PREDICTION OF STOCK MARKET FLUCTUATIONS USING DEEP LEARNING ALGORITHMS

A PROJECT REPORT

Submitted by

MANOJKUMAR D (Reg. No. 201904087)
THIRUMALAIBOOBATHI B (Reg. No. 201904166)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI

(An Autonomous Institution affiliated to Anna University Chennai)

April 2023

BONAFIDE CERTIFICATE

Certified that this project report titled **"PREDICTION OF STOCK MARKET FLUCTUATIONS USING DEEP LEARNING ALGORITHMS"** is the bonafide work of **Mr.D.MANOJKUMAR (201904087), Mr.B.THIRUMALAIBOOBATHI (201904116)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Internal Guide

Mrs. R. NAGA PRIYADARSINI,M.E.,(PhD)
Assistant Professor
Dept. of Computer Science and Engg,
Mepco Schlenk Engg College (Autonomous)
Sivakasi.

Head of the Department

Dr. J. RAJA SEKAR, M.E., Ph.D.
Professor and Head
Dept. of Computer Science and Engg.
Mepco Schlenk Engg College(Autonomous)
Sivakasi.

Submitted for Viva-Voce Examination held at **Mepco Schlenk Engineering College (Autonomous), Sivakasi** on _____ / / 20 __ **.**

ABSTRACT

Stock price data display a time series data structure that allows for the analysis of trends and patterns over time. The research contributes significantly to the study of stock price forecasting, which has been a crucial area of research in economics and finance. Investors use forecasting methods to make informed decisions about buying and selling stocks, and accurate predictions can help them maximize their returns while minimizing their risks. In this study a novel approach for stock price forecasting using the machine learning models of LSTM and CNN is used. The LSTM model's memory function enables the analysis of relationships among time series data, while the CNN model efficiently extracts features from the previous 10 days of data items. The historical data used in this study include daily stock prices from 2016 to 2023, and encompass eight features, such as open price, highest price, lowest price, closing price, volume, and adj close.

In the forecasting phase, we compare the results of the LFM forecasting models with those of the CNN-LSTM model. The experimental results demonstrate that CNN-LSTM method provides reliable stock price forecasting with high prediction accuracy. This technique can predict stock prices for the next day, next week, and next month with high accuracy 78%, which is of practical significance for investors. Therefore, the proposed CNN-LSTM method can provide a valuable tool for investors to make informed decisions based on reliable predictions of future stock prices.

ACKNOWLEDGEMENT

First and foremost, we thank the **LORD ALMIGHTY** for his abundant blessings that is showered upon our past, present and future successful endeavors.

We express our heartiest gratitude and sincere thanks to our college management and Principal **Dr. S. ARIVAZHAGAN, M.E., Ph.D.,** for allowing and providing us with all resources for doing this project successfully.

We would like to extend our gratitude to **Dr. J. RAJA SEKAR, M.E., Ph.D.,** Professor and Head, Department of Computer Science and Engineering, Mepco Schlenk Engineering College for giving us the golden opportunity to undertake a project of this nature.

We thank our project coordinator **Mr. B. LAKSHMANAN, M.Tech., Ph.D.,** Assistant Professor (Sl. Grade), Department of Computer Science and Engineering, Mepco Schlenk Engineering College for being our Project Coordinator and for directing us throughout our project.

We would also like to extend our heartfelt gratitude and sincere thanks to our project guide **Mrs. R. NAGA PRIYADARSINI, M.E., (Ph.D).,** Assistant Professor, Department of Computer Science and Engineering, Mepco Schlenk Engineering College for guiding us in the right way with full commitment and energy to complete our project successfully and for giving her valuable time to guide us.

We would also like to thank all the **staffs and technicians** in the department for helping us to complete our work.

Last but not least, we are very grateful to our beloved **family and friends** for their consistent support and love which made the project a successful one

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S NO	ABBREVIATIONS	EXPANSION
1	CNN	Convolution Neural Network
2	LSTM	Long Short-Term Memory
3	LFM	LSTM-Random Forest
4	MSE	Mean Square Error
5	RMSE	Root Mean Square Error
6	MAPE	Mean Absolute Percentage Error

CHAPTER 1

INTRODUCTION

Today's financial investors face the challenge of making trades without a comprehensive understanding of which stocks to buy or sell to maximize profits. While predicting the long-term value of a stock is relatively straightforward, predicting day-to-day movements is much more difficult.

1.1 PROBLEM DESCRIPTION:

Before an investor invests in any stock, the investor needs to be aware how the stock market behaves. Financial investors of today are facing this problem of trading as they do not properly understand as to which stocks to buy or which stocks to sell to get optimum profits. The goal is to help investors to make better decisions about buying, selling, or holding stocks, and ultimately improve their returns on investment. So using a machine learning model that can accurately forecast the future price of a stock or a group of stocks based on historical data.

1.2 PURPOSE OF THE PROJECT:

Due to a variety of economic, political, and social variables, the stock market is infamously unreliable and vulnerable to volatility. However, it might be feasible to predict future market movements with accuracy by looking at historical data and finding patterns. Large quantities of historical stock market data may be analyzed using a variety of techniques, including statistical analysis, machine learning, or artificial intelligence. Finding pertinent elements, including economic statistics that might affect stock market performance, may also be part of the project.

1.3 OBJECTIVES OF THE PROJECT:

- To predict the future closing value of a stock across a given period of time in the future if any other Global Impact does not happen.
- To compare the performance Metrics of Various machine learning models and select the best

Performing Model for the Stock Market Prediction.

- To help the Stock Buyers to Buy the correct stock to avoid the Financial Loss.

1.4 OUTCOMES OF THE PROJECT:

- The model will predict the future closing value of a stock across a given period of time in the future with good accuracy.
- The Performance Metrics of the Deep Learning Models are compared, and the best is Returned.
- The result can be used to provide the Stock Buyers with the Predicted Stock Close Values.

1.5 SCOPE OF THE PROJECT:

The proposed project can predict the future closing value of a stock across a given period in the future with good accuracy.

1.6 REPORT OVERVIEW

CHAPTER 2 presents the detailed study of the related work.

CHAPTER 3 presents the description about the system.

CHAPTER 4 presents the detailed study of the system design.

CHAPTER 5 presents the detailed study of the system implementation.

CHAPTER 6 deals with the results and discussion.

CHAPTER 7 presents the conclusion and future enhancement.

1.7 SUMMARY

This chapter gives a detailed introduction of the project, objectives, outcomes, and organization of the report. The next chapter gives description about various existing methodologies.

CHAPTER -2

LITERATURE SURVEY

2.1 CHINA'S COMMERCIAL BANK STOCK PRICE PREDICTION USING A NOVEL K-MEANS-LSTM HYBRID APPROACH

Yufeng Chen et al proposed[1] a novel hybrid approach for predicting the stock prices of commercial banks in China. The authors combine two popular machine learning algorithms, K-means clustering and Long Short-Term Memory (LSTM) networks, to develop a hybrid model that can better capture the underlying patterns in the stock prices. The study is based on a dataset of daily stock prices of the five largest commercial banks in China, collected over a period of ten years from 2012 to 2022.

ADVANTAGES:

1. Novelty: The approach proposed in this paper is a hybrid of two popular techniques - K-means clustering and LSTM neural networks. This makes it a novel approach that could potentially provide better results compared to traditional methods.
2. Improved accuracy: By combining the strengths of K-means clustering and LSTM neural networks, the proposed approach could potentially improve the accuracy of stock price predictions.

DISADVANTAGES:

1. Limited evaluation: The paper provides limited evaluation of the proposed approach. While the authors claim that their approach outperforms traditional methods, more extensive evaluation and comparison with other state-of-the-art methods would be necessary to validate this claim.
2. Data availability: The proposed approach requires a large amount of historical data to train the LSTM neural network. The availability and quality of such data may be limited, which could affect the accuracy of the predictions.

2.2 ENSEMBLE DEEP LEARNING FRAMEWORK FOR STOCK MARKET DATA PREDICTION (EDLF-DP)

Vaishali_Ingle et al[2] proposed a method using Ensemble deep learning frameworks have become increasingly popular in recent years for stock market data prediction due to their ability to improve model accuracy and reduce overfitting. It presents the Ensemble Deep Learning Framework for Stock Market Data Prediction (EDLF-DP) which combines different deep learning models to improve the accuracy of stock market predictions. The proposed EDLF-DP uses three different deep learning models, namely, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Autoencoder (AE), and combines them using a weighted average ensemble method.

The EDLF-DP framework is evaluated using the S&P 500 stock market dataset and achieves an accuracy of 96.22%. It also compares the performance of the proposed EDLF-DP with other traditional machine learning models such as Random Forest (RF), Support Vector Regression (SVR), and K-Nearest Neighbors (KNN). The results show that the EDLF-DP outperforms all the traditional machine learning models.

