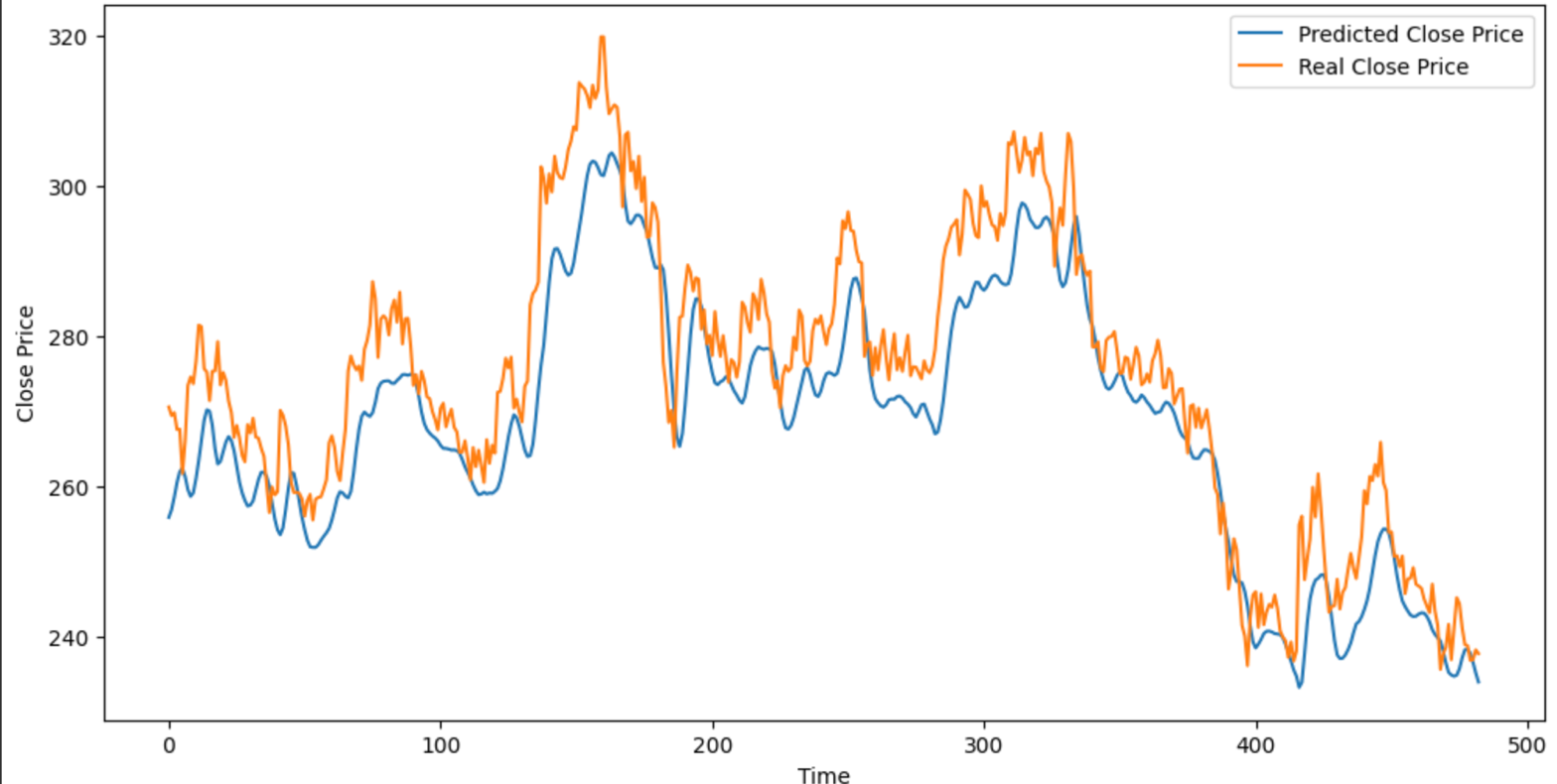
****

Fig.16. ITC Dataset – LSTM With Additional Parameters Results

The graph illustrates the performance of an LSTM model enhanced with additional technical indicators (VWAP, 50-Day EMA, and 200-Day EMA) in predicting the stock prices of ITC. The blue line indicates the actual closing prices, while the orange line represents the model's predictions. Analyzing the chart, it becomes evident that the LSTM model, augmented with these sophisticated indicators, has endeavored to trace the contours of ITC's stock price movements with a reasonable degree of precision. The model's predictions generally trend alongside the actual prices, suggesting an aptitude for capturing the stock's overall direction and momentum. This ability to parallel the actual price path indicates that the model has assimilated the nuanced interplay of price with volume and moving averages, key factors that influence stock valuation.

However, the journey of prediction is not seamless. The model's trajectory occasionally diverges from the actual closing prices, revealing limitations in its forecasting accuracy. These discrepancies, particularly noticeable during periods of swift price changes, underscore the model's challenge in keeping pace with the stock's more abrupt and volatile shifts. Despite these variances, the enhanced LSTM model has shown itself to be a diligent student of the market's past behavior, drawing from the depth provided by VWAP and EMA indicators to inform its forecasts. The graph narrates a story of a model striving to reflect the complex reality of the stock market, highlighting both its learned insights and its areas for growth. The model stands as a testament to the potent combination of machine learning and market intelligence, offering a window into the future of stock prices, even as it acknowledges the unpredictable nature of financial markets.

