Dissertation on

"DeepTicker: Stock Price Prediction using Deep Learning and Sentiment Analysis"

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

Master of Technology in

Data Science & Machine Learning

UE20CS971 – Project Phase - 1

Submitted by:
Ravi Kumar Gupta
PES2PGE24DS043

Under the guidance of
Dr. Sagarika Bohra
Prof. Data Science, PES University

Sept - Dec 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

Certificate

This is to certify that the dissertation entitled “DeepTicker: Stock Price Prediction using Deep Learning and Sentiment Analysis” is a bonafide work carried out by **Student Name (PES2PGE24DS043)** in partial fulfilment for the completion of Fourth Semester Project Phase - 2 (UE20CS971) in the Program of Study - Master of Technology in Computer Science and Engineering under the rules and regulations of PES University, Bengaluru during the period Sept. 2025 – Dec. 2025.

Dr. Sagarika Bohra
Prof. DSML

Declaration

We hereby declare that the Project Phase - 1 entitled “DeepTicker: Stock Price Prediction using Deep Learning and Sentiment Analysis“ has been carried out under the guidance of Dr. Guide Name, Department of CSE, and submitted in partial fulfilment of the course requirements for the award of the degree of Master of Technology in Computer Science and Engineering of PES University, Bengaluru during the academic semester September – December 2025.

Acknowledgement

I would like to express my gratitude to Dr. Sagarika Bohra, Department of Data Science and Machine Learning, PES University, for her continuous guidance and support. I am also thankful to the project coordinators for their help throughout the process. I take this opportunity to thank the Chairperson of the Department of Data Science and Machine Learning, PES University, for the support and guidance. I am grateful to Dr. B.K. Keshavan, Dean of Faculty, and the university leadership for the opportunities provided. Finally, I extend heartfelt thanks to my family and friends for their constant support.

Abstract

This project aims to predict the next-day closing price of NSE-listed stocks using advanced deep learning techniques. The data is sourced from NSETools, NSE India APIs, yFinance, and custom CSV uploads, ensuring a diverse and reliable dataset. In Phase 1, an LSTM-based model is developed to capture temporal dependencies in historical price movements. In Phase 2, the model's accuracy will be enhanced using transformer-based architectures and other advanced neural network approaches. The final output will be an interactive dashboard that visualizes predictions, trends, and performance metrics to support data-driven investment decisions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

- Stock market is a complex and dynamic system that plays a crucial role in the global economy.
- Stock Prices are influenced by many factors like, economic indicator, market sentiment, geopolitical events, company-specific news, environmental events such as natural disasters, epidemics and pandemics, political tensions, wars, fiscal policies etc.
- Accurate prediction of stock prices is of immense importance to investors, financial analysts, and policy makers.
- Successful prediction can lead to significant financial gains, informed decision making, and improved economic stability.

1.2 Objectives

1.2.1 Primary Objective

- To Design and develop a deep learning-based framework for predicting the next-day closing price of stocks using historical market data.

1.2.2 Secondary Objective

- To Collect and preprocess historical OHLCV data from multiple financial data sources such as NSE APIs, Yahoo Finance, and CSV datasets.
- To perform extensive feature engineering using technical indicators like moving averages, RSI, volatility, and volume-based metrics.
- To develop and train an LSTM-based time series forecasting model for stock price prediction.
- To evaluate the model performance using standard metrics such as RMSE, MAE, and R^2 score.

- To compare the performance of LSTM with other deep learning architectures such as GRU, CNN-LSTM, and Transformer-based models.
- To analyze the impact of different features on prediction accuracy.
- To visualize predicted vs actual stock prices through an interactive dashboard.

1.3 Scope of the Project

- The project focuses on predicting the next-day closing price of selected stocks using historical market data.
- It utilizes time-series data such as Open, High, Low, Close, and Volume obtained from reliable sources like NSE and Yahoo Finance.
- Advanced deep learning models, primarily LSTM, are implemented, with scope for experimenting with Transformer-based architectures to improve prediction accuracy.
- The project is limited to short-term price forecasting and does not include live trading execution or portfolio optimization.

