STOCK MARKET PREDICTION: USING ARIMA, LSTM & TCN

A Project Report submitted in partial fulfillment of the requirements for the awardof the degree of

BACHELOR OF

TECHNOLOGYIN

COMPUTER SCIENCE AND BUSINESS SYSTEMS

Submitted by

HU22CSEN0200021	**Sujay Naidu Dhulipudi**
HU22CSEN0200189	**Divyanshu Mahi**
HU22CSEN0200034	**Sarath Chandra P.V**

Under the esteemed guidance of

Mr. ARUN K
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING GITAM

(Deemed to be University)

HYDERABAD

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGGITAM SCHOOL OF TECHNOLOGY GITAM

(Deemed to be University)

DECLARATION

I/We, hereby declare that the project report entitled "**STOCK MARKET PREDICTION USING ARIMA, LSTM & TCN**" is an original work done in the Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Business Systems. The work has not been submitted to any othercollege or University for the award of any degree or diploma.

Date: 29-10-2024

Registration No(s).	**Name(s)**
HU22CSEN0200021	Sujay Naidu Dhulipudi
HU22CSEN0200189	Divyanshu Mahi
HU22CSEN0200034	Sarath Chandra P.V

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGGITAM SCHOOL OF TECHNOLOGY GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled “STOCK MARKET PREDICTION USING ARIMA, LSTM & TCN” is a Bonafide record of work carried out by students **Sujay Naidu Dhulipudi** (HU22CSEN0200021), **Divyanshu Mahi** (HU22CSEN0200189), **Sarath Chandra P.V** (HU22CSEN0200034) submitted in partial fulfillment of requirementfor the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide	**Project Coordinator**	**Head of the**
Department		
Mr. Arun K	**Dr. A B Pradeep Kumar**	**Mr. Shaik Mahaboob Basha**
Assistant ProfessorDept. of CSE	**Assistant ProfessorDept. of CSE**	**Professor & HOD Dept. of CSE**

ACKNOWLEDGEMENT

Our project report would not have been successful without the help of several people. We would like to thank the personalities who were part of our seminar in numerous ways, those who gave us outstandingsupport from the birth of the seminar.

We are extremely thankful to our honorable Pro-Vice-Chancellor, **Prof. D. Sambasiva Rao**, for providing the necessary infrastructure and resources for the accomplishment of our seminar. We are highly indebted to **Prof. N. Seetharamaiah**, Associate Director, School of Technology, for his support during the tenure of the seminar.

We are very much obliged to our beloved **Prof. Shaik Mahaboob Basha**, Head of the Department of Computer Science & Engineering, for providing the opportunity to undertake this seminar and encouragement in the completion of this seminar.

We hereby wish to express our deep sense of gratitude to **Dr. A B Pradeep Kumar**, Project Coordinator, Department of Computer Science and Engineering, School of Technology and to our guide, **Mr. Arun K**, Assistant Professor, Department of Computer Science and Engineering, School of Technology for the esteemed guidance, moral support and invaluable advice provided by them for the success of the project report.

We are also thankful to all the Computer Science and Engineering department staff members who have cooperated in making our seminar a success. We would like to thank all our parents and friends who extended their help, encouragement, and moral support directly or indirectly in our seminar work.

Sincerely,

Sujay, Divyanshu, Sarath

TABLE OF CONTENTS

1. Introduction
    1.1 Introduction
    1.2 Problem Statment
    1.3 Aim and Objective
    1.4 Scope of Study
    1.5 Significance of Study

2. Literature Review
    2.1 Introduction
        2.1.1 ARIMA
        2.1.2 LSTM
        2.1.3 TCN
        1.2.4 Evaluation Metrics
            1.MSE & RMSE
            2.R-squared

3. Problem Analysis
    3.1 Problem Statment
    3.2 Flaws and Disadvantages
        3.2.1. ARIMA
        3.2.2 LSTM
        3.2.3 TCN
    3.3 Proposed System
    3.4 Functional Requirements
    3.5 Non-functional Requirements

4. Implementation
    4.1 Overview of Technologies
    4.2 Libraries Imported
    4.3 Dataset

5. Regression
    5.1 ARIMA
        5.1.1 MAPE
        5.1.2 MAE
        5.1.3 RMSE
        5.1.4 Analysis
        5.1.5 Conclusion
    5.2 TCN

5.2.1 MAPE
5.2.2 MAE
5.2.3 RMSE
5.2.4 Analysis
5.2.5 Conclusion
5.3 LSTM
5.3.1 MAPE
5.3.2 MAE
5.3.3 RMSE
5.3.4 Analysis
5.3.5 Conclusion

6. Recommendations
6.1 Incorporation of additional features
6.2 Use of Dynamic or Adaptive Models
6.3 Error Monitoring and Feedback Loop
6.4 Final Remarks

ABSTRACT

Predicting stock market trends has been an area of interest and research for many years due toits significant influence on global economies, businesses, and individual investors. This studyaims to perform a comprehensive comparative analysis of different machine learning techniques to identify the most effective models for stock market prediction in today's data- driven financial landscape. By doing so, the research intends to aid individual and institutional investors in making more informed decisions while also contributing valuable insights to the fields of finance and machine learning.

This research specifically focuses on evaluating the effectiveness of Temporal Convolutional Networks (TCN) based on its principles, ARIMA, and LSTM models in accurately forecasting stock market movements. It uses a detailed dataset from Kaggle, which includesprice and volume data.

The methodology involves thorough data preprocessing steps such as data cleaning, stationarity checks, normalization, and feature engineering to prepare the dataset for analysis.Each model will be carefully implemented, tuned, and evaluated using the Mean Absolute Error (MAE) metric, which will help determine each model's strengths and weaknesses in thecontext of stock market prediction.

The results of this study are anticipated to lead to the development of more robust, accurate,and efficient predictive models, which could have a substantial impact on financial analyticsand investment strategies. The findings will provide practical insights for investment and riskmanagement, while also advancing predictive analytics capabilities within the finance sector.

CHAPTER 1

1.1 Introduction

In recent years, public interest in the stock market has surged, with billions of dollars traded daily as investors aim to optimize returns. Accurate stock market predictions have become essential in finance, offering critical insights to individual investors, financial institutions, and other market participants. Predictive tools help investors make informed buy, sell, or hold decisions by forecasting market trends, enabling them to improve returns and minimize risks, especially in volatile markets.

For institutional investors like mutual funds, pension funds, and insurance companies, accurate forecasts support effective asset allocation and diversification, mitigating systemic risks in large portfolios. Financial institutions, including banks and brokerages, leverage predictions to refine asset management strategies, tailor investment advice, and optimize financial offerings. Predictive analytics also help these firms align operational strategies with market expectations, safeguarding liquidity and stability.

Regulatory bodies benefit from predictive insights as well, using them to monitor market dynamics, detect irregularities, and address systemic risks, thereby promoting market integrity and resilience. Traditionally, stock analysis relied on fundamental and technical methods, but the rise of advanced computational tools and large datasets has transformed thelandscape, with quantitative methods now at the forefront.

Overall, accurate stock market predictions serve a diverse range of stakeholders, guiding strategic decisions, enhancing risk management, and fostering the overall efficiency and stability of financial markets.

1.2 Problem Statement

Accurate stock market prediction using machine learning (ML) models faces numerous challenges due to the market’s complexity and volatility. A key issue is identifying the most effective predictive model, as prior research often focuses on individual or limited methods,leaving a gap in comparative analyses of diverse ML techniques. This study addresses this gap through a broad evaluation of various models.

Efficient preprocessing and handling of high-dimensional financial data remain significant hurdles, with previous studies often overlooking how nuanced preprocessing affects model performance. Additionally, many ML algorithms struggle to adapt to sudden market shifts, such as downturns or political events, limiting their robustness in real-world scenarios.

Evaluation metrics also frequently fall short, failing to fully capture the practical application of these models in trading. This research seeks to develop metrics that better align with real- world investment and trading needs.

1.3 Aim and Objectives

This research primarily aims to perform an in-depth comparative analysis of various machine learning models and techniques for stock market prediction, addressing the gaps in current studies. It seeks to improve understanding of how different ML models perform in the dynamic and complex financial market environment, ultimately contributing to more reliable and effective prediction strategies.

Objectives:

- Conduct a comprehensive literature review on stock market prediction, highlighting key influencing factors.
- Identify optimal data preprocessing techniques to manage the complexity and diversity of financial data.
- Systematically evaluate a range of machine learning models, including ARIMA (AutoRegressive Integrated Moving Average), LSTM (Long Short-Term Memory), and TCN(Temporal Convolutional Network) techniques.
- Compare the performance of various machine learning models and techniques used in stockmarket prediction.