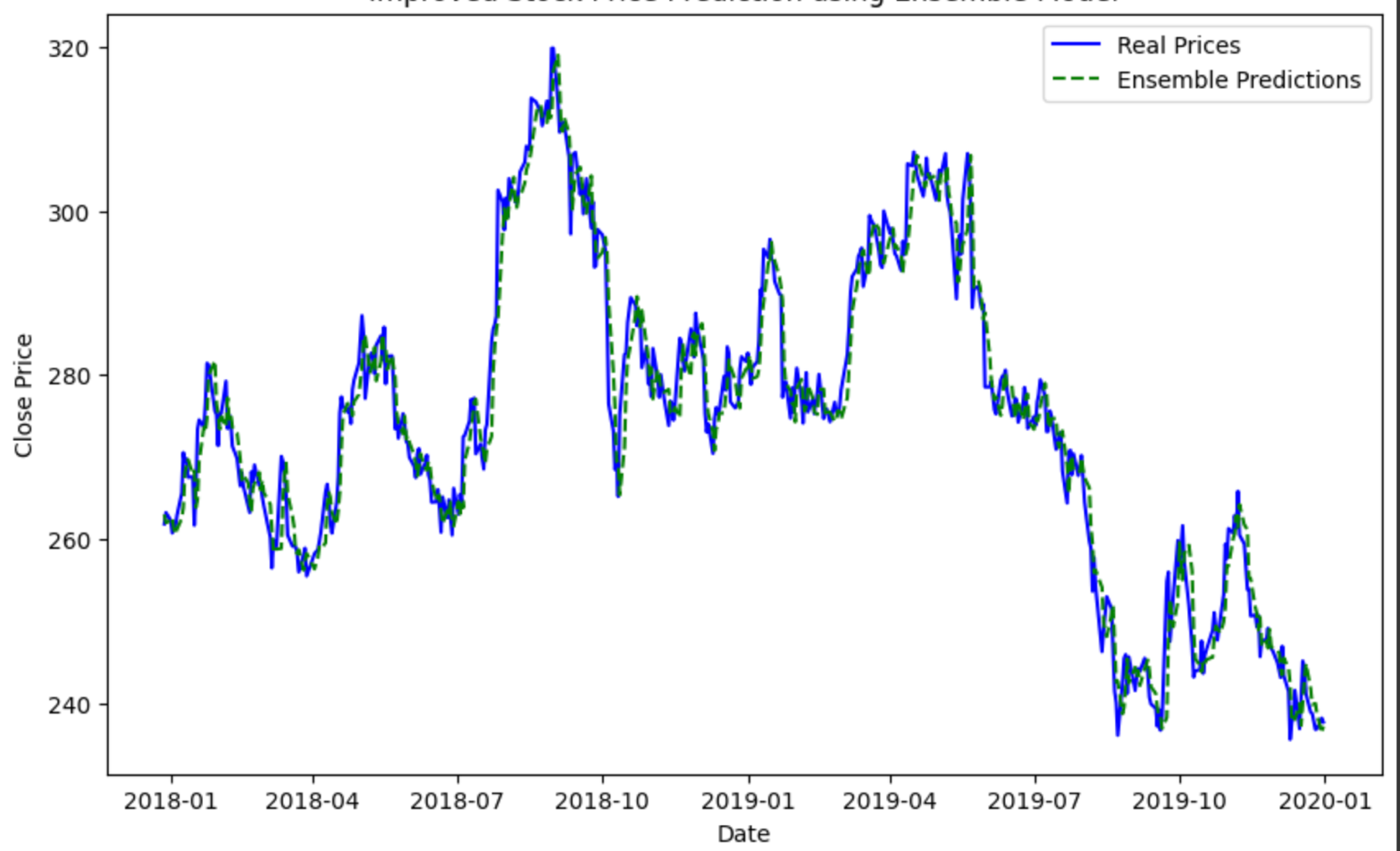
****

Fig.17. ITC Dataset – Ensemble Model of Stacked LSTM & NARX Results

The graph offers a visual comparison of ITC's stock prices as captured by the concerted efforts of an Ensemble Model, fusing Stacked LSTM with NARX, against the backdrop of the actual market prices. The solid blue line charts the real stock prices, while the green dashed line traces the predictions from the Ensemble Model. The Ensemble Model appears to be in a harmonious tandem with the actual prices, reflecting a deep understanding of the stock's behavior through the lenses of both long-term trends and non-linear patterns. The model's ability to echo the real-life undulations of ITC's stock prices suggests a sophisticated level of learning from historical data.

Yet, as with any predictive endeavor, the model's forecasts are not without their moments of discord. The relation between the predicted and actual prices is generally synchronized, but there are small instances where the green dashes step out of line, particularly when the stock experiences sudden turns in the market. These moments serve as humble reminders of the unpredictable tides within the financial markets that even a robust Ensemble Model can sometimes struggle to foresee. Despite these occasional deviations, the Ensemble Model's portrayal of ITC's stock prices tells a story of a technology that has matured enough to offer significant insights into the stock market's future, though it still respects the inherent uncertainty of economic forecasts. This graph is not merely a collection of data points; it's a testament to the model's endeavor to distill wisdom from historical patterns and to cast a light on the path ahead, albeit with an understanding that the road is sometimes shrouded in the mists of market volatility.