Chapter 2

Literature Survey

2.1 Existing Systems

With rapid technological advancements, financial markets have become a cornerstone of the global economy and play a vital role in wealth distribution. Among these, the stock market serves as a critical component of the global financial system and is widely regarded as a reliable indicator of a nation's economic stability. [?]

Recent studies highlight the growing importance of deep learning techniques—particularly Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Gated Recurrent Units (GRU)—in stock market prediction tasks due to their strong capability to model sequential data and capture complex temporal patterns present in financial time series data [?]. Furthermore, hybrid architectures such as CNN–LSTM and BiLSTM–GRU have demonstrated superior predictive performance by effectively combining feature extraction and sequence learning mechanisms [?] [?].

Metric-driven and data-centric approaches have significantly transformed the domain of stock price forecasting, prompting extensive research into various machine learning and deep learning models. In the proposed study, multiple predictive models have been reviewed to evaluate their effectiveness in stock market price prediction. This comparative analysis aids in identifying the most suitable neural network architectures for forecasting tasks and provides deeper insights into the development of highly accurate hybrid models.

Stock price prediction is predominantly approached as a time series forecasting problem. Existing research in this area can broadly be classified into two major categories. The first category focuses on LSTM-centric architectures, where LSTM is integrated with complementary techniques such as CNN, GRU, and Graph Neural Networks (GNN), along with advanced methodologies including Variational Mode Decomposition (VMD), Triangular Maximally Filtered Graphs (TMFG) [?], and Self-Supervised Attention Mechanisms (SSAM). Additionally, several studies incorporate sentiment analysis to capture market psychology and investor sentiment for enhanced predictive accuracy [?].

The second category emphasizes transformer-based models as the core architecture, often combined with CNNs, GNNs, Time2Vec [?] [?] and advanced variants such as Mamba or Bidirectional Mamba models [?]. Research under this category has been conducted across multiple global stock exchanges, achieving exceptionally high prediction accuracies, in some cases exceeding 98% [?]. However, despite extensive studies on international markets, limited research explicitly reports accuracy benchmarks for stock price prediction within the Indian stock market, particularly for the BSE and NSE exchanges.

2.2 Limitations of Existing Systems

Despite significant advancements in stock price prediction using machine learning and deep learning techniques, existing systems still face several limitations that restrict their effectiveness and general applicability.

Most traditional statistical models such as ARIMA and linear regression assume linear relationships and stationarity in financial time-series data. However, stock market data is inherently non-linear, noisy, and highly volatile, making these models insufficient for accurate long-term predictions.

Although deep learning models such as LSTM and GRU have improved prediction performance by capturing temporal dependencies, many existing systems rely on a single model architecture, which limits their ability to generalize across different market conditions. Additionally, several studies focus on short-term historical datasets and do not adequately evaluate model robustness across extended time periods.

Another major limitation is the lack of comprehensive feature engineering. Many systems depend solely on OHLC data without incorporating advanced technical indicators or market behavior patterns. Furthermore, sentiment analysis-based approaches often rely on limited or unstructured data sources, which may introduce bias and inconsistency.

Transformer-based models have demonstrated high accuracy in global markets; however, their application in the Indian stock market (BSE and NSE) remains limited. Existing studies often lack comparative analysis between recurrent and attention-based models under identical datasets and evaluation metrics. Additionally, most systems do not support real-time scalability, multi-stock training, or automated retraining mechanisms.

2.3 Proposed Solution

To overcome the limitations of existing systems, this project proposes a hybrid deep learning-based stock price prediction framework designed specifically for the Indian stock market.