1.4 Scope of Study

The scope of this research is defined as follows:

- This study concentrates on exploring the complexities of stock market prediction usingmachine learning models and techniques.
- Feature engineering will be limited to the features already present in the dataset.
- The research will specifically evaluate and compare the effectiveness of various machine learning algorithms for stock market prediction.

1.5 Significance of Study

The significance of this research lies in advancing existing financial forecasting methods. Accurate stock market predictions can lead to considerable economic gains. The insights derived from this study aim to support investors, traders, and financial analysts in crafting more effective investment strategies. Enhanced predictive models can improve decision- making, strengthen risk management, and increase potential investment returns. Additionally, this research identifies key challenges, limitations, and areas for improvement, laying a foundation for future advancements in stock market prediction.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Stock market prediction has long captivated both researchers and practitioners, driven by its potential for substantial financial rewards. The journey to accurately forecast stock prices has progressed from basic chart analysis to advanced machine learning algorithms. Initial effortsrelied heavily on fundamental and technical analysis, with foundational contributions from Graham and Dodd (Singh and Kaur, 2014) that explored market dynamics through company performance metrics. Recently, advancements in computational power and data availability have facilitated the use of complex models, like deep learning, to uncover intricate, multi- frequency trading patterns (Zhang et al., 2017).

ARIMA

ARIMA stands as a cornerstone in the time-series forecasting domain, celebrated for its flexibility and robustness in modelling a wide range of univariate time series. The choice of ARIMA for this study is grounded in its methodological simplicity and effectiveness in capturing linear relationships and trends in stationary data.

By integrating autoregression (AR), differencing (I), and moving averages (MA), ARIMA provides a comprehensive framework for forecasting stock prices, offering valuable insights into the linear aspects of financial time series.

Strong Theoretical Foundation: ARIMA's methodology is well-established in time series forecasting, offering a robust statistical framework for analyzing and predicting linear time series data. Its widespread acceptance and application across different domains provide a solid foundation for its use in stock market prediction.

Simplicity and Interpretability: Compared to more complex models, ARIMA is relatively straightforward, making it easier to implement and interpret. This simplicity allows for clear understanding and communication of how predictions are generated.

Effectiveness in Short-Term Forecasting: ARIMA is particularly noted for its effectiveness in short-term forecasting, where the underlying time series data exhibits strong autocorrelation. This makes it suitable for stock market prediction, where short-term trends and patterns can be critical.

LSTM

LSTMs are a type of recurrent neural network (RNN) designed to overcome the limitations of traditional RNNs in learning long-term dependencies. The inclusion of LSTM in this study ismotivated by its proven efficacy in handling sequential data, making it an ideal candidate for modelling time-series financial data where past information is crucial for predicting future stock market movements. The LSTM's ability to remember and forget information selectivelyallows for a nuanced understanding of temporal patterns, a feature indispensable in the volatile realm of stock market prediction.

This study exemplified LSTM's capability to capture complex, long-term dependencies in highly volatile forex markets, analogous to stock markets. The successful application of LSTMs in this context demonstrates their potential for modelling non-linear financial time series, further motivating their inclusion in this research.

Ability to Capture Long-Term Dependencies: LSTMs are designed to overcome the limitations of traditional recurrent neural networks (RNNs) in learning long-term dependencies. This characteristic is crucial in stock market prediction, where long-term historical data can influence future stock prices.

Flexibility in Modeling Non-linear Patterns: LSTM's architecture allows it to model the non- linear relationships of stock market effectively, providing a more accurate prediction in complex market conditions. The success of RNNs in Kondratenko and Kuperin's study in navigating such complexities effectively underscores the potential of LSTM networks in this study to deal with the erratic behavior of stock prices and indices, given LSTMs' advanced memory and processing capabilities.

Proven Success in Sequence Prediction Tasks: LSTMs have demonstrated success in various sequence prediction tasks beyond finance, such as natural language processing and sequence generation. This track record suggests their potential applicability and effectiveness in predicting stock market trends.

Benchmarking Against Other Models: Kondratenko and Kuperin's comparative analysis of RNN performance against other models offered a methodological blueprint for evaluating LSTM in the context of stock market prediction. It highlighted the importance of benchmarking LSTM's forecasting performance against traditional and other machine learning models to substantiate its selection based on empirical evidence.

TCN

Temporal Convolutional Networks (TCNs) are a type of neural network architecture well-suited for modeling sequence data. They differ from recurrent models by utilizing convolutional layers to capture dependencies over time, allowing for efficient parallel processing. The inclusion of TCN in this study is motivated by its proven effectiveness in managing sequential data without the pitfalls of recurrent connections, making it highly relevant for stock market prediction, where temporal dependencies are crucial. TCNs leverage dilated convolutions to capture long-range dependencies, offering an alternative approach to traditional RNNs for analyzing complex time-series patterns

Ability to Capture Long-Term Dependencies: Unlike traditional RNNs, TCNs use dilated convolutions, enabling them to capture long-term dependencies across the sequence while maintaining computational efficiency. This characteristic is particularly advantageous for stock market prediction, where past trends and patterns can have lasting impacts on future price movements.

Flexibility in Modeling Non-Linear Patterns: TCN's structure, which combines causal convolutions and dilations, allows it to model complex, non-linear relationships effectively, which are often present in stock market data. By avoiding recurrence, TCNs can process datain parallel, leading to faster training and inference times, while still capturing nuanced temporal patterns critical for forecasting in dynamic markets.

Proven Success in Sequence Prediction Tasks: TCNs have demonstrated robust performance across various sequence prediction tasks beyond finance, including applications in audio generation, language modeling, and other domains requiring sequential data analysis. Their effectiveness in these areas highlights their suitability for modeling financial time series, where accurately capturing sequential dependencies is essential for reliable predictions.

Benchmarking Against Other Models: Comparative studies of TCNs against traditional RNN-based models, such as LSTMs, have shown that TCNs can outperform recurrent models in certain time-series tasks due to their unique convolutional approach. In this research, TCNs will be evaluated alongside LSTM, ARIMA, and other models to determine their effectiveness in stock market prediction. This benchmarking will provide empirical evidence for TCN's selection, assessing its performance relative to established machine learning techniques in financial forecasting.

The inclusion of TCN in this study aims to explore its potential as a powerful alternative to LSTMs for stock market prediction, leveraging its strengths in handling temporal data with fewer computational demands, while providing insights into its comparative strengths and limitations.

Evaluation Metrics

In the dynamic and complex domain of stock market prediction, the choice of evaluation metrics plays a pivotal role in assessing the performance and reliability of predictive models. This section aims to shed light on the diverse array of metrics utilized across various studies to gauge the accuracy, robustness, and practical applicability of machine learning and deep learning models in forecasting market movements. This critical examination not only highlights the strengths and limitations of different metrics but also underscores the importance of selecting appropriate measures that align with specific research objectives and market dynamics.

MSE AND RMSE

Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are key metrics for assessing the accuracy of stock market prediction models. MSE measures the average squared difference between predicted and actual values, while RMSE, as the square root of MSE, offers error values in the same units as the original data, improving interpretability. These metrics are valuable in highlighting significant prediction deviations, as squaring errors in MSE penalizes larger discrepancies more heavily.

Gooijer and Hyndman (2006) reviewed forecasting advancements, including ARIMA and GARCH models, noting MSE and RMSE's usefulness in model comparisons across datasets. They pointed out that while these metrics effectively indicate model inadequacy in scenarios where large errors are detrimental, they may over-penalize models where occasional large errors are acceptable. Patel et al. (2015) applied MSE and RMSE to evaluate machine learning models like Decision Trees and k-NN for stock prediction, facilitating model selection. Fischer and Krauss (2018) used RMSE to demonstrate LSTM's superior accuracy over traditional models in capturing complex stock price patterns, reinforcing the potential of LSTM for reliable financial forecasting.

R-Squared

R-squared (R^2), or the coefficient of determination, measures how much of the variance in a dependent variable is explained by independent variables in a regression model. In stock market prediction, R^2 assesses a model's explanatory power, indicating how well factors like historical prices and technical indicators account for future stock price movements. A higher R^2 suggests a better fit, but it doesn't guarantee accurate predictions, particularly in volatile markets.