1. TATA STEEL DATASET

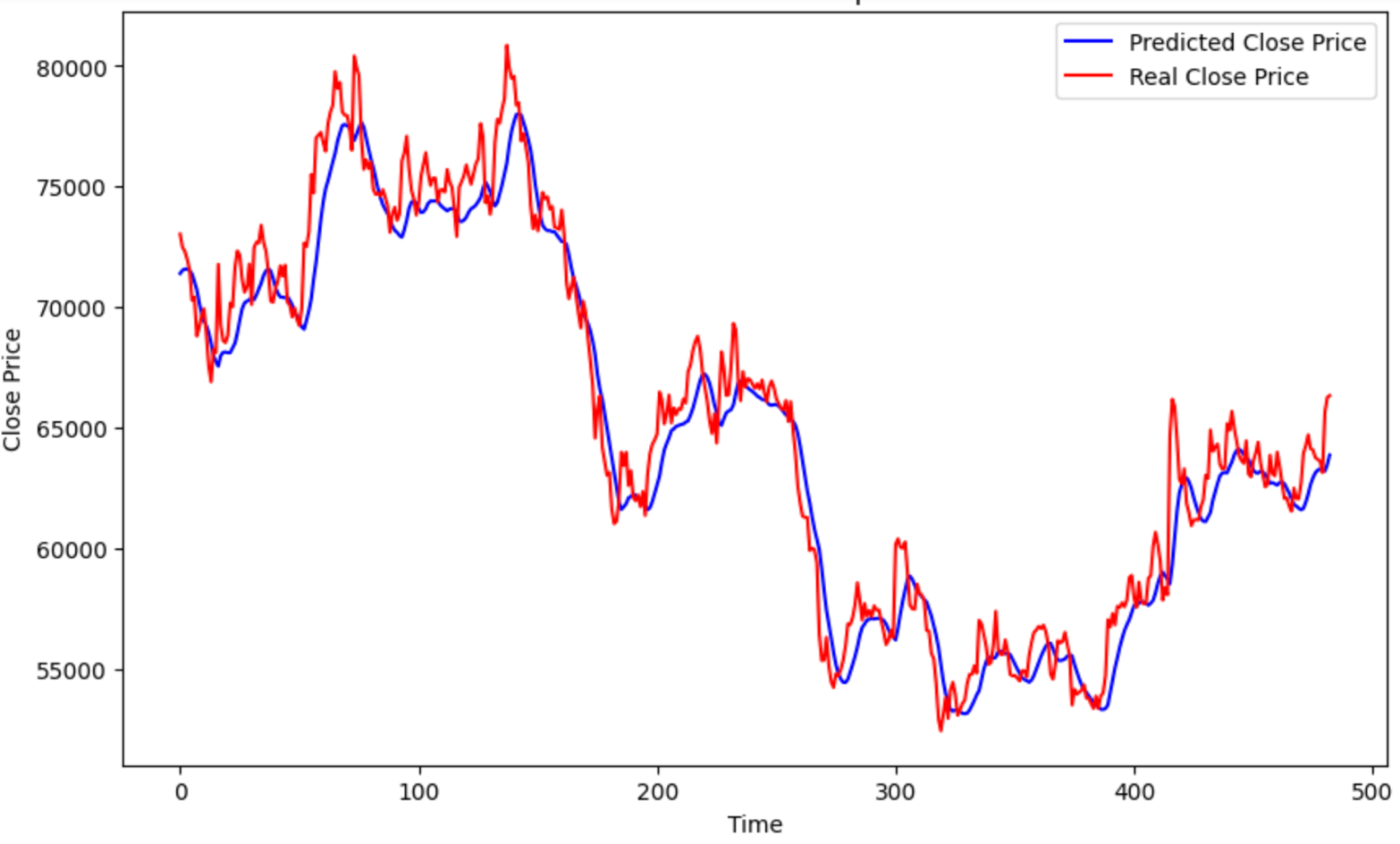
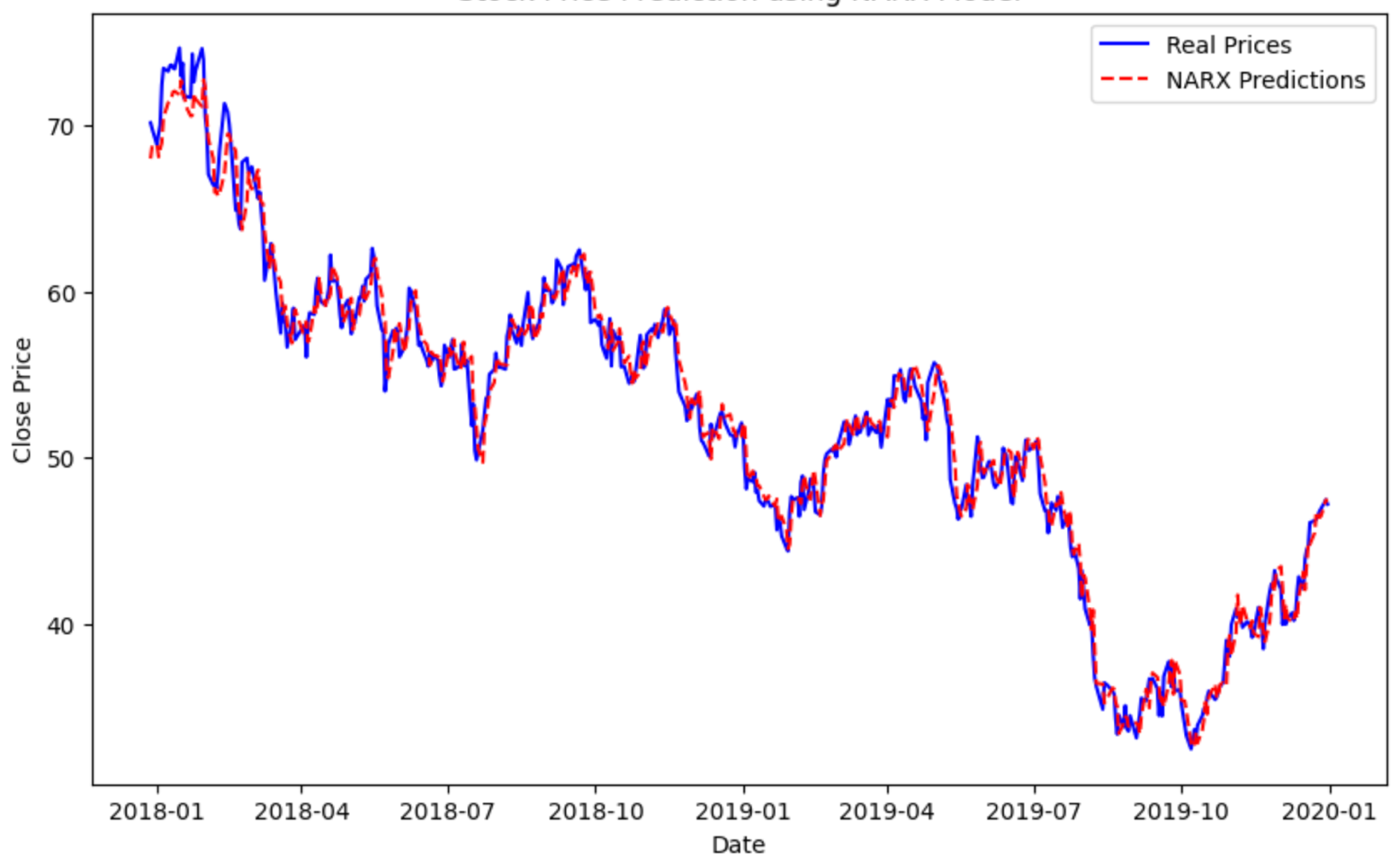


Fig.18. TATA STEEL Dataset – LSTM Results

The provided chart showcases the efforts of an LSTM model to predict Tata Steel's stock prices, with the model's forecasts drawn in blue and the actual closing prices shown in red. The blue line attempts to map out the price trajectory of Tata Steel's stock, drawing from historical data in an attempt to mirror the stock's actual performance. The model's predictions seem to rise and fall in tandem with the real movements of the stock, capturing the overarching trends of its market journey.