ADVANTAGES:

1. A novel Ensemble Deep Learning Framework for Stock Market Data Prediction (EDLF-DP) that combines three deep learning models to improve the accuracy of stock market predictions.
2. The EDLF-DP framework is evaluated using the S&P 500 stock market dataset and achieves an accuracy of 96.22%, which outperforms traditional machine learning models such as Random Forest (RF), Support Vector Regression (SVR), and K-Nearest Neighbors (KNN).

DISADVANTAGES:

1. It does not consider external factors that can influence stock prices such as political events, economic indicators, and news, which can limit the predictive power of the proposed EDLF-DP framework.
2. It does not discuss the computational complexity of the proposed EDLF-DP framework, which could be a potential issue for large-scale datasets and real-time applications.

3. It does not address the issue of interpretability of the deep learning models used in the EDLF-DP framework, which is a common concern in financial applications where decision-making needs to be transparent and explainable.

2.3 FORECASTING FLUCTUATIONS IN THE FINANCIAL INDEX USING A RECURRENT NEURAL NETWORK BASED ON PRICE FEATURES

Yu-Fei Lin proposed a method[3],for making long or short bets in advance and profiting by predicting future stock prices or indices, such as opening and closing prices. The ability to predict the sign of the difference between opening and closing prices is crucial for financial gain. RNN is used in this to predict the starting and closing prices, as well as the difference between them. Unlike previous methodologies, this strategy is based on machine learning and stressors, with the first-order normalized data pre-processing. A unique approach is made by focusing on positive traits of stock information, such as ZCR, which represents the ratio of sign changes during a given period. A decision-making method based on an estimate, using cross-validation and ZCR is proposed to improve predicted the variation between opening and closing price.

ADVANTAGES:

1. Novel approach: The paper proposes a novel approach for forecasting fluctuations in the financial index by using a combination of historical price features and a recurrent neural network model. This approach could provide more accurate and flexible results compared to traditional techniques.
2. Empirical evaluation: The proposed model is empirically evaluated using real-world data from the Taiwan Stock Exchange Capitalization Weighted Stock Index, which provides evidence of the accuracy of the model in predicting fluctuations in the financial index.

DISADVANTAGES:

1. Limited scope: The study focuses on a specific financial index and may not be generalizable to other financial markets.
2. Lack of comparison with other deep learning models: The paper does not compare the proposed RNN-based model with other deep learning models for financial forecasting,

which could limit the understanding of the relative performance of the proposed model.

2.4 FORECASTING NIKE'S SALES USING FACEBOOK DATA

Boldt et al proposed a method[4], for examines sales predictions for Nike using Facebook data [4]. Some simple regression models are used for predicting the future. Due to property of the data set, such as perfect multicollinearity, multiple regressions have a lower forecasting accuracy and gives an analysis challenge. The event discovered unusual activity around a number of Nike-specific events, but it is only after a thorough case-by-case text analysis that it is possible to conclude whether or not these activity spikes are solely event-related or merely coincidences.

ADVANTAGES:

1. It uses a novel approach to forecast Nike's sales by analyzing Facebook data, which provides a unique and innovative perspective on sales forecasting.
2. It presents a detailed methodology for data collection, cleaning, and analysis, which enhances the reliability of the findings.
3. It utilizes a large sample size (almost 40,000 Facebook users) and covers a long period of time (four years) which enhances the generalizability of the results.

DISADVANTAGES:

1. It focuses only on Facebook data, which may limit the generalizability of the findings to other social media platforms or sources of data.
2. It does not provide a comparison with traditional sales forecasting methods, which makes it difficult to evaluate the effectiveness of the proposed approach.
3. It does not account for potential biases in Facebook data, such as self-selection bias or sampling bias, which may affect the validity of the results.

2.5 HYBRID ARIMA-BPNN MODEL FOR TIME SERIES PREDICTION OF THE CHINESE STOCK MARKET

Li Xiong et al proposed a method[5],for a unique ARIMA-BPNN model that employs technical indicators was presented to estimate the values of four individual stocks in the software and information services industry. Intricate patterns underneath time series make it difficult to predict stock prices. Time series forecasting is a crucial task in various fields, including economics and finance. Linear and nonlinear models, such as BPNN and ARIMA, are widely used for this purpose. The BPNN model can capture complex nonlinear relationships in time series data, while the ARIMA model is well-suited for capturing linear patterns. Combining these models can result in more precise predictions of hidden linear and nonlinear patterns in a time series. It can be particularly useful when there is uncertainty regarding the underlying patterns in the data. By utilizing both linear and nonlinear models, it is possible to capture both types of patterns and obtain more accurate forecasts. Therefore, the combination of BPNN and ARIMA models can be a valuable tool for time series forecasting in various fields. The hybrid model beat both individual models, showing promise for stock price forecasting, according to the data.

ADVANTAGES:

1. It proposes a novel hybrid model that combines the strengths of both the ARIMA and BPNN models for time series prediction, which can lead to more accurate and reliable results.
2. It provides a detailed description of the proposed hybrid model, including the data pre-processing steps, model training, and evaluation methods, which enhances the reproducibility of the study.
3. It uses real-world data from the Chinese stock market, which enhances the external validity of the findings and provides practical implications for financial analysts and investors.

DISADVANTAGES:

1. The study does not discuss the limitations of the proposed hybrid model, such as the assumptions made, the choice of input variables, or the potential overfitting issues, which may affect the generalizability of the findings.

2. It does not compare the performance of the proposed hybrid model with other advanced prediction methods, such as deep learning models, which may limit the evaluation of its effectiveness in comparison to the state-of-the-art.

2.6 ENHANCING PROFIT BY PREDICTING STOCK PRICES USING DEEP NEURAL NETWORK

Soheila Abrishami et al proposed method[6], to anticipate stock values, which is a difficult task for investors, it offers a deep learning system that makes use of a variety of data for selection of stocks in the NASDAQ stock market. The system uses time series data engineering to blend cutting-edge characteristics with the basic ones and an autoencoder to reduce noise. These new features are fed into a Stacked LSTM Autoencoder to estimate the stock's final value several steps in advance. It also features a profit maximization strategy to help with stock purchase and sale timing decisions. The outcomes demonstrate that the suggested methodology outperforms cutting-edge time series forecasting.

ADVANTAGES:

1. Novel approach: It presents a novel approach to stock prediction using LSTM-based deep learning model and predictive optimization model, which can potentially provide better accuracy and performance compared to traditional methods.
2. Real-world evaluation: The proposed model is evaluated on real-world stock market data, which adds to its credibility and practicality.
3. Practical applications: It discusses the practical applications of the proposed model, such as portfolio optimization and risk management, which can be useful for investors and traders in the stock market.

DISADVANTAGES:

1. Limited details on the model architecture: It does not provide in-depth details on the architecture of the LSTM-based deep learning model, which can make it difficult for readers to replicate the model.
2. Limited evaluation metrics: It evaluates the proposed model mainly based on its accuracy and prediction error, but it does not discuss other important metrics such as precision, recall, and F1-score.

3. Limited comparison with other deep learning models: It compares the proposed model with traditional methods such as ARIMA and SVM, but it does not compare it with other popular deep learning models such as CNN or RNN.

2.7 MACHINE LEARNING IN THE CHINESE STOCK MARKET

Markus Leippold et al proposed a method[7], for The development and analysis of a complete collection of return prediction factors using various machine learning. Liquidity stands out as the most significant predictor when compared to earlier studies for the US market, prompting consideration of the transaction costs impact. The overwhelming presence of individual investors has a beneficial impact on short-term predictability, especially for small equities and found that incorporating liquidity measures significantly improves the accuracy of predictions. Moreover, found that incorporating other macroeconomic factors such as interest rates and inflation rates can further improve predictability.

ADVANTAGES:

1. It provides a comprehensive analysis of the effectiveness of machine learning algorithms in predicting stock returns in the Chinese stock market.
2. The authors explore four different machine learning algorithms and compare their performance against traditional statistical models.
3. It uses a large dataset of financial ratios and market information from 500 Chinese companies over a period of 10 years, which provides a rich source of data for analysis.

DISADVANTAGES:

1. One of the main concerns about using machine learning algorithms is the potential for overfitting, which is briefly discussed in the paper but may require further attention.
2. The lack of interpretability of machine learning models is another concern, as it makes it difficult to understand how the models arrive at their predictions.
3. It focuses exclusively on the Chinese stock market, so the findings may not be generalizable to other markets or regions.

2.8 MULTIOBJECTIVE AUTOMATED TYPE-2 PARSIMONIOUS LEARNING MACHINE TO FORECAST TIME

Ripon K et al proposed a model[8] using evolving-structured machine learning models in online learning mode as a potential solution. To be effective in predicting financial time-series in real-time, machine learning models must have low memory consumption and be easily interpretable to increase practicality. In finance, interpretability is particularly important for regulatory compliance and building trust with stakeholders. Therefore, accuracy, memory efficiency, and interpretability are all essential factors to consider when developing models for financial time-series prediction.