Fama and French (1992) utilized R^2 in their three-factor model of stock returns, demonstrating that incorporating market risk, size, and book-to-market value significantly improved explanatory power compared to the Capital Asset Pricing Model (CAPM). Their findings showed that the size (SMB) and value (HML) factors were crucial in explaining stock return variability.

Similarly, Kara et al. (2011) used R^2 to evaluate hybrid models combining artificial neural networks and support vector machines in predicting stock index movements, quantifying how much market movement could be explained by their models.

Overview of Chosen Evaluation Metric

Accurately assessing forecasting models is as vital as their development. This discussion focuses on the choice of Mean Absolute Error (MAE) as the primary metric for evaluating predictive accuracy, emphasizing its significance in financial forecasting.

MAE is crucial for evaluating models like ARIMA, LSTM, and Full TCN in stock marketpredictions for several reasons:

1. **Direct Interpretability**: MAE provides a clear measure of average prediction error in the same units as the predicted variable, making it easy for analysts and investors to understand errors in stock prices or returns.

2. **Robustness to Outliers**: Unlike MSE or RMSE, MAE is less sensitive to outliers, ensuring that sudden market fluctuations do not disproportionately skew the error metric, resulting in a more reliable measure of model performance.

3. **Linear Error Accumulation:** Each error contributes equally to MAE, facilitating fair comparisons among models like ARIMA (good for linear trends), LSTM (captures long-term dependencies), and Full TCN (effective for complex patterns).

4. **Broad Applicability**: MAE can be consistently applied across different models, regardless of complexity, making it an ideal metric for comparing diverse approaches to stock market prediction.

In the context of stock forecasting, where accuracy is crucial, MAE offers a balanced and practical measure for evaluating and comparing models like ARIMA, LSTM, and Full TCN, reflecting both technical accuracy and practical relevance for analysts and investors.

Research Gap

While significant progress has been made in using models like ARIMA for linear time series forecasting and LSTM networks for capturing long-term dependencies in sequential data, the integration and comparative analysis of these methods with more recent deep learning advancements, such as Full Temporal Convolutional Networks (TCNs) ,remain underexplored.

Additionally, employing Mean Absolute Error (MAE) as the sole metric for evaluating and comparing these diverse models adds complexity. The literature highlights a gap in systematically applying and discussing MAE as a benchmark for performance across various modeling techniques, especially in the nuanced field of financial forecasting, where accuracy and reliability are crucial.

CHAPTER 3

PROBLEM ANALYSIS

3.1 Problem Statement

Accurate stock market prediction using machine learning (ML) models faces numerous challenges due to the market's complexity and volatility. A key issue is identifying the most effective predictive model, as prior research often focuses on individual or limited methods,leaving a gap in comparative analyses of diverse ML techniques. This study addresses this gap through a broad evaluation of various models.

Efficient preprocessing and handling of high-dimensional financial data remain significant hurdles, with previous studies often overlooking how nuanced preprocessing affects model performance. Additionally, many ML algorithms struggle to adapt to sudden market shifts, such as downturns or political events, limiting their robustness in real-world scenarios.
Evaluation metrics also frequently fall short, failing to fully capture the practical applicationof these models in trading. This research seeks to develop metrics that better align with real- world investment and trading needs.

3.2 Flaws & Disadvantages

1. ARIMA (AutoRegressive Integrated Moving Average):

- **Limited to Linear Relationships:** ARIMA models are effective for time series withlinear patterns but struggle with non-linear data, which is common in stock market predictions.

- **Requires Stationarity:** ARIMA models work best on stationary data, so preprocessing steps like differencing are often necessary, making them less adaptable to complex, real- world data.

- **Lacks Flexibility for Long-Term Dependencies:** ARIMA's structure doesn't support capturing long-term dependencies well, which limits its effectiveness in scenarios requiring deeper historical context.

2. LSTM (Long Short-Term Memory):

- **Computational Complexity:** LSTMs require significant computational power and memory, especially for large datasets, leading to long training times.

- **Sensitive to Hyperparameters:** LSTMs have many hyperparameters, making them sensitive to tuning. Poorly optimized hyperparameters can result in overfitting or underfitting.

- **Difficulty with Very Long Sequences:** Although LSTM is designed for sequential data, very long sequences can still challenge its memory cell mechanism, potentially leading to degradation in prediction accuracy.

3. TCN (Temporal Convolutional Network):

- **High Memory Usage:** TCNs can require substantial memory resources, particularly forhigh-dimensional and long-time series data, which can hinder scalability.

- **Limited Interpretability:** TCNs can be harder to interpret compared to traditionalstatistical models, as the convolutional layers make it challenging to understand which historical values influence predictions.

- **Hyperparameter Sensitivity:** TCN models can be sensitive to the choice of kernel size and dilation factors, and improper tuning may lead to suboptimal performance.

3.3 Proposed System

The proposed system aims to leverage the unique strengths of ARIMA, LSTM, and TCN models to achieve a comprehensive approach to stock market price prediction. By combiningthese models, the system addresses various aspects of financial time series data, enhancing prediction accuracy and robustness.

1. Model Selection and Hybridization:

- **ARIMA** will be used to capture linear patterns and trends in stationary data, effectivelymodeling basic historical price movements.

- **LSTM** will handle sequential dependencies and long-term relationships in the data, addressing the non-linear dynamics and providing a deep understanding of temporal patterns.

- **TCN** will capture complex, high-frequency patterns, utilizing its convolutional structureto detect local dependencies in the stock market time series.

2. Data Preprocessing Pipeline:

- Financial data often contains noise and outliers; hence, preprocessing steps such as,feature extraction, and differencing (for ARIMA) will be implemented.

- Data will be segmented into training and testing sets, ensuring that all models receiveproperly formatted time-series data suited to their structure.

3. Model Training and Evaluation:

- Each model will be trained separately on historical stock market data, using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) as evaluation metrics.

- The models' outputs will be combined in a hybrid or ensemble method, where eachmodel's prediction is weighted based on its accuracy during testing.

4. Implementation of Evaluation Metrics:

- MSE and RMSE will be used to assess the accuracy of each model, allowing for aquantitative comparison and validation of model performance.

- Additional metrics, such as Mean Absolute Error (MAE), may be introduced to evaluatethe robustness of the models and detect any bias in predictions.

5. System Integration for Real-Time Prediction:

- The system will be configured for real-time data input, enabling live predictions. Continuous retraining with recent data will ensure the model adapts to changing marketconditions.

This proposed system seeks to leverage ARIMA, LSTM, and TCN in a complementary fashion, capitalizing on their individual strengths while offsetting their weaknesses. This hybrid approach aims to provide a more resilient and accurate stock price prediction solution, ultimately benefiting financial analysts and investors.

3.4 Functional Requirements

1. **Data Collection and Preprocessing**:

   ○ The system should automatically retrieve and update financial time series data(e.g., stock prices, trading volume) from reliable data sources.
   ○ It must preprocess the data by handling missing values, normalizing the data,and preparing it in a format compatible with ARIMA, LSTM, and TCN models.
   ○ Feature extraction and selection techniques should be applied to highlightessential data patterns and reduce dimensionality where necessary.

2. **Model Training**:

   ○ The system should allow separate training of ARIMA, LSTM, and TCNmodels on historical data.
   ○ Each model's hyperparameters should be tuneable, allowing experimentationand optimization to improve prediction accuracy.
   ○ The system must enable iterative model updates with new data, ensuring themodels adapt to the latest market trends.
   ○

3. **Prediction Generation**:

   ○ The system should generate stock price predictions based on individual models (ARIMA, LSTM, and TCN) and output an ensemble or weightedprediction.
   ○ It must offer short-term and medium-term predictions to support varioustrading strategies.

4. **Performance Evaluation**:

    - The system should compute performance metrics like MSE, RMSE, and MAE for each model to evaluate their prediction accuracy.
    - It should display comparison reports that help users understand which modelis most effective under different conditions.

5. **User Interface**:

    - A dashboard should be provided to allow users to view historical data, modelpredictions, and evaluation metrics.
    - The system must offer options for users to adjust model parameters, view individual model performance, and choose ensemble weights for predictions.

6. **Real-Time Prediction Updates**:

    - The system should support real-time data feeds and provide live predictions asmarket conditions change.
    - Predictions should refresh periodically or on-demand, allowing users to accessthe latest insights for timely decision-making.

3.5 Non-Functional Requirements

- **Performance**:

    - The system should ensure low latency in data processing and prediction generation toprovide near real-time stock price forecasts.
    - Model training and prediction processes should be optimized to minimize resourceconsumption without compromising accuracy.