The proposed system integrates multiple data sources, including NSE Tools, NSE India APIs, yFinance, and CSV-based historical datasets, ensuring data diversity and reliability. A comprehensive preprocessing pipeline is implemented to clean, normalize, and engineer features such as moving averages, volatility indicators, and volume-based metrics.

In the first phase, an LSTM-based model is developed to capture temporal dependencies in stock price movements. In the second phase, advanced architectures such as CNN-LSTM, GRU, and Transformer models with attention mechanisms are implemented to improve prediction accuracy and capture long-range dependencies. A comparative evaluation is conducted to identify the most effective model for next-day stock price prediction.

The proposed system is designed to support multiple stocks using a unified framework and enables scalable model training and evaluation. Prediction results are stored and visualized through interactive dashboards, aiding in analytical decision-making. By combining hybrid deep learning models, extensive feature engineering, and market-specific data, the proposed solution aims to provide a robust and accurate stock price prediction system for the Indian financial market.

Chapter 3

System Requirements Specification

3.1 Hardware Requirements

- **Processor:**
  Intel Core i5 / AMD Ryzen 5 or higher
- **RAM:**
  Minimum 8 GB (16GB recommended for faster training)
- **Storage:**
  Minimum 50 GB free disk space (for datasets, models, logs)
- **GPU (Optional but Recommended):**
  NVIDIA GPU with CUDA support (for faster deep learning training)
- **Internet Connectivity:**
  Required for fetching real-time and historucal stock market data

3.2 Software Requirements

The following software tools and libraries are required:

- **Operating System**:
  Windows 10 (or later)/ Linux (Ubuntu 20.04 or later)
- **Programming Language:**
  Python 3.8 or higher
- **Database:**
  MySQL / PostgreSQL (for storing historical stocj data)
- **Library and Frameworks:**
  - Numpy
  - Pandas
  - Matplotlib / Seaborn
  - Scikit-learn

- TensorFlow / Pytorch
- Keras
- yFinance / NSE Tools

- **Development Tools:**
    - Jupyter Notebook
    - Visual Studio Code / PyCharm
- **Version Control:**
    - Git

3.3 Functional Requirements

1. The system shall collect historical stock market data from multiple sources such as yFinance, NSE tools, and CSV uploads.
2. The system shall preprocess and clean the stock market data.
3. The system shall perform feature engineering (moving averages, RSI, volatility, etc.).
4. The system shall train deep learning models such as LSTM, CNN-LSTM, GRU, and Transformer-based models.
5. The system shall predict the next-day closing price of selected stocks.
6. The system shall support multiple stocks using a single trained model.
7. The system shall store prediction results in a database.
8. The system shall visualize historical prices and predicted prices through charts and dashboards.

3.4 Non-Functional Requirements

Performance

- The system should generate predictions within acceptable time limits.
- Model retraining should be optimized to minimize computation time.

Scalability

- The system should handle an increasing number of stocks and large datasets.
- The architecture should support future model enhancements.

Reliability

- The system should handle missing or inconsistent data gracefully.
- Predictions should remain stable across multiple executions.

Security

- The system should prevent unauthorized access to stored data.
- API credentials and database credentials must be securely stored.

Maintainability

- The codebase should be modular and well-documented.
- New features and models should be easily integrable.

3.5 Use Case Diagram

Actors

- User (Analyst / Researcher)
- Data Source (yFinance, NSE APIs)

Use Cases

1. Fetch Historical Stock Data
2. Upload CSV Data
3. Preprocess Data
4. Train Prediction Model
5. Predict Stock Prices
6. View Prediction Results
7. Store Data and Predictions

Interaction Summary

- The user initiates data collection.
- The system fetches and processes data.
- The model is trained and generates predictions.
- Results are displayed via charts or dashboards.