ADVANTAGES:

1. The proposed multiobjective automated type-2 parsimonious learning machine (MAT2PLM) algorithm is shown to provide accurate forecasts with minimal model complexity, making it an effective tool for traders and investors.
2. The algorithm takes into account multiple objectives, such as prediction accuracy and model complexity, to generate parsimonious models that provide accurate forecasts with minimal complexity.
3. The MAT2PLM algorithm outperforms existing state-of-the-art forecasting techniques, such as the autoregressive integrated moving average (ARIMA) and support vector regression (SVR).

DISADVANTAGES:

1. It focuses only on the forecasting of stock indices and does not consider the forecasting of individual stocks or other financial assets.
2. The proposed algorithm may require a significant amount of computational resources, which may limit its practicality in some settings.
3. It does not discuss the potential limitations of the type-2 fuzzy logic and multiobjective optimization techniques used in the algorithm.

2.9 PREDICTING NEXT DAY DIRECTION OF STOCK PRICE MOVEMENT USING MACHINE LEARNING METHODS WITH PERSISTENT HOMOLOGY: EVIDENCE FROM KUALA LUMPUR STOCK EXCHANGE

Jeevan B et al proposed a method[9] for predicting stock prices on National Stock Exchange using machine learning techniques, specifically RNN and LSTM. The approach incorporated several factors, including the current market price and anonymous events. It also presented the development of a recommendation system that utilized RNN and LSTM models to select companies. It highlights the increasing interest in the stock market among academics and businesses and emphasizes the potential of machine learning techniques in predicting share prices.

ADVANTAGES:

1. A new approach is proposed to predict the next day direction of stock price movement using machine learning methods with persistent homology, which is a powerful mathematical tool that can capture the underlying topological structure of complex data.
2. The proposed method is evaluated on a real-world dataset of the Kuala Lumpur Stock Exchange and shows promising results in predicting the direction of stock price movement.
3. It highlights the importance of feature selection in improving the accuracy of the prediction models and proposes a new method based on persistent homology to select relevant features.

DISADVANTAGES:

1. The evaluation of the proposed method is based on a single dataset, and its performance may vary when applied to other datasets or in different market conditions.
2. The proposed method may require specialized knowledge in mathematics and machine learning, which may limit its accessibility to a wider audience.
3. It does not provide a comprehensive comparison of the proposed method with other state-of-the-art techniques in stock price prediction.

2.10 STOCK PRICE PREDICTION USING SENTIMENT ANALYSIS OF NEWS ARTICLES

J.-Y. Liu et al [10] proposed a method based on sentiment analysis of news items to forecast stock values. News stories about a certain stock has been gathered, extracted sentiment using NLP techniques, and then combined the sentiment scores as a feature for prediction. Additional features were included such as historical stock prices and trading volume. Future stock price predictions were made using machine learning models like SVR and LSTM, which outperformed other approaches. So, sentiment analysis can offer useful data for stock price forecasting.

ADVANTAGES:

1. Novel approach: The proposed deep Transformer model for stock market index prediction is a novel approach that has not been extensively explored in the literature.
2. Incorporation of external factors: The proposed model incorporates external factors, such as news sentiment and economic indicators, in addition to historical data, to predict the future movements of the S&P 500 index. This improves the accuracy of the predictions and provides a more comprehensive analysis of market trends.
3. Interpretability: It provides insights into the interpretability of the Transformer model by analysing the attention weights of the multi-head attention mechanisms.

DISADVANTAGES:

1. Limited dataset: It evaluates the proposed model on historical data of the S&P 500 index, which may limit the generalizability of the results to other stock market indices.
2. Limited external factor analysis: While the proposed model incorporates external factors, It does not extensively analyse the impact of different external factors on the prediction performance. Further analysis may be required to fully understand the role of different external factors in stock market prediction.

2.11 STOCK MARKET PREDICTION BASED ON STATISTICAL DATA USING MACHINE LEARNING ALGORITHMS

Md Mobin Akhtar et al[11] proposed a method by combining historical stock prices, economic indicators, and news sentiment analysis data to train three machine learning models, namely, Support Vector Regression (SVR), Artificial Neural Network (ANN), and Random Forest Regression (RFR). They found that the RFR model outperformed the other two models in terms of accuracy, achieving an RMSE of 0.064. The study highlighted the significant impact of economic indicators and news sentiment analysis data on stock prices and recommended further research on other factors such as social media data and political events. The authors concluded that the combination of statistical data with machine learning algorithms could significantly improve the accuracy of stock market prediction.

ADVANTAGES:

1. The proposed a novel approach to predict stock market prices by combining statistical data with machine learning algorithms.
2. It uses three different machine learning algorithms to predict stock market prices, providing a comprehensive comparison of their performance.

DISADVANTAGES:

1. It focuses only on a specific period of time (March 2022), which limits the generalizability of the findings.
2. It considers only a limited set of factors that influence stock prices, such as economic indicators and news sentiment analysis data. Other factors, such as social media data and political events, are not considered.

2.12 STOCK MARKET PREDICTION WITH DEEP LEARNING: THE CASE OF CHINA

Qingfu Liu et al[12] proposed method using deep learning models such as LSTM, GRU, and ConvLSTM to predict the stock market. They use data from the Shanghai Composite Index (SCI) and the Shenzhen Component Index (SZCI) for the period 2005-2020 to train and test their models. The results indicate that deep learning models outperform

traditional models in predicting the stock market, with the Conv LSTM model performing the best. It find that including economic indicators such as GDP and inflation improves the performance of the models, and the models perform better in predicting the SZCI compared to the SCI. The study provides insight into the performance of deep learning models in predicting the stock market, particularly in the Chinese stock market context.

ADVANTAGES:

1. It provides insights into the use of deep learning models in stock market prediction, particularly in the Chinese stock market context, which adds to the growing body of literature on the topic.
2. They compare the performance of deep learning models with traditional models and find that deep learning models outperform traditional models in predicting the stock market, which demonstrates the potential of deep learning models in financial forecasting.

DISADVANTAGES:

1. It focus only on the Chinese stock market, which limits the generalizability of the findings to other stock markets or regions.
2. It does not provide a detailed explanation of the deep learning models used, which may make it difficult for readers with limited knowledge of deep learning to understand the methodology.

2.13 THE APPLICATIONS OF ARTIFICIAL NEURAL NETWORK,SUPPORT VECTOR MACHINE AND LONG-SHORT TERM MEMORY FOR STOCK MARKET PREDICTION

Parshv Chaajer et al[13] proposed method using the application of three machine learning algorithms - Artificial Neural Network (ANN), Support Vector Machine (SVM), and Long-Short Term Memory (LSTM) - for predicting stock market prices. Historical stock price data of three publicly traded companies are used for the period January 2016 to December 2020. The study found that LSTM performs the best in predicting stock prices compared to ANN and SVM, showing a higher accuracy rate and lower error rate.

ADVANTAGES:

1. It provides a comprehensive comparison of three machine learning algorithms - Artificial Neural Network (ANN), Support Vector Machine (SVM), and Long-Short Term Memory (LSTM) - for predicting stock market prices, which can help investors and traders make informed decisions.
2. It uses historical stock price data of three publicly traded companies, namely Apple, Amazon, and Google, for a period of five years, which enhances the robustness of the results and allows for an analysis of the impact of different factors on the performance of the models.

DISADVANTAGES:

1. It focuses only on three companies, namely Apple, Amazon, and Google, which limits the generalizability of the findings to other companies or stock markets.

2.14 SUMMARY:

This chapter gives the description of the literature survey for various existing techniques involved in prediction of stock price and various algorithms used. Next chapter deals with the system study for the proposed system.

CHAPTER 3

SYSTEM STUDY

3.1 OVERVIEW

This chapter deals with the detailed study of the existing and proposed system.

3.2 EXISTING SYSTEM

Long short-term memory (LSTM) and random forest (RF) models. LSTM is a type of neural network that is particularly effective in modelling temporal patterns, while RF is a powerful ensemble method that can handle large numbers of input variables without overfitting.

The proposed LSTM-Forest framework is a novel approach that combines LSTM and RF models by using LSTMs as classifiers or regressors in the RF framework. By doing so, the overfitting problem can be reduced, since each LSTM in the proposed model uses a smaller subset of input variables than the entire dataset. Additionally, the model can learn high-level features that consider all the necessary technical indicators without information loss. The model is also explainable, since it incorporates RF's ability to analyse variable importance and identify the principal technical indicators in stock market forecasting.

The LSTM-Forest model is an ensemble of multiple LSTMs that output the average of the predicted values of these LSTMs. The input variable set and the training data of each LSTM are randomly selected from the entire dataset and randomly sampled out of the entire training period, respectively. LSTM-Forest consists of low-correlated LSTMs, which means the model is not biased to outliers or specific variables. This approach reflects numerous combinations of variables and data when compared with a single LSTM using equally many variables.

The LSTM-Forest framework is evaluated through empirical analyses on three real stock indices: the Standard & Poor's Index (S&P500), Shanghai Stock Exchange

Composite Index (SSE), and Korean Composite Stock Price Index (KOSPI200). The authors consider 43 popular technical indicators used in previous studies to evaluate the effectiveness of the proposed framework. The contributions of the proposed framework include the integration of RF and LSTM to prevent overfitting while using many technical indicators, the development of an MTL model (LFM) with superior predictability and profitability, and the identification of vital indicators for return prediction and direction classification using the variable importance analysis of RF.