However, the model does not always keep pace with the stock's actual prices. There are points where the predicted path diverges from the stock's actual path, especially during sudden market changes. These instances highlight the unpredictable nature of the stock market, filled with variables that can shift outcomes in unforeseen ways. The graph illustrates that while the LSTM model offers insight into the general direction of Tata Steel's stock prices, predicting the stock market remains a complex challenge. It showcases the model's ability to provide a directional compass for stock prices but also the necessity for continual refinement to navigate the intricate dynamics of the market. The blue forecast line represents more than just predictions; it's the ongoing journey of a model striving to unravel the complex dance of stock prices within the ever-changing financial market.

**** Fig.19. TATA STEEL Dataset – NARX Results

In the graph displayed, we're given a view of how a NARX model has tackled the task of forecasting Tata Steel's stock prices. The solid blue line charts the actual closing prices, and the dashed red line is the story told by the NARX model's predictions. As we observe the lines, it's clear that the model has learned to shadow the stock's real movements to some extent. It captures the rises and falls with a fair degree of accuracy, showing that it has picked up on some key patterns in the price data. But the relationship between the model's predictions and the actual prices isn't perfect — there are points where the red line doesn't quite match up with the blue, particularly when the stock price makes sudden movements.

These differences point to the complex nature of stock price prediction. While the NARX model can get the general trend right, the intricacies of the market can sometimes throw it off track. This underscores the challenge of modelling a financial world that's full of surprises and changes that can come out of nowhere. Overall, the NARX model's effort in predicting Tata Steel's stock prices shows promise, but it also shows there's room for improvement. It serves as a reminder that even with advanced models, the unpredictable nature of the stock market can still lead to unexpected twists in the tale of stock price prediction.

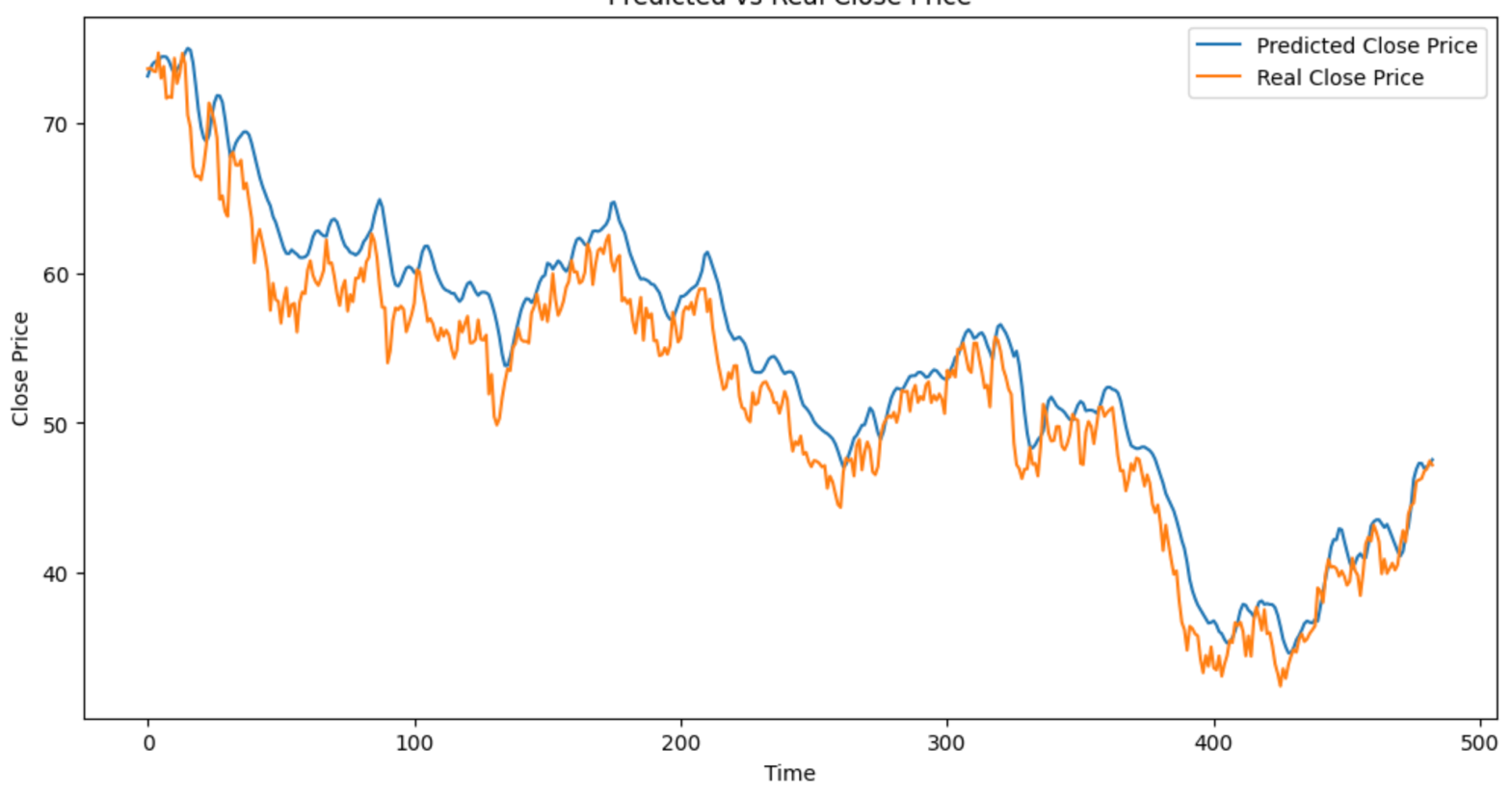
****

Fig.20. TATA STEEL Dataset – LSTM With Additional Parameters Results

This graph illustrates the efforts of an LSTM model, bolstered by additional technical indicators, to predict Tata Steel's stock prices. The real prices are traced in orange, while the model's predictions are drawn in blue.