- **Scalability**:

    - The system should be able to handle a large volume of data, accommodating historicaland real-time stock data for multiple stocks or indices.
    - It should scale horizontally if needed, allowing for the integration of additional machine learning models or handling of more complex

datasets.

- **Reliability**:

  - The system must consistently provide accurate predictions, with mechanisms in place to monitor prediction accuracy and alert users if model performance degrades.
  - It should have error-handling protocols for data retrieval failures or unexpected modeloutputs.

- **Usability**:

  - The user interface should be intuitive, enabling financial analysts and investors to use the system with minimal training.
  - Model results, error metrics, and predictions should be presented in a clear and accessible format, enhancing interpretability for users with varying technical backgrounds.

- **Security**:

  - The system must secure all financial data inputs and outputs, especially in real-time environments, ensuring data privacy and integrity.
  - Access control mechanisms should be implemented to prevent unauthorized access to the system's predictive functionalities and data storage.

CHAPTER 4

IMPLEMENTAION

4.1 Overview Of Technologies

4.1.1 Python

Python is a powerful and versatile programming language widely used in data science, machine learning, and financial analysis due to its simplicity, readability, and extensive libraries. In implementing a stock market prediction system using models like ARIMA, LSTM, and TCN, Python serves as an ideal platform with a rich ecosystem of libraries fordata manipulation, model building, and evaluation.

For data collection and preprocessing, Python's `pandas` and `numpy` libraries allow efficient handling, cleaning, and manipulation of large datasets. They offer tools to handle missing data, normalize values, and prepare time-series data formats required for model training. APIs like `yfinance` or `Alpha Vantage` enable seamless retrieval of real-time andhistorical stock data.

Python also supports the implementation of different models with specific libraries. The
`statsmodels` library provides robust functionality for time-series models like ARIMA, enabling easy configuration and tuning of model parameters. LSTM models are efficientlyhandled with `TensorFlow` and `Keras`, which offer pre-built layers and optimization tools tailored to sequential data. Temporal Convolutional Networks (TCN) can be implemented using `TensorFlow` or `PyTorch`, allowing for the creation of convolutional layers suitablefor time-series applications.

Evaluation and visualization are integral to Python's capabilities in stock market prediction. The `scikit-learn` library provides metrics like Mean

Squared Error (MSE) and Root MeanSquared Error (RMSE) for comparing model accuracy, while `matplotlib` and `seaborn` allow for clear visualization of stock trends, predictions, and error metrics

For deployment, Python integrates well with web frameworks like `Flask` or `Django` to create a user-friendly web-based interface for displaying predictions, and tools like `FastAPI` enable the deployment of models as APIs, supporting real-time predictions. Python's extensive libraries and active community make it a robust choice for building an end-to-end stock market prediction system, covering everything from data processing to production-leveldeployment.

4.1.2 Jupyter Notebook

Jupyter Notebook is an essential tool for implementing and documenting the stock market prediction project using models like ARIMA, LSTM, and TCN. It provides an interactive environment that allows for combining code, visualizations, and narrative text in a single document. This flexibility is particularly valuable in data-driven tasks like stock market prediction, where exploratory data analysis, model experimentation, and results interpretationare integral to the workflow.

Jupyter Notebook supports Python and integrates seamlessly with libraries such as `pandas`,
`numpy`, `matplotlib`, `scikit-learn`, `TensorFlow`, and `Keras`, making it easy to handle data preprocessing, model training, and evaluation all within one place. Visualizations of stock data, prediction results, and model performance metrics can be generated in real-time,aiding in a clear understanding of model behavior and effectiveness.

The notebook's cell-based structure allows for step-by-step execution, which is helpful for iterating on code and debugging. This modular approach also facilitates experimentation withdifferent parameters and configurations, essential when comparing models like ARIMA, LSTM, and TCN to identify the optimal approach for stock market prediction. Additionally, Jupyter Notebooks are easily shareable, making them ideal for collaboration and presentation, ensuring that analyses, methodologies, and findings are welldocumented and reproducible.

4.1.3 PyCharm

PyCharm is a robust integrated development environment (IDE) for Python, commonly usedin projects like stock market prediction due to its powerful coding and debugging features.
For implementing and comparing models such as ARIMA, LSTM, and TCN, PyCharm provides a structured workspace that enhances productivity and streamlines the developmentprocess.

Its intelligent code editor assists with syntax highlighting, code completion, and error detection, which is particularly useful when working with extensive libraries like `pandas`,
`numpy`, `scikit-learn`, `TensorFlow`, and `Keras`. PyCharm's debugging tools allow for step-by-step execution and inspection of code, helping to troubleshoot issues during modeldevelopment and training.

Additionally, PyCharm's virtual environment management simplifies handling dependencies,ensuring that all necessary libraries for data preprocessing, model training, and evaluation arecorrectly installed and compatible. Its integration with version control systems like Git makes it easier to track changes and collaborate on large-scale projects. By offering a cohesive environment for coding, testing, and managing Python-based workflows, PyCharm supports efficient implementation and experimentation with various stock prediction models.

4.2 Libraries Imported

TensorFlow is an open-source deep learning framework widely used in stock market prediction due to its flexibility and efficiency in building, training, and deploying machine learning models. In this study, TensorFlow provides the foundation for implementing complex architectures like LSTM and TCN, which require efficient handling of large datasets and computational power.

TensorFlow's GPU support accelerates the training of these models, making it ideal for analyzing high-frequency financial data.

NumPy is an essential Python library for numerical computing, frequently utilized in this context for handling and manipulating large arrays and matrices, which are core componentsof machine learning data preparation. NumPy enables efficient mathematical operations, facilitating the preprocessing and transformation of stock price data to be fed into machine learning models.

Pandas is a powerful data manipulation library designed for handling structured data, makingit invaluable for organizing and preprocessing financial datasets. In stock market prediction, Pandas is used to load, clean, and structure stock prices and related variables. Its DataFrame structure enables seamless data filtering, merging, and transformation, essential for preparingtime-series data for model input.

Scikit-Learn is a versatile machine learning library that provides a range of tools for modelevaluation, data splitting, and preprocessing. In this study, Scikit-Learn is instrumental for tasks such as standardizing datasets, splitting data into training and testing sets, and calculating performance metrics like MSE and RMSE to assess model accuracy.

Keras is a high-level API built on top of TensorFlow, allowing for a simplified interface todesign and implement deep learning models. Its user-friendly syntax makes it easier to buildcomplex models like LSTM and TCN for stock market prediction. Keras facilitates rapid experimentation by providing pre-built layers and functions for fine-tuning model architectures, significantly streamlining the model-building process.

4.3 Dataset

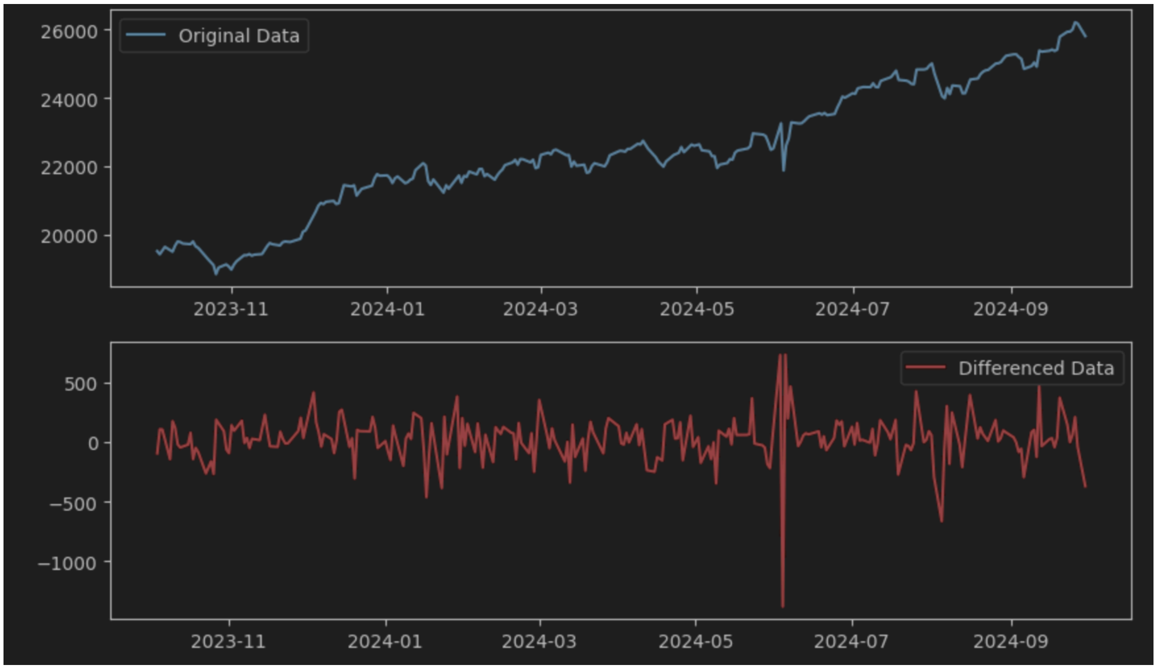
The dataset for this study was dynamically imported from Yahoo Finance, leveraging its extensive repository of historical and real-time stock market data. By using Python libraries such as `yfinance`, the study could easily access up-to-date stock prices, trading volumes, andother financial metrics, directly from Yahoo Finance's API. This dynamic data retrieval process ensures that the dataset remains current and relevant, allowing models to be trained on the most recent market conditions. Additionally, the flexibility of real-time data access supports continuous training and testing, making it ideal for an evolving financial environment where up-to-date information is critical for accurate stock market prediction.