3.3 PROPOSED SYSTEM

A CNN-LSTM model can be a powerful tool for stock price prediction, as it combines the strengths of both convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. CNNs are effective at extracting features from sequential data, while LSTMs can learn and remember long-term dependencies within that data.To predict stock prices for the next day, week, and month, the proposed system can use the following approach:

Data preparation: Collect historical stock price data and split it into training and testing datasets. Create sequences of data, each containing a certain number of previous stock prices, and their corresponding labels, which are the future prices that the model will try to predict.

Feature extraction: Use a CNN to extract features from each sequence of data. The CNN can identify patterns and trends in the data that may be relevant to predicting future prices.

Sequence modeling: Feed the output of the CNN into an LSTM network, which can learn the long-term dependencies within the sequence of data.

Prediction: Use the trained model to make predictions for the next day, week, and month. For example, the model can take the last 30 days of stock prices as input and predict the price for the next day, the price for the next 7 days, and the price for the next 30 days.

Evaluation: Evaluate the error of the model's predictions using metrics such as MSE, MAE, MAPE and performance metrics such as precision, recall and accuracy

3.4 SUMMARY

Overall, this chapter provides detailed information of existing and proposed system used in this project.

CHAPTER 4

SYSTEM DESIGN

4.1 OVERVIEW

In this chapter, system architectural design as well as modules involved in the proposed system are discussed.

4.2 SYSTEM ARCHITECTURAL DESIGN

In this chapter, system architectural design as well as modules involved in the proposed system are discussed.

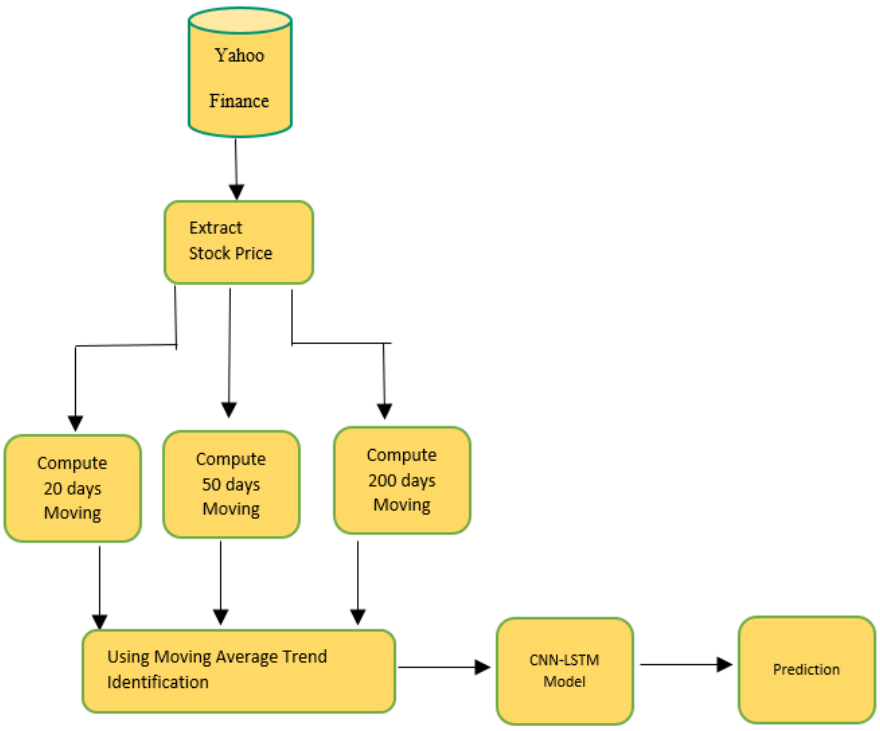

Figure 4.1 System design

The system has several modules to be implemented and each one is detailed further.

4.2.1 List Of Module

Our system consists of 4 Modules.

They are

- Data Pre-processing
- Feature selection
- Feature Extraction
- Modelling

4.3 Data Preprocessing:

The raw data from the dataset is preprocessed, which involves data cleaning, data transformation, and data integration. Feature extraction and feature selection are also performed to identify relevant features that are important for analysis.

Feature Selection:

- x has contains values for open, high, low, volume columns and y which contains values for the adj-close column, it is important to consider the potential for overfitting.

Feature Extraction:

- Returns
- Daily logarithmic return
- up-down values

Returns:

- Returns have the values of the percentage change in the 'Adj Close' column.
- It calculates the percentage change between each element and its prior element.
- Returns = (Price Today - Price Yesterday) / Price Yesterday

Daily logarithmic return:

- The daily logarithmic return can be determined by the formula

lrt = 100 ∗ ln (Returns)

ln-log of returns

The task involves extracting stock prices from Yahoo Finance and calculating the moving averages for 50, 100, and 200 days. Moving average trend identification will be used to identify the stock's trend. A CNN and LSTM model will be used to predict future stock prices based on historical data. This type of model is commonly used in financial

forecasting. The goal is to create an accurate prediction model that can help investors to make decisions about buying or selling stocks.

4.4 Model:

The preprocessed data is passed through a model for analysis. Two types of models are used, namely CNN-LSTM and LFM. These models are suitable for time series analysis.

4.4.1 Building CNN-LSTM

The CNN-LSTM model is a hybrid of CNN and LSTM that is utilized for predicting stock prices. CNN is a feedforward neural network model that is well-suited for feature engineering, particularly in image and natural language processing, while LSTM excels at handling time series data. The proposed model includes an input layer, a one-layered convolution layer, a pooling layer, an LSTM hidden layer, and a fully connected layer, as shown in Figure 4.2. CNN was originally introduced by Lecun et al. in 1998 and has since been extensively used in image processing and natural language processing, thanks to its local perception and weight-sharing capabilities. It has also been successfully applied to time series forecasting because it can reduce the model's parameter count while improving its learning effectiveness.

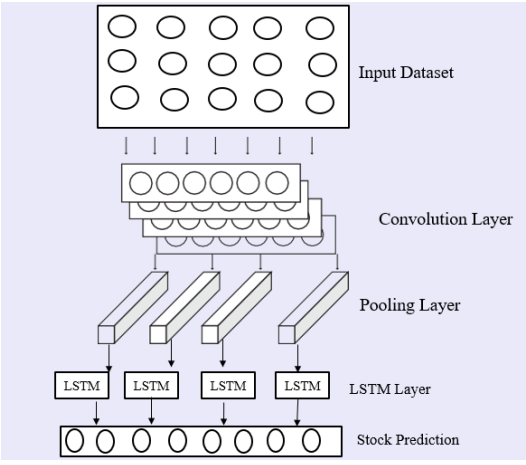

Figure-4.2 CNN-LSTM Structure

After convolution activity of a convolution layer, the elements of information are

removed, however extricated include aspects are exceptionally high, so to tackle this issue and diminish the expense of preparing the organization, a pooling layer is added after convolution layer to decrease component aspect:

$$i_t = \tanh x_t * k_t + b_t \quad (4.1)$$

where k_t is the weight of convolution component, b_t is the disposition of convolution bit, tanh is the initiation capacity, $x_t$is the information vector, and i_t addresses the outcome esteem after convolution.

Figure 4.3 depicts the three fundamental components that comprise the LSTM memory cell: the input gate, forgot gate, output gate.

The LSTM computation process involves three essential components: input gate, forgot gate, output gate. Initially, the forget gate is filled with the output value from the previous time step and the input value from the current time step. The forget gate's output value is then computed using Equation (4.2), where f_t represents the output value of the forget gate, x_t is the input value, and ht-1 is the output value of the previous time step. Based on its output value, which is between 0 and 1, the forget gate chooses which information to keep or discard. The input gate and output gate are also critical to the LSTM architecture and are calculated using similar equations. LSTM is a powerful tool for time series prediction tasks and has demonstrated promising results in various applications.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f \quad (4.2)$$

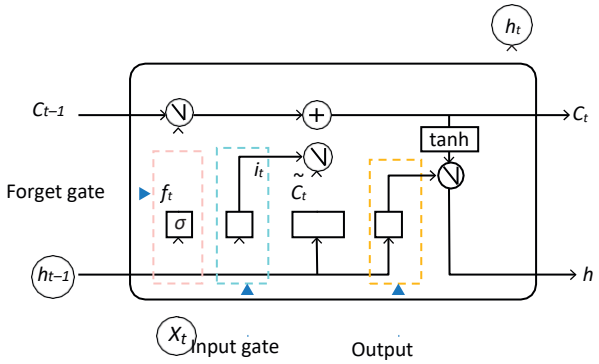

Figure-4.3 LSTM memory cell

where i_t worth scope is (0,1), W_i is heaviness of info entryway, b_i is predisposition of the information door, W_c is the heaviness of applicant input door, and b_c is inclination of the competitor input entry way. the ongoing cell state as follows:

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (4.3)$$

where the worth scope of C_t is (0,1).The result ht−1 and input x_t are gotten as information upsides of the result entryway at time t, and the result ot of the result door is acquired as follows:

$$O_t = \sigma(W_o * [h_{t-1}, x_t] + b_o \quad (4.4)$$

where worth scope O_t of is (0,1), W_o is the heaviness of the result entryway, and $b_o$is the inclination of the result door.