What stands out immediately is the close following of the blue line to the orange one, indicating that the LSTM model, with its extra layers of technical analysis, is quite adept at modelling the stock's behaviour. This suggests the model has incorporated the insights provided by the VWAP, 50-Day EMA, and 200-Day EMA to closely mirror the stock's actual movements. However, the prediction isn't perfect; there are moments where the model's blue line veers away from the actual prices. These instances, particularly during more volatile periods, show that even with sophisticated tools at its disposal, the model can't foresee every twist and turn.

In essence, the LSTM model has proven itself a useful tool, offering a good approximation of where Tata Steel's stock prices might head. Yet, the unpredictability of the stock market means that even the most advanced models must contend with occasional surprises. This graph is a testament to both the potential and the limits of using technical indicators to enhance stock price predictions.

****

Fig.21.TATA STEEL Dataset –Ensemble Model of Stacked LSTM & NARX Results

This graph provides a look at how an Ensemble Model, combining Stacked LSTM and NARX, performed in forecasting Tata Steel's stock prices. The actual prices are depicted by the solid blue line, while the Ensemble Model's predictions are shown with the dashed line. From what we can see, the Ensemble Model does a commendable job of following the real stock price movements. The dashed line moves in sync with the actual prices, suggesting the model has effectively learned from historical trends. It shows the model's ability to understand and anticipate the stock's general pattern, although it's not always spot on.

There are moments when the model's predictions don't quite match up with the actual prices. These moments particularly stand out during the more volatile phases where the stock prices have sharp peaks and troughs. It's in these periods that the Ensemble Model's predictions tend to deviate, highlighting the inherent challenges in predicting the stock market with precision. Overall, the Ensemble Model's predictions for Tata Steel's stock prices paint a picture of a sophisticated approach that's largely effective but still has room for improvement, especially in handling the market's unpredictable swings.

Therefore, the performance of models based on the above graphs can be ranked as:

*Stacked LSTM – NARX Ensemble Model > NARX Model > LSTM Model > LSTM Model with Additional Parameters*

5.3 TABULAR REPRESENTATION & ANALYSIS OF RESULTS

In the quest to present a comprehensive understanding of the model's performance in forecasting stock prices, the visual illustrations of the predictions made by the LSTM model, NARX model, LSTM with additional technical indicators, and the innovative ensemble approach combining Stacked LSTM and NARX models have already been examined. This section, however, pivots towards a more granular view, encapsulating the results within the structured clarity of tables, focusing on quantitative performance metrics: Accuracy Percentage (%) and Mean Squared Error (MSE).

To adapt the concept of accuracy to the fluid nature of time series forecasting, a tailored approach is applied. Rather than the rigid classification accuracy, a more dynamic measure is employed, where predictions within a 5% margin of the actual stock prices are considered accurate. This threshold acknowledges the inherent fluctuations over time, offering a realistic gauge of the model's predictive precision.

Mean Squared Error (MSE), a staple in evaluating regression models, is used to quantify the average squared difference between the predicted and actual stock prices. This metric provides an aggregate measure of prediction error, with lower values indicating a model's heightened accuracy. However, it's crucial to note that MSE values are relative to the scale of the stock's price and the currency's value; thus, they can vary significantly across different stocks.

The ensuing tables distill these metrics, presenting Accuracy Percentage and MSE for each model applied to the stocks in focus. The tables serve not just as a numerical summary but as a narrative of each model's performance, spotlighting both their prowess and areas where refinement is needed. They stand as a testament to the analytical rigor behind the forecasting process, with numbers telling the tale of each model's journey through the intricate landscape of stock price prediction.

Note: Google & Nike are represented by U.S. Dollars while ITC and Tata Steel are represented by Indian Rupees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stocks/Models** | **LSTM** | **NARX** | **LSTM Additional Parameters** | **Stacked LSTM-NARX (Ensemble)** |
| **Google ($)** | 93.522 | 94.422 | 80.200 | 98.801 |
| **Nike ($)** | 91.902 | 88.247 | 70.890 | 98.007 |
| **ITC (₹)** | 97.308 | 99.389 | 80.700 | 99.592 |
| **Tata Steel (₹)** | 86.542 | 96.334 | 78.630 | 96.537 |

Table 1. Comparison of Accuracy % of Models with Various Stocks

The table offers a detailed look at the accuracy of four different predictive models across stocks listed in two distinct currencies: U.S. Dollars for Google and Nike, and Indian Rupees for ITC and Tata Steel. Each model's accuracy is represented as a percentage, reflecting how closely the model's predictions align with the actual stock prices.

Starting with Google, we see that the Stacked LSTM-NARX Ensemble model leads with an impressive 98.801% accuracy, suggesting a near-perfect prediction capability. The NARX model also performs strongly, with an accuracy of 94.422%, followed by the LSTM model at 93.522%. The LSTM model with additional parameters, however, shows a significant drop in accuracy, down to 80.200%.

Nike's stock also reflects a superior performance from the Ensemble model at 98.007% accuracy. Interestingly, the LSTM model outperforms the NARX model with a closer accuracy to the Ensemble model at 91.902% versus 88.247%. Here too, the LSTM model augmented with additional parameters lags behind at 70.890%.

For ITC, the Ensemble model shines with an exceptional accuracy of 99.592%, indicating its strong predictive performance with this particular stock. The NARX model follows closely at 99.389%, while the LSTM model also achieves a high accuracy rate at 97.308%. The additional parameters in the LSTM model reduce its accuracy to 80.700%.

Tata Steel's predictions showcase the NARX model's robustness with an accuracy of 96.334%. The Ensemble model performs admirably with a similar accuracy of 96.537%. The LSTM model scores lower at 86.542%, and once again, incorporating additional parameters into the LSTM model decreases its accuracy to 78.630%.

Across all stocks and models, we can discern a pattern: the Ensemble model consistently ranks highest in accuracy, indicating the combined strength of Stacked LSTM and NARX models. The NARX model alone also scores well, particularly with Indian stocks. The standard LSTM model holds its ground but is outshone by the Ensemble and NARX models. The additional technical indicators in the LSTM model, however, seem to detract from its accuracy, which could suggest overfitting or a need for better integration of these indicators.