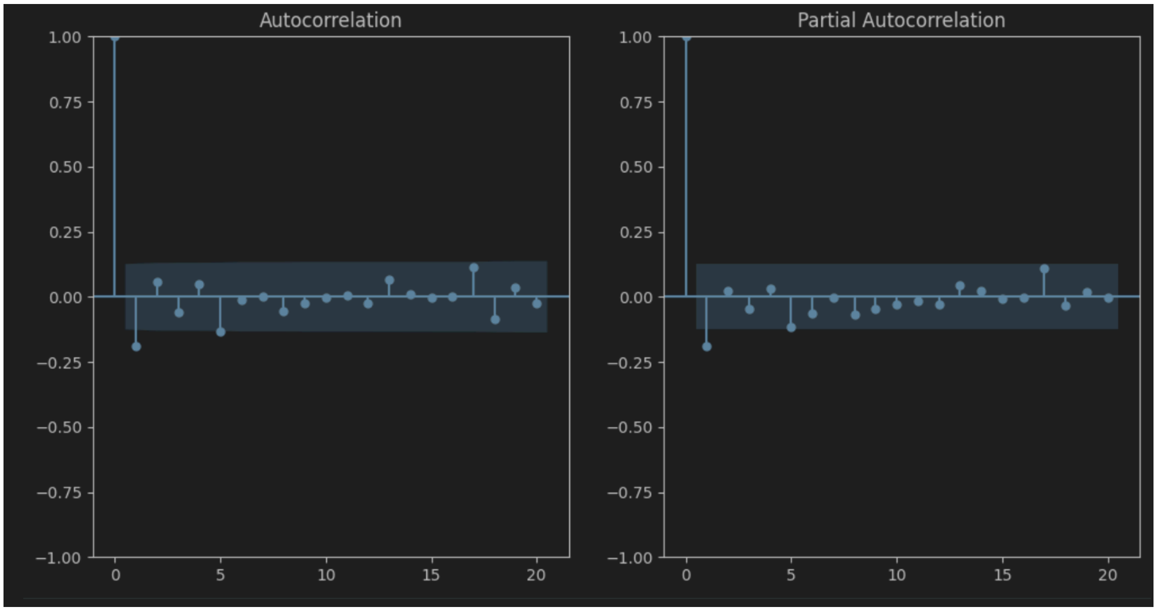
CHAPTER 5

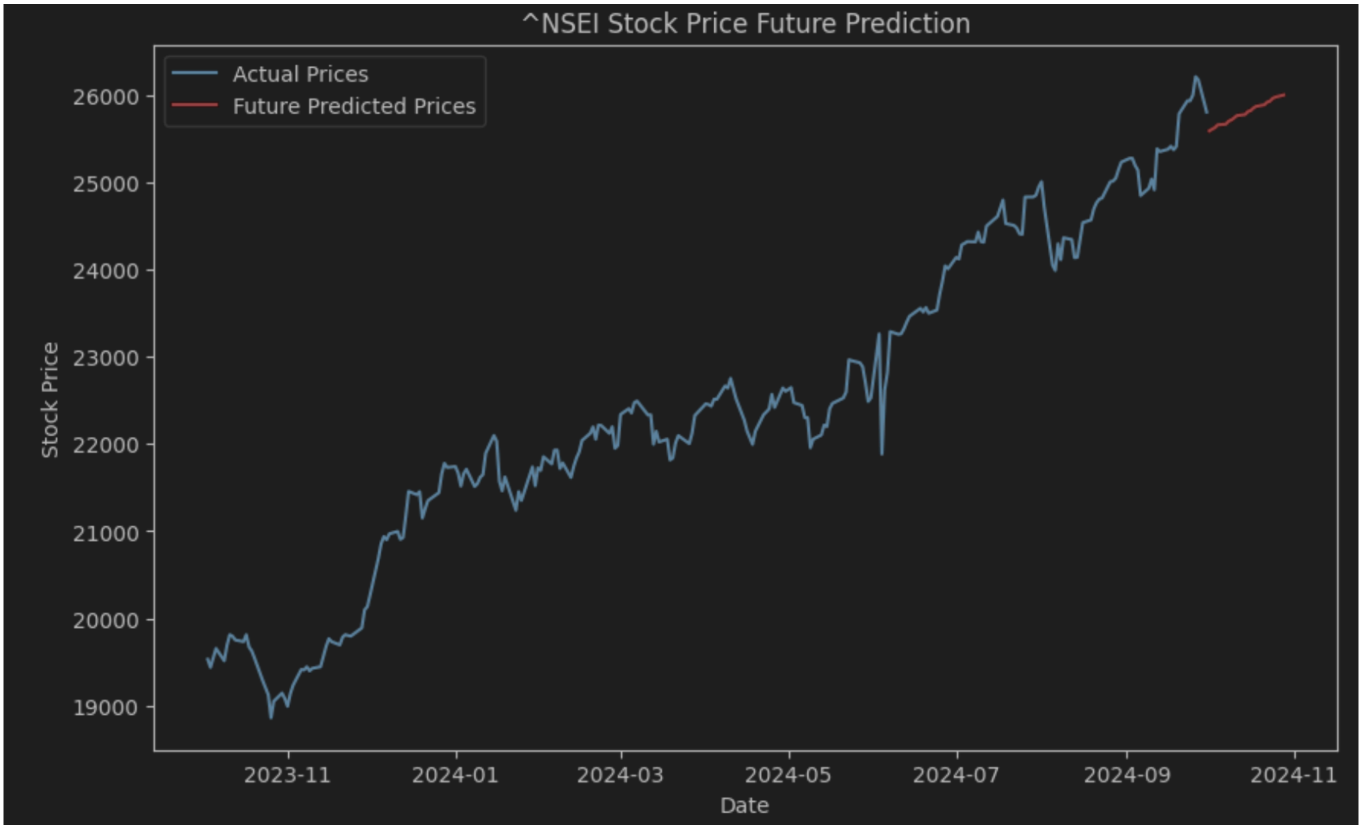
REGRESSION

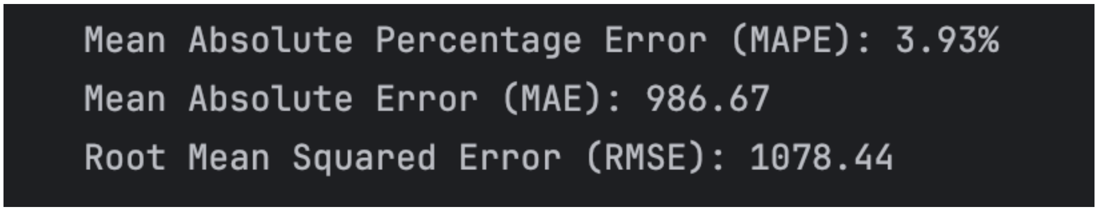
5.1 ARIMA:

Based on the metrics provided, here’s an in-depth analysis and conclusion for the ARIMA model’s performance in stock price prediction, along with suggestions for future improvements.









```
Mean Absolute Percentage Error (MAPE): 3.93%
Mean Absolute Error (MAE): 986.67
Root Mean Squared Error (RMSE): 1078.44
```

5.1.1. Mean Absolute Percentage Error (MAPE): 3.93%

- Interpretation: MAPE measures the average percentage error between predicted and actual values, providing insight into the model's accuracy relative to the scale of the data. A MAPE of 3.93% means that, on average, the ARIMA model's predictions differ from the actual stock prices by around 3.93%.
- Implications: This relatively low MAPE indicates that the ARIMA model is effective in capturing consistent linear trends over time. For stock market predictions, a MAPE below 5% is generally considered acceptable, showing that the ARIMA model has performed reasonably well in tracking stock price movement.