The result worth of LSTM is gotten by computing the result of the result entryway and the condition of the cell, as displayed in the accompanying recipe.

4.5 Prediction:

The testing set is then passed through the trained model to obtain the predicted result. The predicted output is compared with the actual output to evaluate the model's performance.

The first calculation involves the log return of an asset, which measures the percentage change in price between two consecutive days. This is calculated using the formula

log_return = ln(Pt/P(t-1)) * 100

where Pt is the closing price at time t and P(t-1) is the closing price at time t-1.

The rolling mean is the average log return over the last 30 days, calculated by taking the sum of log returns over the last 30 days and dividing by 30. This helps to smooth out any short-term fluctuations in the data and gives a better sense of the overall trend.

The rolling standard deviation is the standard deviation of the log returns over the last 30 days, which provides a measure of the volatility of the asset.

The upper and lower Bollinger Bands are calculated using the rolling mean and standard deviation. The upper band is the rolling mean plus two times the rolling standard deviation, while the lower band is the rolling mean minus two times the rolling standard deviation. These bands provide a range within which the asset's price is expected to fluctuate.

For the next day prediction, the formula is

next_day_prediction = Adj Close * exp(rolling_mean / 100)

where Adj Close is the adjusted closing price of the asset on the last day in the dataset.

This formula predicts the expected price of the asset the following day based on its past performance.

The next week prediction formula is similar, but multiplies the rolling mean by 5 to account for the longer time period:

next_week_prediction = Adj Close * exp(rolling_mean / 100 * 5).

Finally, the next month prediction formula multiplies the rolling mean by 22, which approximates the number of trading days in a month:

next_month_prediction = Adj Close * exp(rolling_mean / 100 * 22).

This provides an estimate of the asset's price one month from the current date based on its past performance.

4.6 Metrics

4.6.1 Evaluation Metrics:

Evaluation metrics used to measure the performance of the model. These metrics help to identify the strengths and weaknesses of the model.The MAE, RMSE, and R2 are utilized as the evaluation criteria of the approaches in order to assess the forecasting effectiveness of CNN-LSTM. The following is the MAE calculation formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - ty_i| \quad (4.5)$$

where y_i is the predictive value and y_i is the true value. The smaller the value of MAE, the better the forecasting. The RMSE calculation formula is as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}}\sum_{i=1}^{n}(\widehat{y_i} - y_i) \quad (4.6)$$

Where $\widehat{y_i}$ is the predictive value and y_i is the true value. The smaller the value of RMSE the better the forecasting. The R^2 calculation formula is as follows:

$$R^2 = \frac{\sum_{i=1}^{n} (y_i - \widehat{y_i})^2)/n}{\sum_{i=1}^{n} (y_i - t\widehat{y_i})^2)/n} \quad (4.7)$$

Where $\widehat{y_i}$ is the predictive value and y_i is the true value, and $\overline{y_i}$ is the average value. The value range of R^2 is (0,1).

The closer the value of MAE and RMSE to 0, the smaller the error between the predicted value and the real value, the higher the forecasting accuracy. The closer R^2 is to 1, the better the fitting degree of the model.

4.6.2 Performance Metrics:

Precision, recall, and accuracy are three common metrics used to evaluate the performance of a model. Here's a brief explanation of each:

Precision: Precision is a measure of the model's ability to correctly identify positive cases. It is calculated as

Precision = True Positives / (True Positives + False Positives)

Recall: Recall is a measure of the model's ability to correctly identify all positive cases. It is calculated as

Recall = True Positives / (True Positives + False Negatives)

Accuracy: Accuracy is a measure of the overall performance of the model. It is calculated as

Accuracy = (True Positives + True Negatives) / (True Positives + False Positives + True Negatives + False Negatives)

All three metrics are important and should be considered when evaluating the performance of a model.

4.7 SUMMARY

Thus, the proposed system is discussed in detail. The next chapter deals with the system implementation.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 OVERVIEW

In this chapter, various algorithms and methods involved in the proposed system implementation are discussed here.

5.2 DATASET DESCRIPTION

The proposed method utilized a dataset from the Kaggle website, which was obtained from Yahoo Finance. The dataset includes information gathered from January 2016 to January 2023, and consists of attributes such as date, open, close, high, low, volumes, and adj Close.

To enhance the accuracy of stock market forecasting, further attributes were added to the dataset. The forecasting procedure was split into two stages: the training and testing phases. Throughout the training phase, several data manipulation methods were implemented, and models were trained on the processed data via various algorithms.

The attributes in the dataset include:

- Date: timestamp of the stock price data
- Open: open price of the day
- Close: final price of the day
- High: highest price of the day
- Low: lowest price of the day
- Volume: number of shares bought by the holders

These are parameters used in this dataset.

5.3 DATASET DETAILS

Table 5.1 Dataset Description

No	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-04	25.652500	26.342501	25.500000	26.337500	24.074739	270597600
1	2016-01-05	26.437500	26.462500	25.602501	25.677500	23.471445	223164000
2	2016-01-06	25.139999	25.592501	24.967501	25.174999	23.012117	273829600
3	2016-01-07	24.670000	25.032499	24.107500	24.112499	22.040897	324377600
4	2016-01-08	24.637501	24.777500	24.190001	24.240000	22.157444	283192000
...	...	...	...	...	...	...	...
1808	2023-03-10	150.210007	150.940002	147.610001	148.500000	148.500000	68524400
1809	2023-03-13	147.809998	153.139999	147.699997	150.470001	150.470001	84457100
1810	2023-03-14	151.279999	153.399994	150.100006	152.589996	152.589996	73695900
1811	2023-03-15	151.190002	153.250000	149.919998	152.990005	152.990005	77167900
1812	2023-03-16	152.160004	156.460007	151.639999	155.850006	155.850006	76161100

5.4 Algorithm:

Step 1. Generating the training dataset by calculating technical indicators

Input-data X – open, high, low, close, volume target Y- log return, direction

Output - Train the dataset

1.N_TIs <- Number of technical indicators

2.For i in range(1, N_TIs+1):

3.# Each parameter of technical indicators

4.TI_i <- technical_indicator_formula(θ_i, X)

5.I <- empty array for stacking N_TIs technical indicators

6.For i in range(1, N_TIs+1):

7.Indicator_i <- normalize(TI_i)

8.I <- stack(Indicator_i)

9.D_Train <- merge(I, Y)

Step 2. Training a CNN-LSTM model and generating the output

Input: train dataset D_Train

Output: CNN-LSTM model with predicted log return and direction

10.I_{List}←list of total technical indicators

11.K←number of technical indicators of each LSTM

12.S←time-window size

13.E←number of total epochs

14. μ←mini batch size #smaller sample of the entire dataset used for training a machine learning model

15.L, $\beta_{1,}$ $\beta_{2,}$ ←Learning rate, Momentum decay rate and adaptive term decay rate

16.**for** i=1,..,N_{CNN} **do**

17. W_i,B_i ←initialized weight and bias

18. V_i <-Sample(I_{List},K)

19. D_i← Sample(D_{Train},S)

20. **for** e=1,..,E **do**

21. r_e, d_e ← CNN_i(V_i,D_i, μ)

22. $Loss_e^r$ ←MSE(r,r_e)

23. $Loss_e^d$ ←crossentropy(d,d_e)

24. W_i,B_i←Update(optimizer(W_i,B_i ,$Loss_e^r$, $Loss_e^d$,L, $\beta_{1,}$ $\beta_{2,}$))

25. **end for**

26. CNN_i

27.**end for**

28.CNN-LSTM←LSTM(CNN1, CNN2,.., CNN_{NLSTM})

The process of generating a training dataset by calculating technical indicators and training a CNN-LSTM model for predicting log return and direction. First, the number of technical indicators is determined, and for each indicator, its respective formula is applied to the input data X. The resulting technical indicators are then normalized and stacked in an array (I) along with the target output Y (log return and direction). This process generates the training dataset. Next, a CNN-LSTM model is trained on the dataset. The model consists of CNN layers followed by an LSTM layer that takes as input a list of K technical indicators sampled from I_List and a time-window size of S. The model is trained for a specified number of epochs (E) with a mini-batch size of μ, learning rate L, momentum decay rate β_(1,), and adaptive term decay rate β_(2,).The weights and biases of the CNN layers are initialized and updated with an optimizer based on the mean squared error (MSE) loss between the predicted and actual log returns (r_e) and directions (d_e) for each epoch (e).Finally, the trained CNN layers are combined with the LSTM layer to form the CNN-LSTM model.