This analysis reveals the varying efficacy of each model and underscores the Ensemble model's potential as a powerful tool for stock price prediction. It also suggests that while additional parameters can offer depth, they may not always translate to better accuracy and could require further refinement for optimal integration.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stocks/Models** | **LSTM** | **NARX** | **LSTM Additional Parameters** | **Stacked LSTM-NARX (Ensemble)** |
| **Google ($)** | 2.482 | 2.582 | 3.276 | 0.909 |
| **Nike ($)** | 6.569 | 7.404 | 36.197 | 2.693 |
| **ITC (₹)** | 35.178 | 17.262 | 33.682 | 14.260 |
| **Tata Steel (₹)** | 2.800 | 1.578 | 3.595 | 1.314 |

Table 2. Comparison of MSE (Mean Square Error) of Models with Various Stocks

The provided table lays out the Mean Squared Error (MSE) for each of four different predictive models as applied to four distinct stocks. The MSE is a measure of the average squared difference between the predicted and actual stock prices, with lower values indicating more accurate predictions.

For Google, the Stacked LSTM-NARX Ensemble model demonstrates exceptional precision with the lowest MSE of 0.909, indicating highly accurate predictions. The individual NARX and LSTM models exhibit slightly higher errors, at 2.582 and 2.482, respectively. The LSTM model with additional parameters shows a more significant error at 3.276, suggesting a decrease in predictive accuracy when these parameters are included.

Nike's stock sees a similar trend, where the Ensemble model outperforms with an MSE of 2.693. The LSTM model's error is moderately low at 6.569, but the NARX model's error increases slightly to 7.404. A substantial jump in error is noted for the LSTM model with additional parameters, which escalates to 36.197, indicating a much less accurate prediction.

For ITC, measured in Indian Rupees, the errors are generally higher across all models, reflecting the increased difficulty of predicting this stock's price movements. The NARX model performs notably well with an MSE of 17.262, followed by the Ensemble model at 14.260. The LSTM model registers a higher error at 35.178, and the addition of parameters to the LSTM model does not significantly improve the error, bringing it to 33.682.

Tata Steel's predictions reveal the best performance by the NARX model, with an impressively low MSE of 1.578. The Ensemble model also shows strong accuracy with an MSE of 1.314. The LSTM model's error is relatively low at 2.800, but again, the addition of parameters results in a slightly increased error of 3.595.

Across all stocks, the Ensemble model consistently shows a strong performance with low MSE values, indicating that the combination of Stacked LSTM and NARX models is quite effective for this task. The individual NARX model often outperforms the LSTM model, particularly for Tata Steel. The addition of technical indicators to the LSTM model does not consistently improve performance and, in Nike's case, significantly worsens it, suggesting that the integration of these indicators needs to be optimized. This analysis highlights the effectiveness of the Ensemble and NARX models in minimizing prediction error and underscores the need for careful consideration when integrating additional parameters into predictive models for stock price forecasting.

Therefore, the performance of models based on the above tables can be ranked as:

*Stacked LSTM – NARX Ensemble Model > NARX Model > LSTM Model > LSTM Model with Additional Parameters*

5.4 PREDICTION INTO THE FUTURE USING ENSEMBLE MODEL (30 DAYS)

The proposed Ensemble Model with Stacked LSTM and NARX is utilized here to forecast values in the future that were not in the dataset. The program containing the model would utilize past data to forecast values for the next 30 days after the last of the original datasets. This is tested on all four stocks of corporations, including Google, Nike, ITC, and Tata Steel. The graphical charts of these projections for the next 30 days are presented below.

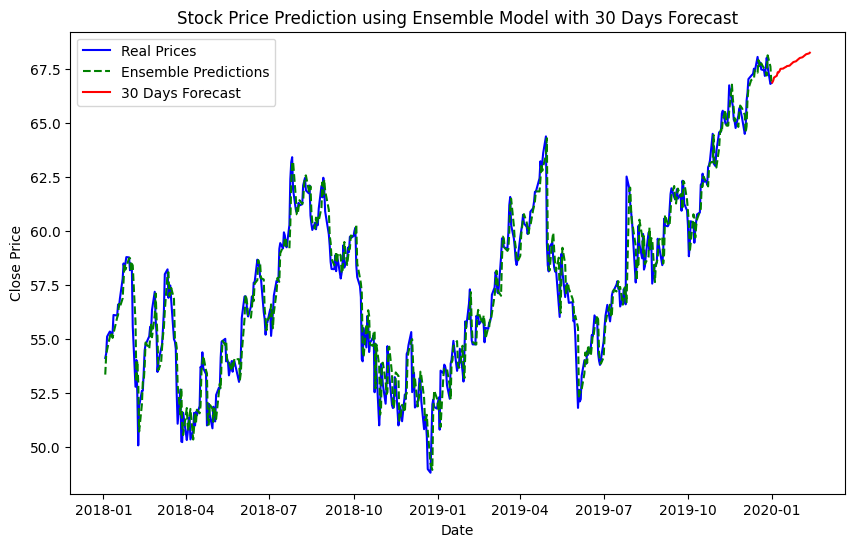


Fig.22. 30 Days Forecast Using Ensemble Model (GOOGLE Dataset)

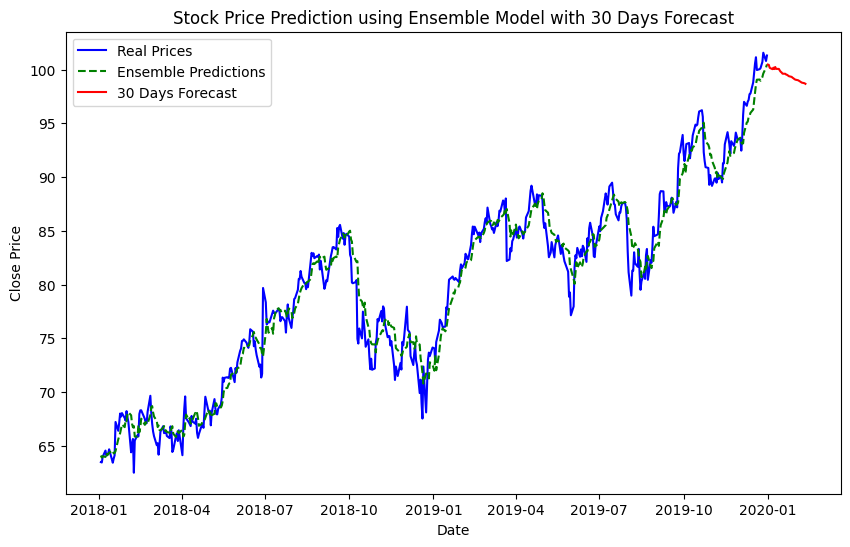


Fig.23. 30 Days Forecast Using Ensemble Model (NIKE Dataset)