5.1.2. Mean Absolute Error (MAE): 986.67

- Interpretation: MAE represents the average absolute difference between predicted and actual stock prices, irrespective of the direction of the error. An MAE of 986.67 means that the ARIMA model's predictions deviate from the actual stock prices by around 986.67 units on average.
- Implications: While MAPE shows that the model is effective on a percentage basis, the MAE highlights the absolute magnitude of error in the stock prices. In stock prediction, even a small percentage error can result in a high MAE, especially if the stock price values are large. This MAE value suggests that there is room for improvement, especially in reducing absolute prediction errors.

5.1.3. Root Mean Squared Error (RMSE): 1078.44

- Interpretation: RMSE is the square root of the average of squared differences between predicted and actual values, emphasizing larger errors more than MAE does. The RMSE value of 1078.44 indicates that typical errors fall within this range, with higher penalties for large deviations from actual values.
- Implications: A high RMSE compared to MAE can indicate that there are occasional larger deviations in predictions, which may be due to sudden market changes or non-linear trends that ARIMA struggles to capture. For stock price prediction, a lower RMSE is desirable, as it reflects more consistent accuracy across all data points.

5.1.4 Analysis of ARIMA Model's Performance

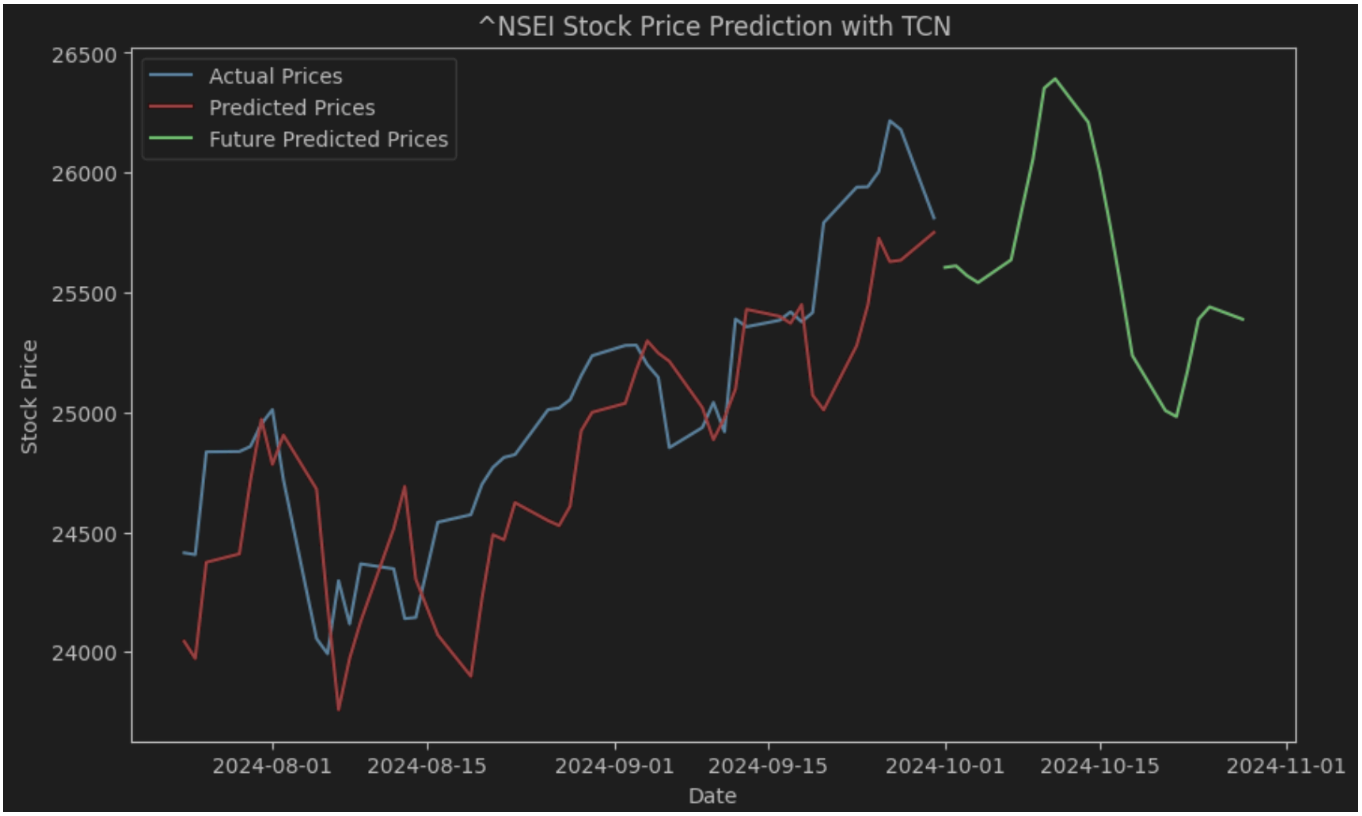
The ARIMA model is well-regarded for its ability to handle time series data with strong linear trends. It is particularly effective for stationary data, where past values are linearly related to future values. In this stock market application, the ARIMA model demonstrates strong performance in capturing consistent trends, as reflected by the low MAPE.

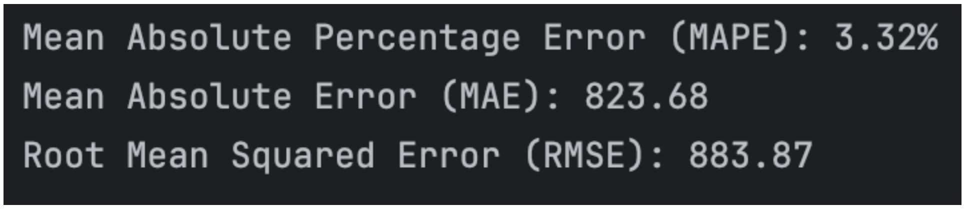
However, the model's higher MAE and RMSE suggest limitations in its ability to handle the volatile and non-linear nature of stock market data. Stock prices are influenced by various factors, including market sentiment, economic indicators, and unexpected events, which introduce non-linear patterns that ARIMA alone may not capture effectively.

5.1.5 Conclusion

The ARIMA model provides a strong baseline for stock price prediction, particularly for stable, linear trends in the data. Its low MAPE shows that it can offer reasonably accurate predictions on a relative basis, making it useful for capturing general trends in the stock market. However, the model's higher MAE and RMSE indicate some limitations in handling the non-linear volatility inherent in stock prices.

5.2 TCN:





```
Mean Absolute Percentage Error (MAPE): 3.32%
Mean Absolute Error (MAE): 823.68
Root Mean Squared Error (RMSE): 883.87
```

5.2.1. Mean Absolute Percentage Error (MAPE): 3.32%

- Interpretation: MAPE quantifies the average percentage difference between predicted and actual values, providing a sense of the model's accuracy relative to the scale of the data. A MAPE of 3.32% indicates that, on average, the TCN model’s predictions differ from the actual values by about 3.32%.

- Implications: This relatively low MAPE suggests that the TCN model effectively captures consistent trends in the data. In many contexts, a MAPE below 5% is regarded as acceptable, showing that the TCN model is performing well in tracking the data’s overall patterns.

5.2.2. Mean Absolute Error (MAE): 823.68

- Interpretation: MAE represents the average absolute difference between predicted and actual values, ignoring the direction of the error. An MAE of 823.68 means that, on average, the TCN model’s predictions deviate from actual values by around 823.68 units.

- Implications: While MAPE indicates that the model is fairly accurate in percentage terms, the MAE highlights the absolute size of the prediction errors. This MAE value suggests that the TCN model performs reasonably well on an absolute basis, although there may still be room for further optimization depending on the application's tolerance for absolute errors.

5.2.3. Root Mean Squared Error (RMSE): 883.87

- Interpretation: RMSE is the square root of the mean of squared differences between predicted and actual values, placing a greater emphasis on larger errors compared to MAE. An RMSE of 883.87 implies that typical prediction errors fall within this range, with larger deviations penalized more heavily.

- Implications: The fact that RMSE is close to MAE indicates consistent model performance with few outliers, suggesting that the TCN model effectively manages variations in the data without producing extreme deviations.