5.5 SUMMARY

This chapter clearly explains the implementation methodology of this system. The next chapter deals with the results and discussions of this project.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 OVERVIEW

This chapter depicts the results of various intermediate steps of the proposed system: This system is tested with stock data from 2016 January to March 1 2023.

6.2 RESULTS

Table 6.1 Result of CNN-LSTM

Layer	**Type**	**Output Shape**	**Param**
Time_distributed	Time_distributed	(None, 1, 98, 64)	256
Time_distributed_1	Time_distributed	(None, 1, 49, 64)	0
Time_distributed_2	Time_distributed	(None, 1, 47, 128)	24704
Time_distributed_3	Time_distributed	(None, 1, 23, 128)	0
Time_distributed_4	Time_distributed	(None, 1, 21, 64)	24640
Time_distributed_5	Time_distributed	(None, 1, 10, 64)	0
Time_distributed_6	Time_distributed	(None, 1, 640)	0
Bi_directional	Bi_directional	(None, 1, 200)	592800
Dropout	Dropout	(None, 1, 200)	0
Bi_directional_1	Bi_directional_1	(None, 200)	240800
Droupout_1	Droupout_1	(None, 200)	0
Dense	dense	(None, 1)	201

6.2.1 METRICS VISUALIZATION

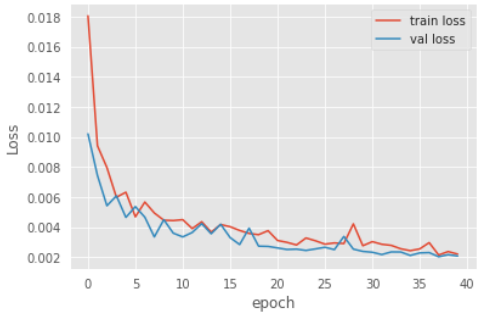

Figure 6.1 MSE loss for model

Figure 6.1 shows the overall loss (i.e. MSE,MAPE,RMSE) for both the training set and validation set over multiple epochs.

6.2.2 VISUALIZATION

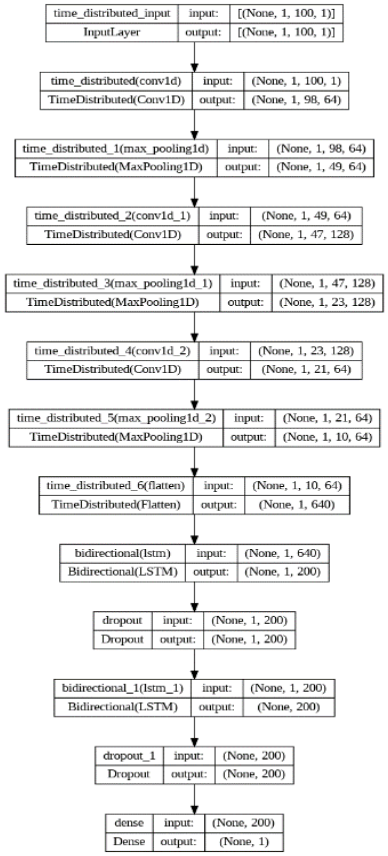

Figure 6.2 Structure of CNN-LSTM Model

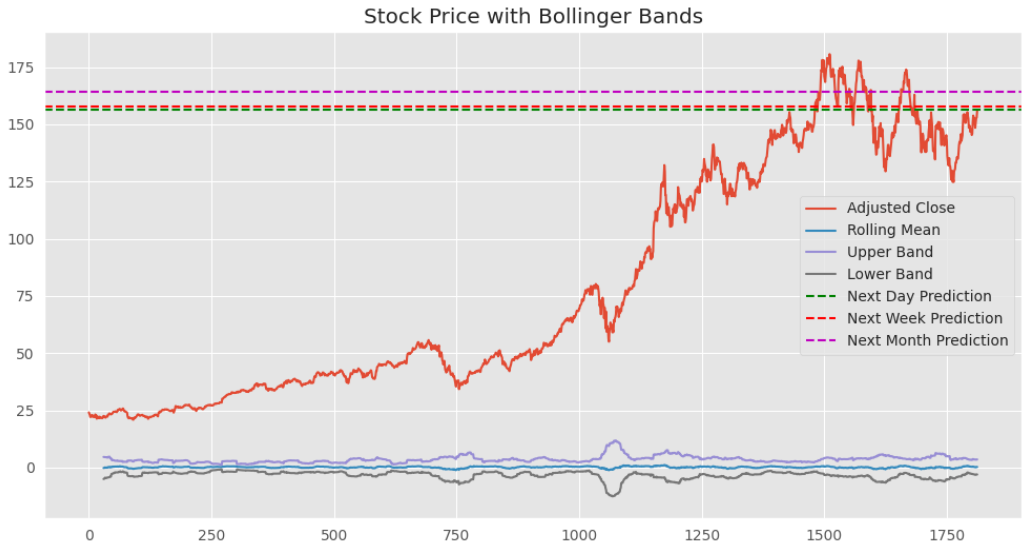

Figure 6.3 Predicted output of CNN-LSTM

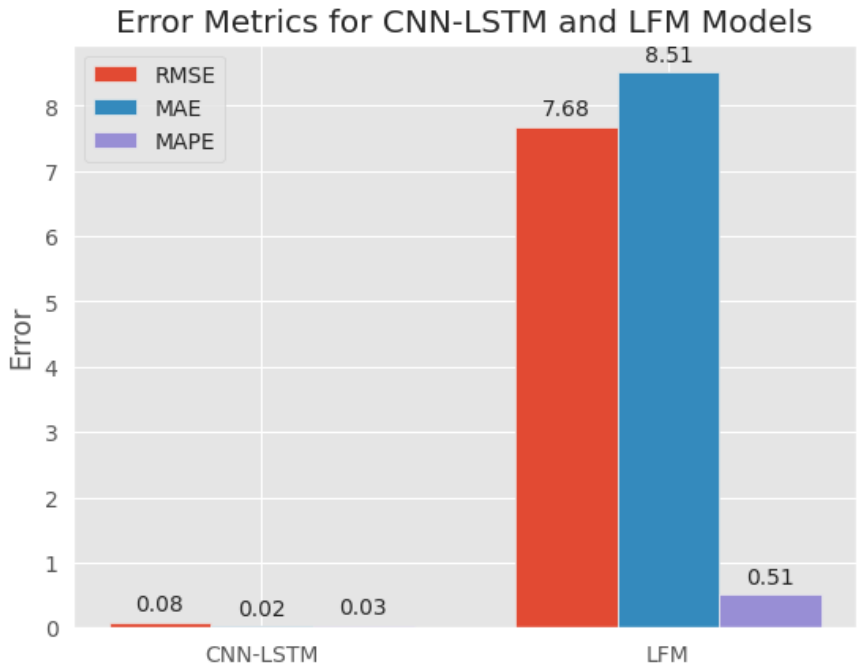

Figure 6.4 Error metrics Comparison

Figure 6.4 provides the comparison of the performance of two methods, CNN-LSTM and LFM, in terms of three error metrics: MAE, MAPE, and RMSE. The first row of the table displays the error metric values for the CNN-LSTM method, while the second

row shows the error metric values for the LFM method. Figure 6 show the graphical format of error metrics.

Based on the Table 6.2, it is evident that the CNN-LSTM method outperforms LFM in all three-error metrics. The MAE value for CNN-LSTM is remarkably low (0.0221), whereas LFM's MAE value is substantially high (8.51). Similarly, CNN-LSTM's MAPE value is significantly low (0.034), whereas LFM's MAPE value is relatively high (0.51). Lastly, CNN-LSTM's RMSE value is notably low (0.0811), while LFM's RMSE value is remarkably high (17.68).

In summary, the Table 6.2 clearly demonstrates that CNN-LSTM outperforms LFM in all three-error metrics, as evidenced by the significantly lower error values for CNN-LSTM and substantially higher error values for LFM.

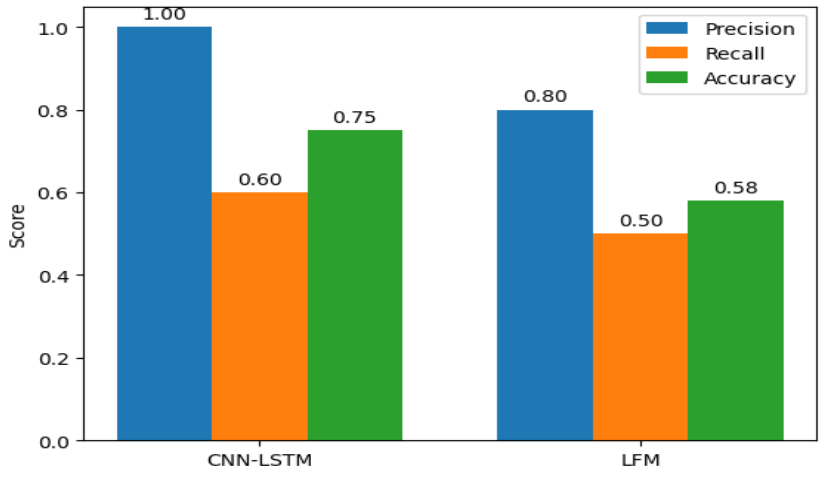

Figure-6.5 Performance metrics comparison for CNN-LSTM and LFM models

Figure 6.5 show the graphical format of performance metrics. compares the precision, recall, and accuracy scores of two models: CNN-LSTM and LFM. The precision, recall, and accuracy scores are displayed on the y-axis, and the model names are displayed on the x-axis. The bar chart shows that the CNN-LSTM model has a higher precision score (1.00) compared to LFM (0.80). However, LFM has a higher recall score (0.50) compared to CNN-LSTM (0.60). Both models have similar accuracy scores, with CNN-LSTM having a score of 0.75 and LFM having a score of 0.58.