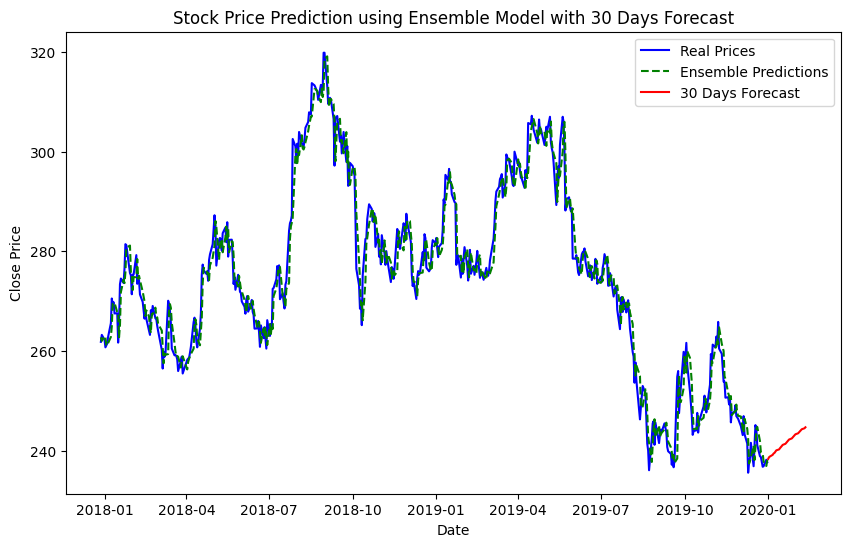


Fig.24. 30 Days Forecast Using Ensemble Model (ITC Dataset)

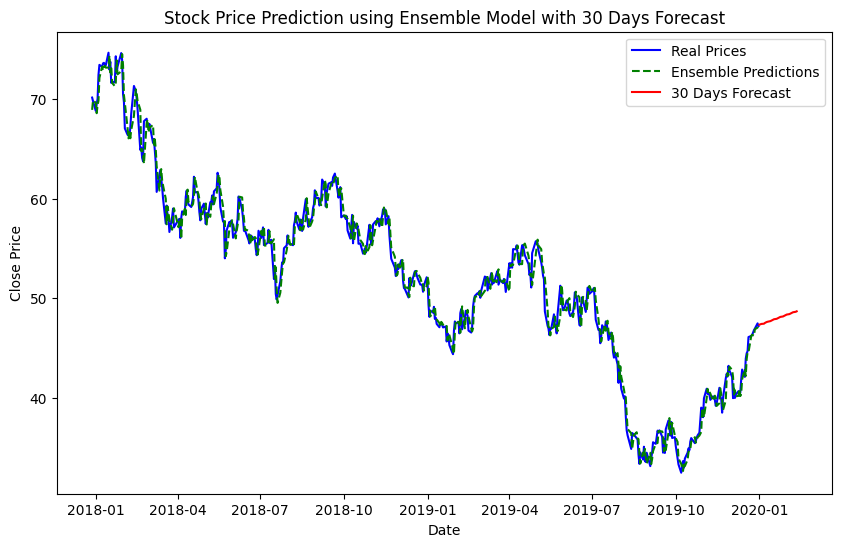
****

Fig.24. 30 Days Forecast Using Ensemble Model (TATA STEEL Dataset)

Upon evaluating the actual stock prices from the newly obtained dataset against the predicted values from the previous dataset, it was observed that there were slight deviations from the real prices, but the model successfully identified the general direction of the trends, be they rising or falling.

**Chapter 6**

**Conclusion and Future Work**

This study represents a significant stride towards enhancing the accuracy, reliability, and consistency of Deep Learning/Machine Learning models in the realm of stock price predictions across various companies, irrespective of their currency or the stock exchange they are listed on. Central to this research is the introduction of an innovative Ensemble model that synergizes Stacked LSTM & NARX, alongside a thorough investigation into the impact of key technical indicators such as VWAP, 50-Day EMA, and 200-Day EMA on stock prices.

Additionally, the paper introduces fresh perspectives by employing stocks from companies traded in USD and INR, showcasing the model's versatility across different financial markets. A noteworthy aspect of this research is the model's ability to forecast stock prices 30 days into the future, an approach aimed at testing the model's performance under conditions of uncertainty not covered in the training dataset.

The Ensemble model, combining Stacked LSTM & NARX, demonstrated exceptional capability in predicting stock prices both within the test set and in the 30-day future forecasting scenario. It notably surpassed the performance of its individual components (existing models) and models that incorporated additional technical indicators like VWAP, 50 Day EMA, and 200 Day EMA, as evidenced in the Results section.

Looking ahead, the paper suggests avenues for future research to further refine the model's predictive accuracy and consistency. One promising direction involves the integration of Sentiment Analysis using Natural Language Processing (NLP) tools. This approach would allow for the incorporation of real-time data into the model, potentially enhancing the precision of predictions over short intervals and bolstering the model's consistency, reliability, and accuracy in forecasting stock prices.

**Appendices**

1. Detailed Architecture of Popular Deep Learning Models

* Long Short-Term Memory (LSTM) Networks: Explanation of LSTM units, gates (input, output, forget), and how they are specifically adapted to capture long-term dependencies in time series data.
* Gated Recurrent Units (GRU) Networks: Description of GRU architecture, comparison with LSTMs, and their efficiency in learning from time series data.
* Convolutional Neural Networks (CNN) for Time Series: Adaptation of CNNs for time series analysis, including the use of 1D convolutions to capture temporal dependencies.
* Transformer Models: Insight into the transformer architecture, self-attention mechanism, and its application in time series forecasting, highlighting any adaptations made for the stock price prediction.