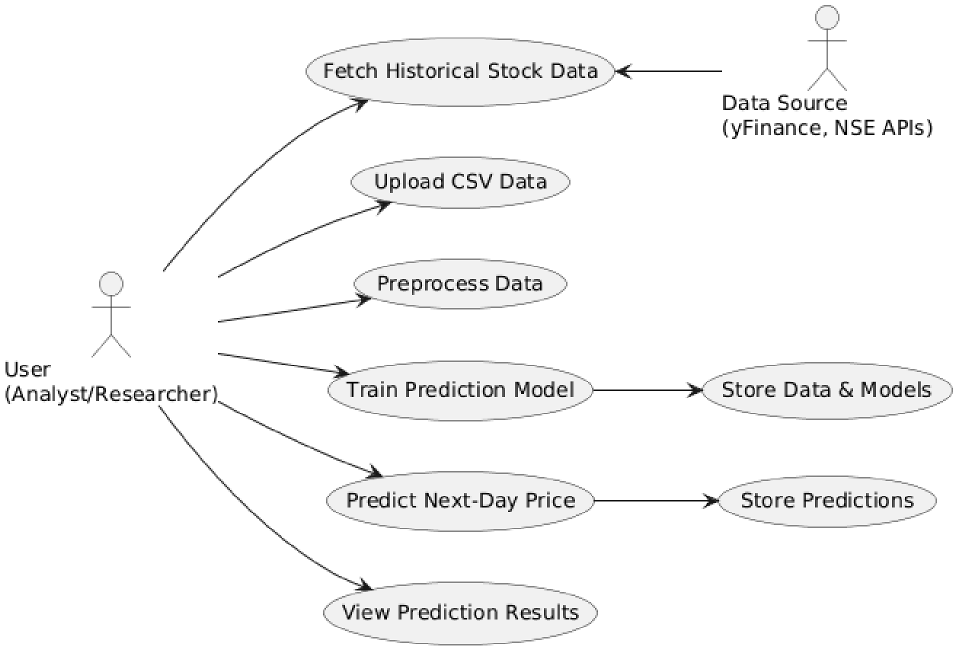

Figure 3.1: Use Case Diagram

Chapter 4

Proposed Methodology

4.1 System Architecture

The proposed system follows a modular and layered architecture to efficiently handle data collection, preprocessing, model training, and stock price prediction. The architecture is designed to support multiple data sources and deep learning models while ensuring scalability and flexibility.

The system begins with a data acquisition module that gathers historical stock market data from APIs such as yFinance, NSE Tools, NSE India APIs, and user-uploaded CSV files. The collected data is stored in a centralized relational database for further processing.

A data preprocessing and feature engineering module cleans the data, handles missing values, normalizes features, and generates technical indicators such as moving averages and volatility measures. The processed data is then passed to the model training module, which trains deep learning models including LSTM, CNN-LSTM, GRU, and Transformer-based architectures.

Finally, the prediction and visualization module generates next-day stock price predictions and displays historical and predicted values through graphical dashboards, enabling effective analysis and decision support.

4.2 Algorithm or Model Description

The core objective of the proposed system is to predict the next-day closing price of stocks using deep learning models capable of capturing temporal dependencies in financial time-series data.

LSTM Model

Long Short-Term Memory (LSTM) networks are used as the primary model due to their ability to learn long-term dependencies and mitigate the vanishing gradient problem. LSTM processes sequential OHLCV data and captures temporal patterns influencing stock price movements.

Hybrid Models

Hybrid architectures such as CNN-LSTM and GRU-based models are employed to enhance prediction accuracy. CNN layers extract meaningful features from time-series data, while LSTM or GRU layers learn sequential dependencies.

Transformer Model

In the second phase, Transformer models with attention mechanisms are introduced to capture global dependencies in the data. These models enable parallel processing and improved learning of long-range relationships, potentially outperforming recurrent architectures.