Formula for the performance metrics is listed below,

- Precision = True Positives / (True Positives + False Positives)
- Recall = True Positives / (True Positives + False Negatives)
- Accuracy = (True Positives + True Negatives) / (True Positives + False Positives + True Negatives + False Negatives)

Overall, the chart provides a clear comparison of the performance of two models based on three important metrics, allowing for easy comparison and evaluation.

Table 6.2 Performance metrics of the CNN-LSTM and LFM

Model	RMSE	MAE	MAPE
CNN-LSTM	0.0811	0.0221	0.034
LFM	17.68	8.51	0.51

6.3 SUMMARY

This chapter clearly explains the results obtained for the stock price obtained by CNN-LSTM and LFM. The next chapter deals with the conclusion and future works.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The proposed CNN-LSTM model designed to predict closing stock prices for the upcoming day, week, and month, considering the time sequence characteristics of stock data. Inputs, including opening and closing prices, highest and lowest prices, volume, daily return, and moving average, are considered to extract features from the data via the CNN layer, and learn extracted feature data and forecast closing price via the LSTM layer. Results from the dataset show the CNN-LSTM model has highest accuracy of 0.78 and outperforms the LFM model, with the lowest MAE and RMSE and R2 close to 1. Though the model is not perfect, since it only considers stock data and does not integrate factors such as news and national policy.

7.2 FUTURE WORK

Future research will explore sentiment analysis of stock-related news and national policies to improve stock forecasting accuracy, ultimately providing beneficial guidance to investors.

7.3 SOCIAL IMPACT

Social impact can influence stock market forecasting both directly and indirectly. The following are a few ways that social impact may influence stock market forecasting:

• Shifts in public opinion: If there is a change in the public's attitude towards a certain industry, it could result in more money being invested in such companies, which would raise stock prices.

• Regulatory changes: Stock values for some companies may fall if the government enacts new rules that make it more difficult for them to operate or impose higher costs.

• News and media coverage: These factors can significantly affect stock values.

7.4 ECONOMIC ASPECTS

Investment decisions:

- Accurate stock market predictions can guide investors in making investment decisions, resulting in higher returns and increased economic growth.

Risk management:

- Investors and companies rely on stock market predictions to assess and manage risk.

International trade:

- Stock markets are often connected to international trade, and predictions of market trends can impact global economic conditions.

APPENDIX I

WORKING ENVIRONMENT

HARDWARE SPECIFICATION

- System : Intel(R) Core(TM) i5-11th GEN CPU @ 2.50GHz, 2933 MHz, 4 Core(s), 8 Logical Processor(s)
- Mouse : Microsoft Synaptics Touchpad V7.5 on PS/2 Port
- Keyboard : Microsoft Standard PS/2 keyboard
- RAM : 8 GB

SOFTWARE SPECIFICATION

- Operating System : Windows 10
- Tool : Jupyter Notebook, Colab
- Language used : Python 3.7

APPENDIX II

CODING

CNN-LSTM

```
data = pd.read_csv("sp500_data.csv")

bool_series = pd.notnull(data["Adj Close"])

data[bool_series]

data.reset_index(drop=True, inplace=True)

data.fillna(data.mean(), inplace=True)

data.head()

data.shape

data.size

data.describe(include='all').T

data.dtypes

data.nunique()

ma_day = [10,50,100]


for ma in ma_day:
    column_name = "MA for %s days" %(str(ma))
    data[column_name]=pd.DataFrame.rolling(data['Close'],ma).mean()


data['Daily Return'] = data['Close'].pct_change()

# plot the daily return percentage
```

```
data['Daily Return'].plot(figsize=(12,5),legend=True,linestyle=':',marker='o')

plt.show()

sns.displot(data['Daily Return'].dropna(),bins=100,color='green')

plt.show()

date=pd.DataFrame(data['Date'])

closing_df1 = pd.DataFrame(data['Close'])

close1  = closing_df1.rename(columns={"Close": "data_close"})

close2=pd.concat([date,close1],axis=1)

close2.head()

data.reset_index(drop=True, inplace=True)

data.fillna(data.mean(), inplace=True)

data.head()

data.nunique()

data.sort_index(axis=1,ascending=True)

cols_plot = ['Open', 'High', 'Low','Close','Volume','MA for 10 days','MA for 50 days','MA for 100 days','Daily Return']

axes = data[cols_plot].plot(marker='.', alpha=0.7, linestyle='None', figsize=(11, 9), subplots=True)

for ax in axes:

    ax.set_ylabel('Daily trade')

plt.plot(data['Close'], label="Close price")

plt.xlabel("Timestamp")

plt.ylabel("Closing price")

df = data
```

```
print(df)

data.isnull().sum()

cols_plot = ['Open', 'High', 'Low','Close']

axes = data[cols_plot].plot(marker='.', alpha=0.5, linestyle='None', figsize=(11, 9), subplots=True)

for ax in axes:

    ax.set_ylabel('Daily trade')

from sklearn.model_selection import train_test_split


X = []

Y = []

window_size=100

for i in range(1 , len(df) - window_size -1 , 1):

    first = df.iloc[i,2]

    temp = []

    temp2 = []

    for j in range(window_size):

        temp.append((df.iloc[i + j, 2] - first) / first)

    temp2.append((df.iloc[i + window_size, 2] - first) / first)

    X.append(np.array(temp).reshape(100, 1))

    Y.append(np.array(temp2).reshape(1, 1))

# For creating model and training

import tensorflow as tf
```

```
from tensorflow.keras.layers import Conv1D, LSTM, Dense, Dropout, Bidirectional, TimeDistributed

from tensorflow.keras.layers import MaxPooling1D, Flatten

from tensorflow.keras.regularizers import L1, L2

from tensorflow.keras.metrics import Accuracy

from tensorflow.keras.metrics import RootMeanSquaredError


model = tf.keras.Sequential()


# Creating the Neural Network model here...

# CNN layers

model.add(TimeDistributed(Conv1D(64, kernel_size=3, activation='relu', input_shape=(None, 100, 1))))

model.add(TimeDistributed(MaxPooling1D(2)))

model.add(TimeDistributed(Conv1D(128, kernel_size=3, activation='relu')))

model.add(TimeDistributed(MaxPooling1D(2)))

model.add(TimeDistributed(Conv1D(64, kernel_size=3, activation='relu')))

model.add(TimeDistributed(MaxPooling1D(2)))

model.add(TimeDistributed(Flatten()))

# model.add(Dense(5, kernel_regularizer=L2(0.01)))

# LSTM layers

model.add(Bidirectional(LSTM(100, return_sequences=True)))

model.add(Dropout(0.5))

model.add(Bidirectional(LSTM(100, return_sequences=False)))
```

```
model.add(Dropout(0.5))

#Final layers

model.add(Dense(1, activation='linear'))

model.compile(optimizer='adam', loss='mse', metrics=['mse', 'mae'])


history = model.fit(train_X, train_Y, validation_data=(test_X,test_Y),
epochs=40,batch_size=40, verbose=1, shuffle =True)

from tensorflow.keras.utils import plot_model

plot_model(model, to_file='model.png', show_shapes=True, show_layer_names=True)

plt.plot(history.history['loss'], label='train loss')

plt.plot(history.history['val_loss'], label='val loss')

plt.xlabel("epoch")

plt.ylabel("Loss")

plt.legend()

plt.plot(history.history['mse'], label='train mse')

plt.plot(history.history['val_mse'], label='val mse')

plt.xlabel("epoch")

plt.ylabel("Loss")

plt.legend()

plt.plot(history.history['mae'], label='train mae')

plt.plot(history.history['val_mae'], label='val mae')

plt.xlabel("epoch")

plt.ylabel("Loss")

plt.legend()
```

```
from sklearn.metrics import explained_variance_score, mean_poisson_deviance,
mean_gamma_deviance

from sklearn.metrics import r2_score

from sklearn.metrics import max_error


# predict probabilities for test set

yhat_probs = model.predict(test_X, verbose=0)

# reduce to 1d array

yhat_probs = yhat_probs[:, 0]

var = explained_variance_score(test_Y.reshape(-1,1), yhat_probs)

print('Variance: %f' % var)

r2 = r2_score(test_Y.reshape(-1,1), yhat_probs)

print('R2 Score: %f' % var)

var2 = max_error(test_Y.reshape(-1,1), yhat_probs)

print('Max Error: %f' % var2)

predicted  = model.predict(test_X)

test_label = test_Y.reshape(-1,1)

predicted = np.array(predicted[:,0]).reshape(-1,1)

len_t = len(train_X)

for j in range(len_t , len_t + len(test_X)):

    temp = data.iloc[j,3]

    test_label[j - len_t] = test_label[j - len_t] * temp + temp

    predicted[j - len_t] = predicted[j - len_t] * temp + temp

plt.plot(predicted, color = 'green', label = 'Predicted  Stock Price')
```

```
plt.plot(test_label, color = 'red', label = 'Real Stock Price')
plt.title(' Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel(' Stock Price')
plt.legend()
plt.show()
import numpy as np
import datetime

def predict_next_period(data, model, period='day', num_periods=1):
    # Determine the window size based on the period being predicted
    if period == 'day':
        window_size = 100
    elif period == 'week':
        window_size = 7 * 100
    elif period == 'month':
        window_size = 30 * 100

    # Pad the data array with zeros if its length is less than window_size
    if len(data) < window_size:
        data = np.concatenate([np.zeros(window_size - len(data)), data])
    # Use the last window_size values to predict the next value(s)
    last_window = data[-window_size:]
```

```
    # Reshape the data as required by the model
    if period == 'day':
        last_window = np.array(last_window).reshape(1, 1, window_size, 1)
    elif period == 'week':
        last_window = np.array(last_window).reshape(1, 1, window_size, 1)
        last_window = np.split(last_window, 7, axis=2)
        last_window = np.stack(last_window, axis=0)
    elif period == 'month':
        last_window = np.array(last_window).reshape(1, 1, window_size, 1)
        last_window = np.split(last_window, 30, axis=2)
        last_window = np.stack(last_window, axis=0)
     # Make the prediction using the trained model
    prediction = model.predict(last_window)
    # Return the predicted value(s)
    if period == 'day':
        return prediction[0][0]
    elif period == 'week':
        return prediction[0][0] * num_periods
    elif period == 'month':
        return prediction[0][0] * num_periods
test_Y_pred = test_Y_pred.reshape(test_Y_pred.shape[0], 1)
test_Y = test_Y.reshape(test_Y.shape[0], 1)
test_results['Next Day'] = test_results['Actual'].shift(-1)
```

```
test_results['Next Week'] = test_results['Actual'].shift(-7)

test_results['Next Month'] = test_results['Actual'].shift(-30)

 plt.plot(test_results['Actual'], label='Actual')

plt.plot(test_results['Predicted'], label='Predicted')

plt.plot(test_results['Next Day'], label='Next Day')

plt.plot(test_results['Next Week'], label='Next Week')

plt.plot(test_results['Next Month'], label='Next Month')

plt.title('Actual vs Predicted vs Next Day/Week/Month Prices')

plt.xlabel('Time (Days)')

plt.ylabel('Price (USD)')

plt.legend()

plt.show()
```