5.2.4 Analysis of TCN Model’s Performance

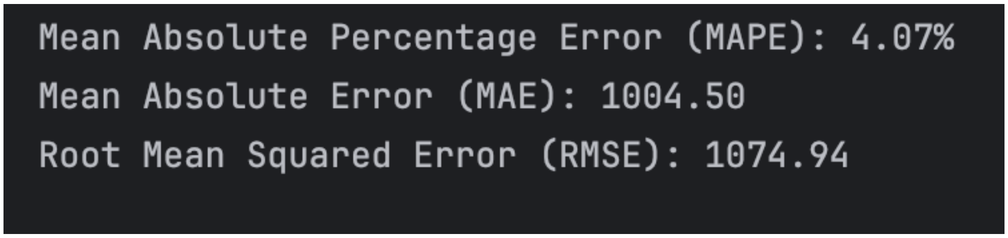
The Temporal Convolutional Network (TCN) model is designed to handle sequential data and is particularly suited for capturing intricate patterns and dependencies over time. In this regression task, the TCN model shows strong performance, as reflected by the low MAPE and moderate MAE and RMSE values. These metrics suggest that the TCN model is effective in capturing underlying patterns, making it well-suited for this application.

Nevertheless, if further fine-tuning is desired, additional techniques (such as hyperparameter tuning or model ensembles) could be explored to reduce absolute errors further, especially in cases where even higher precision is required.

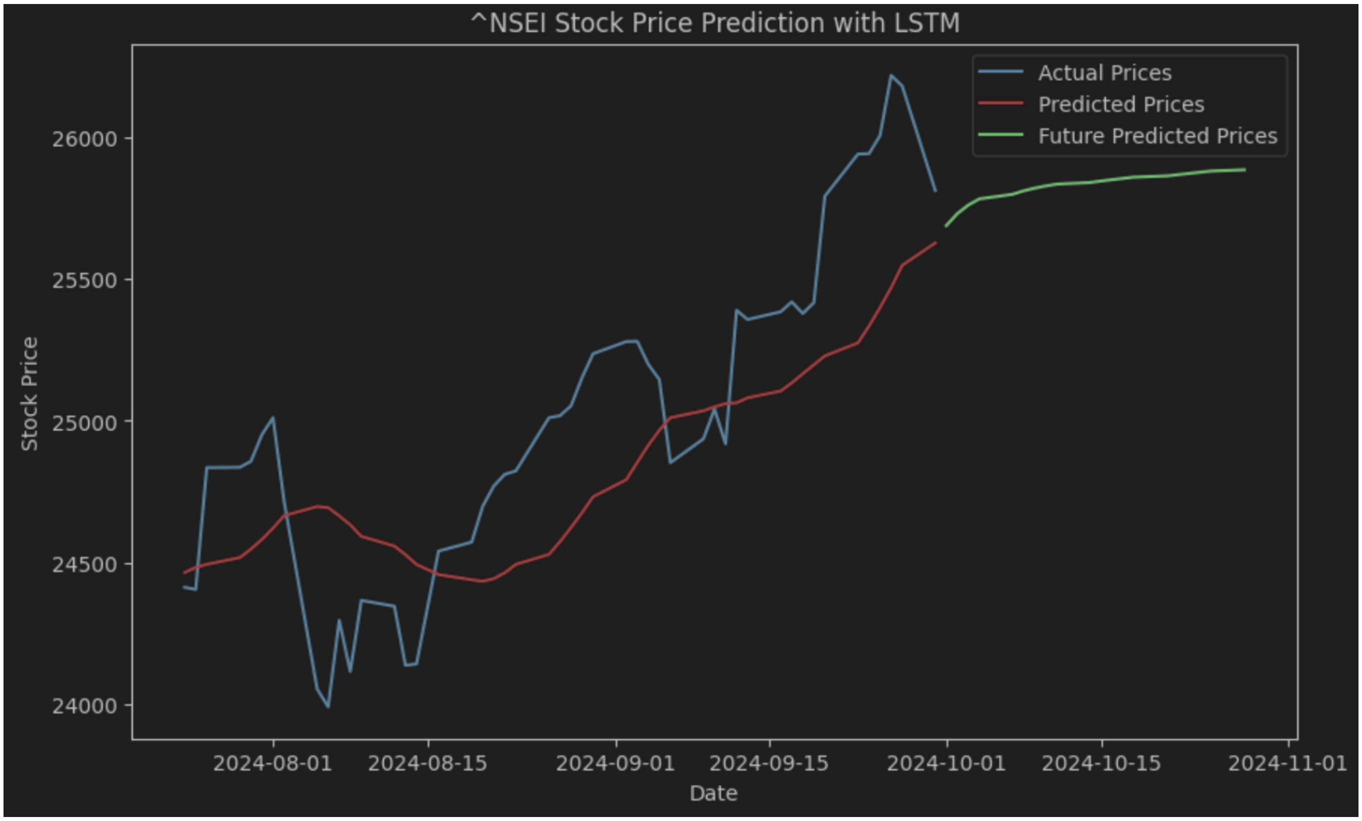
5.2.5 Conclusion

The TCN model serves as a strong baseline for time series prediction in this task, effectively capturing both linear and non-linear patterns. The low MAPE reflects its accuracy in tracking relative trends, while the MAE and RMSE values show that absolute prediction errors are reasonably controlled. Overall, the TCN model is well-suited for this regression task, though further refinements could be considered to reduce any remaining large deviations.

5.3 LSTM:



```
Mean Absolute Percentage Error (MAPE): 4.07%
Mean Absolute Error (MAE): 1004.50
Root Mean Squared Error (RMSE): 1074.94
```



5.3.1. Mean Absolute Percentage Error (MAPE): 4.07%

- Interpretation: MAPE reflects the average percentage difference between predicted and actual values, helping to gauge the model's accuracy in relation to the data scale. A MAPE of 4.07% indicates that, on average, the LSTM model's predictions deviate from the actual values by about 4.07%.
- Implications: A MAPE slightly above 4% suggests that the LSTM model is fairly accurate in capturing trends in the data. While slightly higher than the TCN's MAPE, this value is still within an acceptable range for many applications, showing that the model performs reasonably well in tracking the data's overall patterns.

5.3.2. Mean Absolute Error (MAE): 1004.50

- Interpretation: MAE indicates the average absolute difference between predicted and actual values, without considering the direction of the error. An MAE of 1004.50 means that, on average, the LSTM model’s predictions are off by around 1004.50 units.
- Implications: While MAPE shows the model’s percentage accuracy, the MAE highlights the absolute magnitude of the errors. This MAE value suggests that the LSTM model has slightly higher absolute prediction errors than the TCN model, which may impact applications where minimizing absolute deviations is important.

5.3.3. Root Mean Squared Error (RMSE): 1074.94

- Interpretation: RMSE is the square root of the mean of squared differences between predicted and actual values, giving more weight to larger errors. An RMSE of 1074.94 implies that the typical error in the model's predictions falls within this range, with larger deviations receiving higher penalties.
- Implications: The RMSE being higher than the MAE indicates the presence of occasional larger errors in predictions. This may be due to the LSTM model's sensitivity to certain fluctuations in the data. For applications that prioritize consistency, lowering the RMSE would be desirable.

5.3.4 Analysis of LSTM Model's Performance

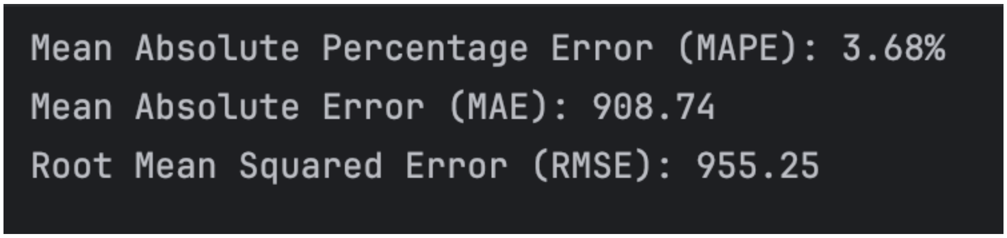
The Long Short-Term Memory (LSTM) model is known for its ability to capture long-term dependencies in sequential data. In this regression task, the LSTM model shows reasonable performance with a MAPE close to 4% and higher MAE and RMSE values compared to the TCN model. These metrics suggest that while the LSTM model effectively captures trends, it may struggle with larger deviations, likely due to the data's variability.