All models are trained using historical stock data and evaluated using performance metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

4.3 Workflow Diagram

The workflow of the proposed system follows a sequential pipeline:

1. Historical stock market data is collected from external APIs and CSV uploads.
2. The raw data is validated and stored in a database.
3. Data preprocessing is performed to clean, normalize, and engineer features.
4. Time-series sequences are created for model training.
5. Deep learning models are trained and validated using historical data.
6. The trained model predicts the next-day closing price.
7. Prediction results are stored and visualized through dashboards.

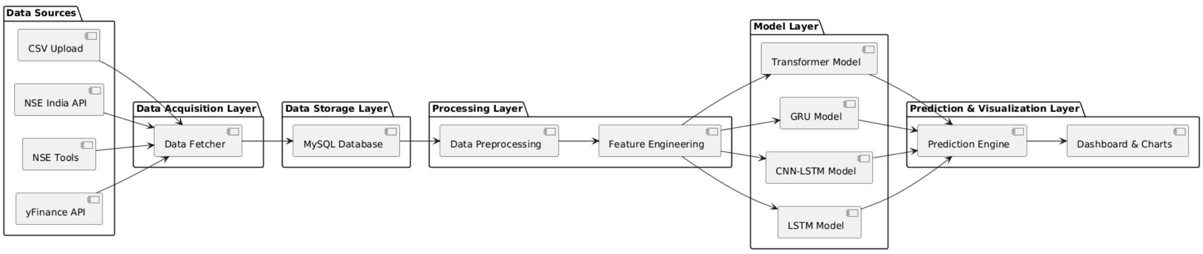

Figure 4.1: System Architecture Diagram

Chapter 5

Implementation Details

5.1 Development Environment

The proposed system is implemented using modern data science and deep learning tools to ensure efficiency, scalability, and reproducibility.

- Operating System:
  Windows 10 / Ubuntu Linux
- Programming Language:
  Python 3.8+
- Development Tools:
  Jupyter Notebook, Visual Studio Code
- Libraries and Frameworks:
  - NumPy and Pandas for data manipulation
  - Matplotlib and Seaborn for data visualization
  - Scikit-learn for preprocessing and evaluation
  - TensorFlow / Keras for deep learning model implementation
- Database:
  MySQL for storing historical stock data and predictions
- Version Control:
  Git for source code management
- Hardware:
  - Intel Core i5 or higher
  - 8–16 GB RAM
  - Optional GPU for accelerated training

5.2 Dataset Description

The dataset used in this project consists of historical stock market data collected from multiple sources to ensure reliability and completeness.

Data Sources

- NSE Tools
- NSE India APIs
- yFinance
- User-uploaded CSV files

Dataset Attributes

Each record in the dataset includes the following attributes:

- Open price
- High price
- Low price
- Close price
- Trading volume
- Delivery quantity (where available)
- Date and stock identifiers (ISIN, symbol, exchange)

Preprocessing Steps

- Removal of duplicate and inconsistent records
- Handling of missing values
- Date alignment across different data sources
- Normalization and scaling of numerical features
- Feature engineering including moving averages and volatility indicators

The processed dataset is converted into time-series sequences suitable for deep learning models.

5.3 Model Implementation

The model implementation is carried out in two phases to systematically improve prediction accuracy.

Phase 1: LSTM Model
An LSTM-based neural network is implemented to capture long-term dependencies in stock price movements. The model is trained using sequences of historical OHLCV data and optimized using backpropagation through time.

Several approaches were experimented with using the available dataset. Multiple training iterations were conducted by grouping all stocks together, training individual stocks separately, and organizing them sector-wise. Limiting the dataset to the most recent two years further enhanced the model's accuracy. Additionally, feature selection techniques were applied based on performance results, leading to the identification and use of only the most relevant features for prediction.

Phase 2: Hybrid and Transformer Models
To enhance performance, advanced models such as CNN-LSTM, GRU, and Transformer architectures with attention mechanisms are implemented. CNN layers extract local temporal features, while recurrent and attention-based layers capture sequential dependencies and global patterns.