1. Standard Data Pre-processing Techniques

* Feature Engineering: Overview of features commonly used for stock price prediction (e.g., moving averages, RSI, MACD) and their significance.
* Normalization and Standardization: Importance of data scaling methods in deep learning and how they are applied to stock price data.
* Sequence Creation: Methodology for transforming time series data into a format suitable for deep learning models, including windowing techniques.

1. Standard Hyperparameter Tuning and Model Optimization

* Hyperparameter Selection: Discussion on the selection of crucial hyperparameters (e.g., learning rate, batch size, number of layers) and their impact on model performance.
* Regularization Techniques: Use of dropout, L1/L2 regularization to prevent overfitting in deep learning models.
* Optimization Algorithms: Comparison of different optimizers (e.g., Adam, RMSprop) and their effectiveness in training deep learning models for stock prediction.

1. Popular Evaluation Metrics for Model Performance

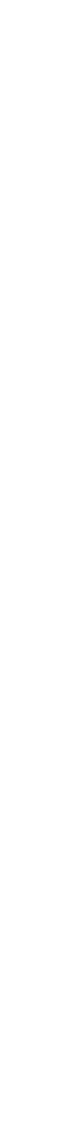
* Root Mean Square Error (RMSE): Importance of RMSE in evaluating the accuracy of stock price forecasts.
* Mean Absolute Error (MAE): Benefits of using MAE as a more robust measure against outliers.
* Mean Absolute Percentage Error (MAPE): Using MAPE to understand model performance in percentage terms, making it easier to interpret in financial contexts.

1. Case Studies and Applications

* Application in Algorithmic Trading: A notable case study involves the use of LSTM networks to predict stock prices for high-frequency trading. Researchers trained models on historical price data, incorporating features like price changes, volume changes, and technical indicators. The LSTM model outperformed traditional time series models, providing actionable insights for real-time trading algorithms.
* Sector-specific Predictions: GRU networks have been applied to predict the stock prices of companies within specific sectors, such as technology or energy. This approach leverages sector-specific dynamics and correlations between companies to enhance prediction accuracy. For instance, a model trained on tech stocks might incorporate data from market sentiment analysis, tech industry news, and global economic indicators, leading to improved forecasting performance.
* Enhancing Risk Management: Convolutional Neural Networks (CNN) have been adapted for time series forecasting to analyze patterns across multiple financial markets simultaneously. This holistic view enables traders and risk managers to better understand market correlations and volatility, thereby enhancing risk assessment and management strategies.
* Portfolio Optimization: Transformer models, known for their effectiveness in capturing long-range dependencies, have been used for predicting the future performance of a portfolio of stocks. By understanding the complex interplay between different assets, these models assist in constructing optimized portfolios that maximize returns while minimizing risk.

1. Challenges and Future Directions

* Market Volatility: The inherent volatility of stock markets, influenced by economic, political, and social factors, poses a significant challenge to predictive models. Future research might focus on integrating real-time news analysis and sentiment analysis to improve models' responsiveness to unforeseen events.
* Data Quality and Availability: High-quality, granular data is crucial for training accurate models. However, access to detailed financial datasets can be limited or expensive. Future advancements may rely on novel data sourcing strategies, such as leveraging alternative data sources (social media sentiment, satellite imagery of retail parking lots, etc.) to gain additional insights into market movements.
* Model Interpretability and Explainability: As deep learning models become more complex, their decisions become harder to interpret. This is particularly problematic in finance, where stakeholders require clear explanations for trading decisions. Future work could focus on developing more interpretable models or methods that provide insight into how models make predictions.
* Adaptation to Emerging Markets: Emerging markets behave differently from more established markets, presenting unique challenges and opportunities for predictive models. Research into models that can adapt to the less predictable nature of these markets, perhaps by incorporating macroeconomic indicators and local market sentiment, is a promising future direction.



**REFERENCES**

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

[2]. Dinesh, S., Raju, A. R., Rahul, S., Sandeep, O. N., & Relangi, M. N. K. STOCK PRICE PREDICTION USING LSTM.

[3]. Zhang, R. (2022, March). LSTM-based Stock Prediction Modeling and Analysis. In *2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022)* (pp. 2537-2542). Atlantis Press.

[4]. Dhafer, A. H., Nor, F. M., Alkawsi, G., Al-Othmani, A. Z., Shah, N. R., Alshanbari, H. M., ... & Baashar, Y. (2022). Empirical analysis for stock price prediction using NARX model with exogenous technical indicators. *Computational Intelligence and Neuroscience*, *2022*.

[5]. Alkhoshi, E., & Belkasim, S. (2018, September). Stable stock market prediction using NARX algorithm. In *Proceedings of the 2018 International Conference on Computing and Big Data* (pp. 62-66).

[6]. Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of big Data*, *7*(1), 1-33.

[7]. Yetis, Y., Kaplan, H., & Jamshidi, M. (2014, August). Stock market prediction by using artificial neural network. In *2014 world automation congress (WAC)* (pp. 718-722). IEEE.

[8]. Zhang, K., Zhong, G., Dong, J., Wang, S., & Wang, Y. (2019). Stock market prediction based on generative adversarial network. *Procedia computer science*, *147*, 400-406.

[9]. Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2020). An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, *76*, 2098-2118. *Neuroscience*, *2022*.

[10]. Maiti, A. (2020, November). Indian stock market prediction using deep learning. In *2020 IEEE REGION 10 CONFERENCE (TENCON)* (pp. 1215-1220). IEEE.

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

[12]. Sonkiya, P., Bajpai, V., & Bansal, A. (2021). Stock price prediction using BERT and GAN. *arXiv preprint arXiv:2107.09055*.

[13]. KALYONCU, Ş., Jamil, A., Karataş, E., Rasheed, J., & Djeddi, C. (2020). Stock market value prediction using deep learning. *International Journal of Data Science and Applications*, *3*(2), 10-14.

[14]. Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia computer science*, *167*, 599-606.

[15]. Lanbouri, Z., & Achchab, S. (2020). Stock market prediction on high frequency data using long-short term memory. *Procedia Computer Science*, *175*, 603-608.