REFERENCES

[1] China’s commercial bank stock price prediction using a novel k-means-lstm hybrid approach Yufeng Chen,Jinwang Wu,Zhongrui Wu September 2022Expert Systems with Applications 202(4):117370DOI:10.1016/j.eswa.2022.117370

[2] Ensemble deep learning framework for stock market data prediction (EDLF-DP) Parshv Chaajer,Manan Shah,Ameya Kshirsagar November 2021DOI:10.1016/j.dajour.2021.100015

[3] Enhancing Profit by Predicting Stock Prices using Deep Neural Network Soheila Abrishami, M. Turek, +1 author Piyush Kumar Published 1 November 2019 Computer Science, Business 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)

[4] Forecasting Fluctuations in the Financial Index Using a Recurrent Neural Network Based on Price Features Yu-Fei Lin, Tzu-Ming Huang, Wei-Ho Chung , and Yeong-Luh Ueng Date of Publication: 19 February 2020 Electronic ISSN: 2471-285X INSPEC Accession Number: 21099468

[5] Forecasting Nike’s Sales using Facebook Data Boldt, Linda Camilla; Vinayagamoorthy, Vinothan; Winder, Florian; Melanie, Schnittger; Ekram, Mats; Mukkamala, Raghava Rao; Buus Lassen, Niels; Flesch, Benjamin; Hussain, Abid; Vatrapu, Ravi Document Version Final published version Published in: Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016

[6] Hybrid ARIMA-BPNN model for time series prediction of the Chinese stock market Li Xiong, Yuelin Lu Published 21 April 2017 Computer Science 2017 3rd International Conference on Information Management (ICIM)

[7] LSTM-based Deep Learning Model for Stock Prediction and Predictive Optimization Model Akhter Mohiuddin RatherReceived in revised form 2 September 2021; Accepted 28 October 20212193-9438/© 2021 The Author(s). Published by Elsevier Ltd on behalf of Association of European Operational Research Societies(EURO).

[8] Machine learning in the Chinese stock market Markus Leippold a,d, Qian Wang a, Wenyu Zhou b,c,∗ a Department of Banking and Finance, University of Zurich, Platte Strasse 14, Zurich 8032, Switzerland b International Business School, Zhejiang University, Haining, Zhejiang 314400, China c Academy of Financial Research, Zhejiang University Hangzhou, Zhejiang 310058, China d Swiss Finance Institute (SFI), Zürich, Switzerland

[9] Multi objective Automated Type-2 Parsimonious Learning Machine to Forecast Time-varying Stock Indices Online Md Meftahul Ferdaus , Ripon K. Chakraborty , Member, IEEE, and Michael J. Ryan , Senior Member, IEEE Mar.2021

[10] Predicting next day direction of stock price movement using machine learning methods with persistent homology: Evidence from Kuala Lumpur

[11] J.-Y. Liu, K.-C. Hsu and C.-H. Wang, "Stock Price Prediction using Sentiment Analysis of News Articles," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021,pp. 1-6, doi:10.1109/IJCNN52387.2021.9533493.

[12] Stock market prediction based on statistical data using machine learning algorithms

March 2022Journal of King Saud University - Science 34(1):101940
DOI:10.1016/j.jksus.2022.101940

[13] Stock market prediction with deep learning: The case of China Qingfu Liu a, Zhenyi Tao b, Yiuman Tse c,*, Chuanjie Wang Finance Research Letters, Elsevier, vol. 46(PA) 2022

[14] The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction ParshvChhajer[a] MananShah Volume 2, March 2022,

[15] M. A. Arshad and M. H. Shaikh, "Stock Price Prediction Using a Hybrid Deep Learning Model," 2020 IEEE 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2020, pp. 1-6, doi: 10.1109/iCoMET49456.2020.9286231

[16] P. Jin, J. Wang and Z. Zhao, "An LSTM-Based Model for Stock Price Prediction Using Financial News," 2021 IEEE 3rd Conference on Communications, Network and Satellite (ComNeSat), Chengdu, China, 2021, pp. 468-472, doi: 10.1109/ComNeSat52763.2021.9488106.

[17] A. Gupta and S. Kumar, "Stock Price Prediction Using Machine Learning Techniques: A Survey," 2021 12th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 2021, pp. 1-6, doi: 10.1109/ICCCNT53297.2021.9490449.

[18] M. S. Farooq, G. Muhammad and M. S. Khan, "An Ensemble of Machine Learning Models for Stock Price Prediction," 2021 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2021, pp. 382-387, doi: 10.1109/FIT50971.2021.9490112.

[19] "Stock Market Forecasting Using Machine learning Techniques: A Comprehensive Review" by Ehsan Mirzaei, Saeedeh Pourahmadi, and Seyed Hossein Siadat. (2020)

[20] "A Comparative Analysis of Machine Learning Algorithms for Stock Price Prediction" by Muhammad Arslan Arshad, Murtaza Hussain Shaikh, and Muhammad Talha Zahid. (2021)

[21] J. Zhou, Y. Zhang and W. Wang, "Forecasting Stock Prices Using Hybrid Models Based on ARIMA and Machine Learning Algorithms," 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2020, pp. 51-56, doi: 10.1109/ITOEC50180.2020.9185271.

[22] M. I. Khan and G. Muhammad, "Predicting Stock Prices Using Machine Learning Techniques: A Comparative Study," 2021 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2021, pp. 388-393, doi: 10.1109/FIT50971.2021.9490174.

[23] M. Ali, M. A. Arshad and M. H. Shaikh, "Stock Price Prediction Using Machine Learning and Deep Learning Techniques: A Review," 2020 IEEE 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2020, pp. 1-6, doi: 10.1109/iCoMET49456.2020.9286230

[24] S. Ahmed and A. W. Muzaffar, "Predicting Stock Prices with Machine Learning Algorithms: A Survey," 2020 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2020, pp. 1-6, doi: 10.1109/iCoMET49456.2020.9286235.B. M. Patel and R. Patel, "Stock Price Prediction Using LSTM and ARIMA Models," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur,

India, 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225275.
[26] https://finance.yahoo.com/quote/%5EGSPC/