Training Strategy

- Sliding window technique for sequence generation
- Train-validation-test split
- Hyperparameter tuning (epochs, batch size, learning rate)
- Early stopping to prevent overfitting

5.4 Evaluation Metrics

The performance of the proposed stock price prediction models is evaluated using standard regression-based evaluation metrics. These metrics quantify the accuracy of predicted stock prices by comparing them with actual observed values.

5.4.1 Mean Squared Error (MSE)

Mean Squared Error (MSE) penalizes larger errors by squaring the differences between predicted and actual values.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{5.1}$$

5.4.2 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) represents the square root of MSE and provides the error magnitude in the same units as the predicted variable.

$$RMSE = \sqrt{MSE} \tag{5.2}$$

5.4.3 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average magnitude of errors between predicted and actual values without considering their direction.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.3}$$

5.4.4 Coefficient of Determination (R^2)

The coefficient of determination (R^2) measures the proportion of variance in the dependent variable that is predictable from the independent variables.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5.4}$$

where $\bar{y}$ represents the mean of the actual values.

Chapter 6

Results and Discussion

6.1 Result Analysis

6.1.1 Individual Share Result (VEDL)

Trials	MSE	RMSE	MAE	R^2
Trial 1	3842.42	61.98	46.62	0.73
Trial 2	2142.04	46.28	39.64	0.75
Trial 3	1110.99	33.33	24.10	0.92

Table 6.1: Model Performance Evaluation

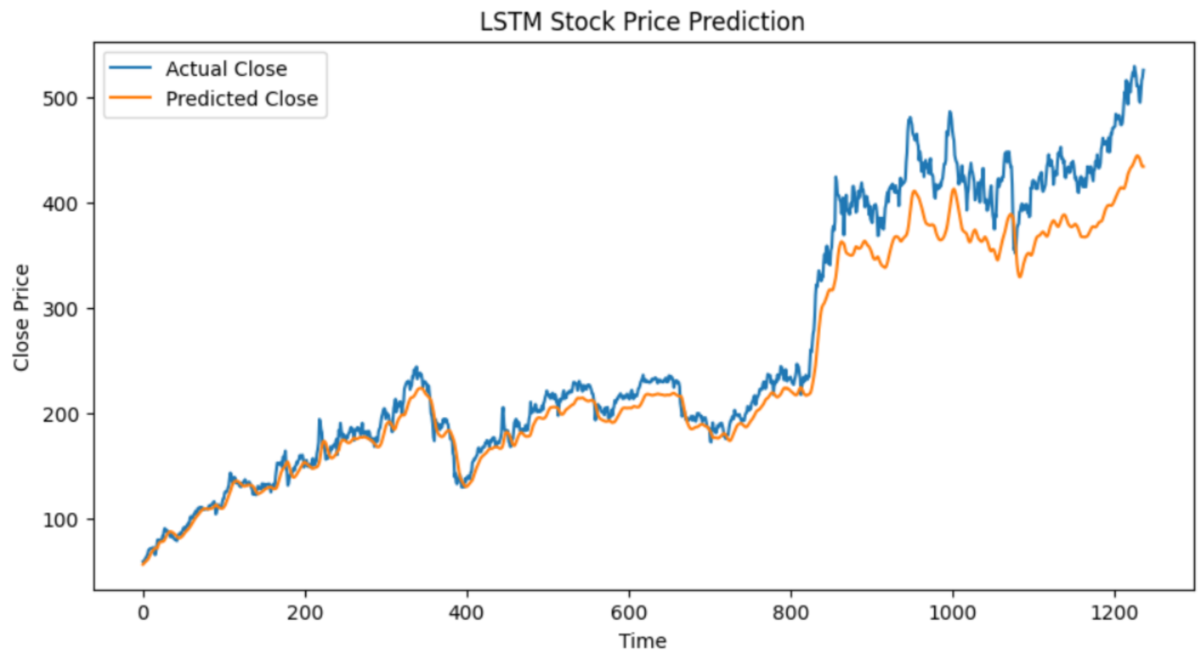

Figure 6.1: Individual Stock Prediction (VEDL)