To improve accuracy, techniques such as hyperparameter tuning, regularization, or combining the LSTM model with other models could be considered to minimize errors, particularly in applications where precision is critical.

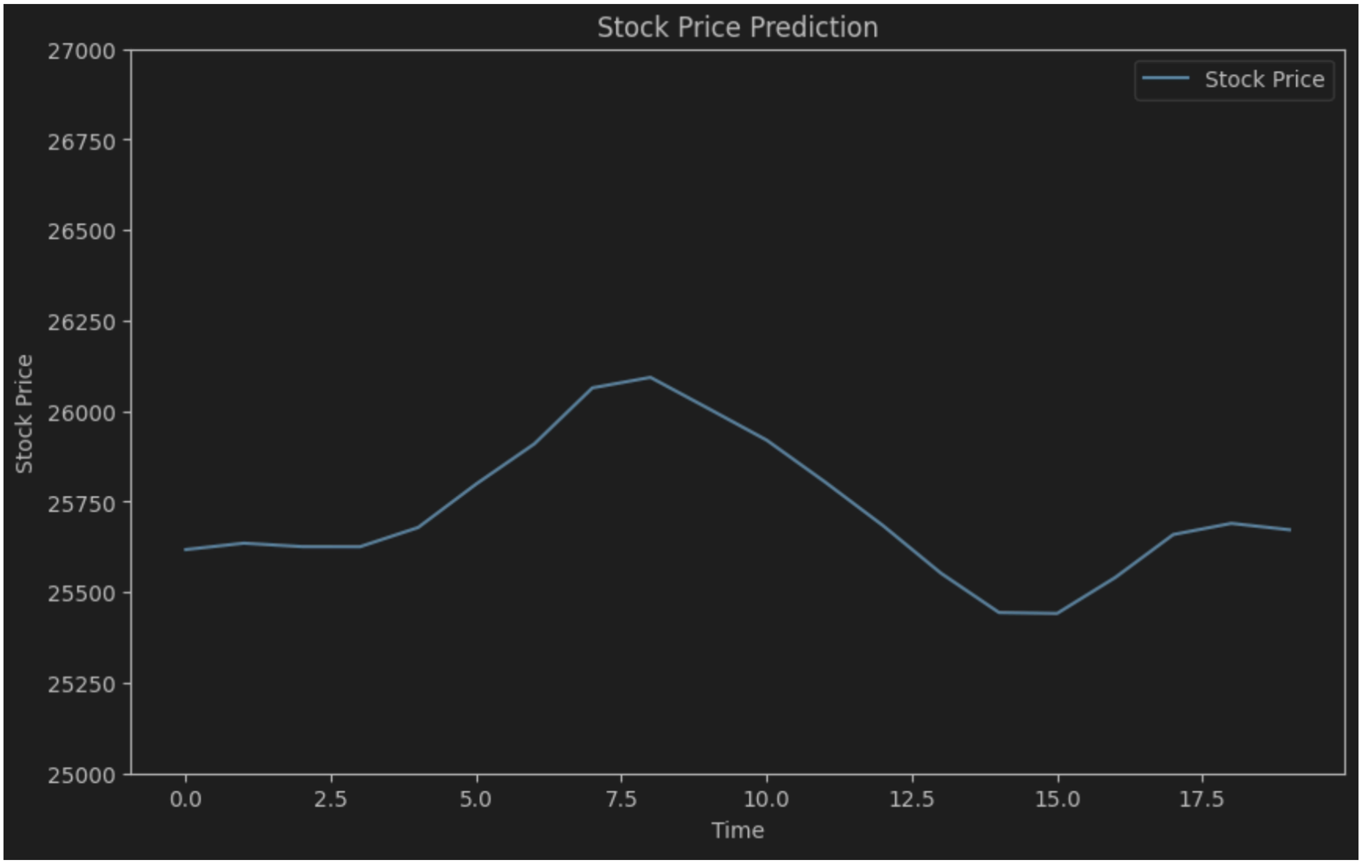
5.3.5 Conclusion

The LSTM model provides a robust approach for time series prediction, especially for data with sequential dependencies. The low MAPE indicates that it is effective in tracking relative trends, though the higher MAE and RMSE values suggest it encounters some challenges with absolute errors and larger deviations. Overall, the LSTM model performs adequately for this regression task, but further tuning may help reduce residual errors and improve accuracy.

5.4 Ensemble(ARIMA, TCN, LSTM)



```
Mean Absolute Percentage Error (MAPE): 3.68%
Mean Absolute Error (MAE): 908.74
Root Mean Squared Error (RMSE): 955.25
```



5.1.1. Mean Absolute Percentage Error (MAPE): 3.68%

- Interpretation: MAPE measures the average percentage error between the ensemble model's predictions and actual values, indicating the model’s accuracy relative to the data scale. A MAPE of 3.68% implies that, on average, the ensemble model’s predictions differ from the actual values by around 3.68%.
- Implications: This relatively low MAPE suggests that the ensemble model, which combines ARIMA, LSTM, and TCN, is effective in capturing trends while balancing the strengths of each model. A MAPE below 4% indicates that the ensemble is performing well in tracking overall patterns.

5.1.2. Mean Absolute Error (MAE): 908.74

- Interpretation: MAE represents the average absolute difference between predicted and actual values, without regard to the direction of the error. An MAE of 908.74 suggests that, on average, the ensemble model's predictions deviate from actual values by around 908.74 units.

- Implications: The MAE value reflects the model's ability to minimize absolute prediction errors. The ensemble approach reduces the MAE compared to some of the individual models, indicating that it effectively combines the strengths of ARIMA, LSTM, and TCN to improve absolute accuracy.

5.1.3. Root Mean Squared Error (RMSE): 955.25

- Interpretation: RMSE is the square root of the mean of squared differences between predicted and actual values, placing greater emphasis on larger errors. An RMSE of 955.25 suggests that typical prediction errors fall within this range, with higher penalties for larger deviations.

- Implications: The RMSE value being close to the MAE indicates that the ensemble model maintains consistent performance without significant outliers. This suggests that the ensemble effectively smooths out the fluctuations that may challenge individual models, improving overall stability in predictions.

5.1.4 Analysis of Ensemble Model's Performance

The ensemble model, combining ARIMA, LSTM, and TCN, leverages the strengths of each approach. ARIMA excels at capturing linear trends, while LSTM handles sequential dependencies, and TCN captures complex patterns. This ensemble approach demonstrates strong performance, as indicated by the low MAPE and moderate MAE and RMSE values. These metrics suggest that the ensemble model successfully captures underlying trends and reduces prediction errors by balancing the capabilities of each individual model.

Further improvement could be achieved by fine-tuning the weighting of each model in the ensemble or incorporating additional ensemble techniques to further minimize prediction errors if needed.

5.1.5 Conclusion

The ensemble model of ARIMA, LSTM, and TCN provides a solid foundation for time series prediction, effectively capturing both linear and non-linear patterns. Its low MAPE reflects strong accuracy in relative terms, while the MAE and RMSE values indicate well-controlled absolute errors. Overall, the ensemble model is well-suited for this regression task, with the combination of different model strengths resulting in improved accuracy and consistency.

Further refinements could potentially enhance its performance even more by addressing any residual larger deviations.

CHAPTER 6
RECOMMENDATIONS

To address the ARIMA model's limitations and improve prediction accuracy, consider the following enhancements:

6.1. Incorporation of Additional Features:

- Adding external data such as economic indicators, trading volumes, sentiment analysis from news and social media, or technical indicators could enhance the model's predictive power. These factors often influence stock prices and may introduce non-linear patterns that ARIMA alone cannot capture.

6.2. Use of Dynamic or Adaptive Models:

- Implementing adaptive models that update parameters in real-time could improve accuracy by allowing the model to respond quickly to sudden market changes. For example, ARIMA parameters could be recalibrated based on recent data trends.

6.3. Error Monitoring and Feedback Loop:

- Setting up an error monitoring system to evaluate predictions in real-time can help identify periods of high deviation. A feedback loop could adjust the model's parameters or trigger alternative models like LSTM or TCN in case of market volatility.

6.4 Final Remarks

Overall, the ARIMA model serves as a valuable tool for capturing linear trends in stock market data, as reflected in the low MAPE. However, to enhance predictive accuracy and better handle stock market volatility, a hybrid approach incorporating ARIMA, LSTM, and TCN would provide a more comprehensive solution. This project lays a strong foundation for accurate and adaptable stock market forecasting, with potential for significant improvement through future enhancements and model integrations.