6.1.2 Sector Wise Result

Sectors	MSE	RMSE	MAE	R^2
Basic Materials	54127.5815	232.6534	134.548	0.9744
Communication Services	39820.3549	199.5504	82.8376	0.9385
Consumer Cyclical	116131002.9	10776.4096	1867.6738	0.5554
Consumer Defensive	37775.4938	194.3592	179.4476	0.816
Energy	1851.8641	43.0333	29.2189	0.9646
Financial Services	33951.4956	184.2593	161.8281	0.9608
Healthcare	161519.4446	401.8948	116.9218	0.9718
Industrials	710198.3989	842.7327	801.3894	0.8177
Real Estate	2991.2171	54.692	37.5018	0.9877
Technology	2429128.579	1558.5662	510.5046	0.7563
Utilities	13956.3	118.1368	69.6904	0.9407

Table 6.2: Model Performance Evaluation

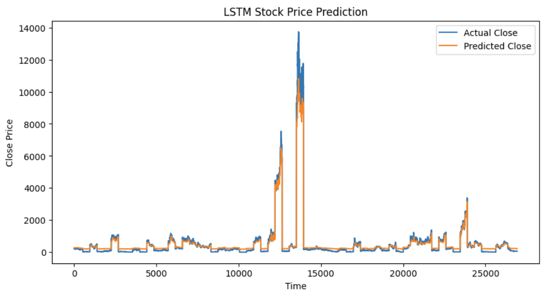

Figure 6.2: Basic Materials

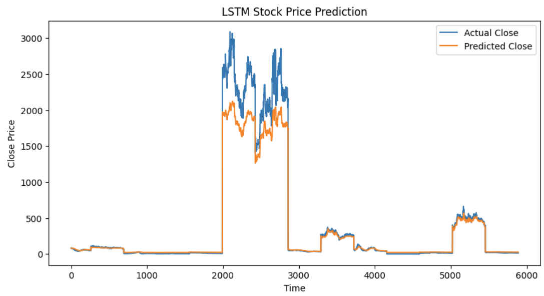

Figure 6.3: Communication Services

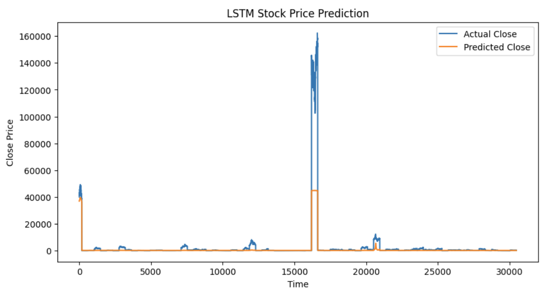

Figure 6.4: Consumer Cyclical

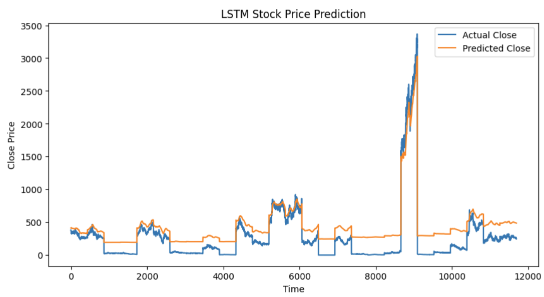

Figure 6.5: Consumer Defensive

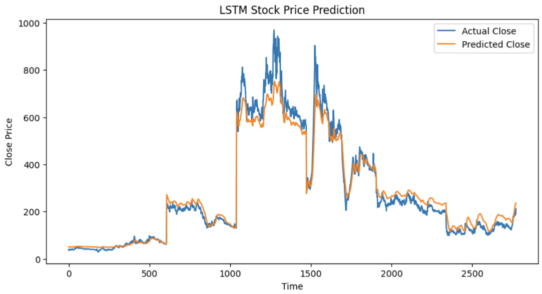

Figure 6.6: Energy

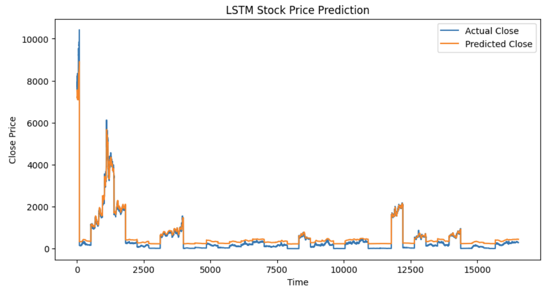

Figure 6.7: Financial Services

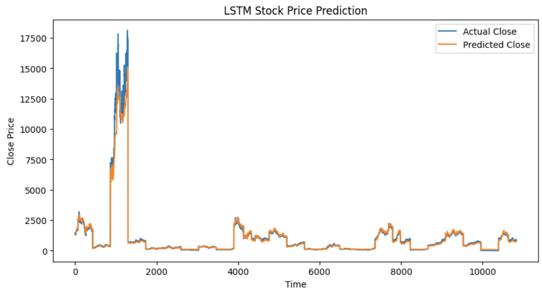

Figure 6.8: Healthcare

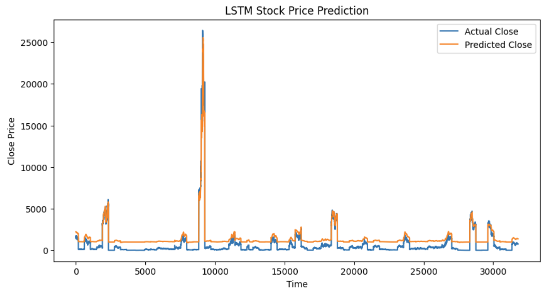

Figure 6.9: Industrials

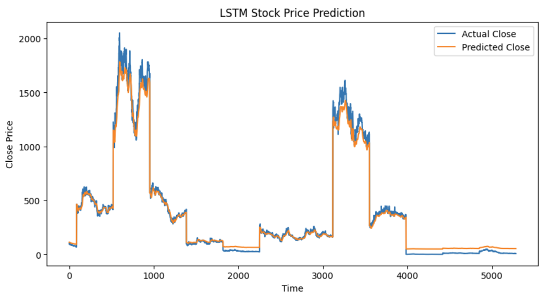

Figure 6.10: Real Estate

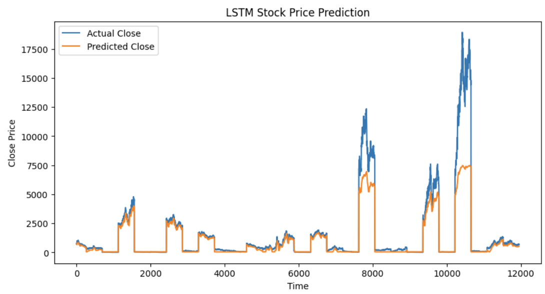

Figure 6.11: Technology

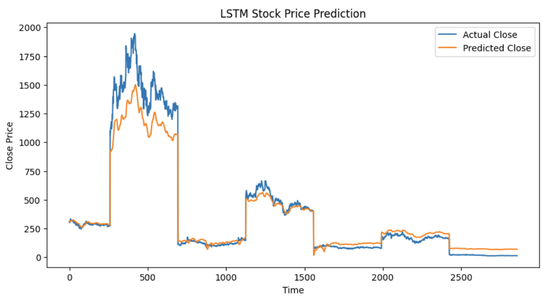

Figure 6.12: Utilities

Chapter 7

Conclusion and Future Work

This chapter summarizes the outcomes of the project, the conclusions drawn, and suggests directions for future work.

7.1 Conclusion

7.2 Future Work