|  |  |  |
| --- | --- | --- |
|  | **MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE**  **Kodambakkam, Chennai-600024** |  |

**SB3001 - PROJECT-BASED EXPERIENTIAL LEARNING**

**PROGRAM**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**TOPIC:**

**STOCK PRICE PREDICTION USING LSTM**

|  |  |
| --- | --- |
| **FACULTY MENTOR:** | **DR S.AARTHI** |
| **INDUSTRY MENTOR:** |  |
|  |  |

**Project Submitted by,**

**ISAAC SAMUEL.J (3115211040018)**

***PROJECT REPORT FORMAT***

* **ABSTRACT**
* **INTRODUCTION**
* ***Project Overview***
* ***Purpose***
* **IDEATION AND PROPOSED SOLUTION**
* ***Problem statement definition***
* ***Ideation and Brainstorming***
* ***Proposed Solution***
* **REQUIREMENTS ANALYSIS**
* ***Functional Requirements***
* ***Non-Functional Requirements***
* **PROJECT DESIGN**
* ***Briefing***
* ***Solution***
* **SOLUTIONS**
* ***Development Part I***
* ***Development Part II***
* **RESULTS**

7.1 ***Performance Metrics***

* **ADVANTAGES AND DISADVANTAGES**
* **CONCLUSION**
* **FUTURE SCOPE**

**SOURCE CODE**

**APPENDIX**

**ABSTRACT;**

Stock price prediction has been a longstanding challenge in financial markets due to its complex and dynamic nature. With the emergence of deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, there has been growing interest in leveraging these models for accurate forecasting. This project presents a detailed analysis of employing LSTM networks for stock price prediction, aiming to provide insights into their effectiveness and limitations.

The study begins by outlining the theoretical foundations of LSTM networks, highlighting their ability to capture long-term dependencies and handle sequential data, which makes them suitable for time-series forecasting tasks such as stock price prediction. Furthermore, the project explores various architectural configurations of LSTM networks, including the number of layers, hidden units, and input features, along with techniques for mitigating over fitting and improving generalization performance.

Next, the project discusses the pre-processing steps involved in preparing the input data for LSTM models, including feature scaling, sequence generation, and data partitioning into training, validation, and testing sets. It also examines different feature engineering techniques and their impact on model performance.

The empirical evaluation involves conducting experiments on real-world stock market datasets, encompassing various stocks and market indices across different time periods. The performance of LSTM models is assessed using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), comparing them with traditional statistical methods and other machine learning approaches.

**INTRODUCTION**

In today's dynamic financial landscape, predicting stock prices accurately is paramount for investors, traders, and financial analysts. Traditional methods often fall short in capturing the complex patterns inherent in stock market data. However, the emergence of deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, offers promising avenues for improved forecasting accuracy.

This project delves into the realm of stock price prediction using LSTM networks, leveraging historical data and advanced neural network architectures to anticipate future price movements. By harnessing the power of deep learning, this endeavour seeks to provide stakeholders with valuable insights to navigate the intricacies of the stock market effectively.

***PROJECT OVER VIEW:***

This project focuses on employing Long Short-Term Memory (LSTM) networks for stock price prediction, addressing the challenges inherent in traditional forecasting methods. Through the acquisition and pre-processing of historical stock price data, an LSTM neural network model will be developed using frameworks like Tensor Flow or PyTorch. This model will undergo rigorous training and evaluation, utilizing metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) to assess its performance.

The project aims to deliver a robust LSTM model capable of accurately forecasting future stock prices, providing valuable insights for investors, traders, and financial analysts. Potential future enhancements include exploring alternative architectures and deploying the model in real-time trading systems.

***PURPOSE OF THE PROJECT:***

The purpose of this project is to develop a Long Short-Term Memory (LSTM) neural network model for stock price prediction. By leveraging historical stock price data and advanced deep learning techniques, the project aims to accurately forecast future price movements.

The primary objective is to provide investors, traders, and financial analysts with a reliable tool for making informed decisions in the stock market. Through rigorous training, evaluation, and potential future enhancements, the project seeks to contribute to the advancement of predictive analytics in financial forecasting.

**IDEATION AND PROPOSED SOLUTION**

***Problem Statement, Definition:***

Despite the notable advancements in deep learning techniques, accurately simulating and forecasting stock price trajectories remains a complex task due to the inherent volatility and non-linear dynamics of financial markets. While Long Short-Term Memory (LSTM) networks have shown promise in sequence prediction tasks, their application as generative models for generating realistic and diverse stock price trajectories presents significant challenges.

The task at hand involves training LSTM-based generative AI models on historical stock price data to generate new price sequences that closely resemble actual market behaviour. However, the intricate temporal dependencies, volatility patterns, and high-dimensional nature of stock price data pose challenges for LSTM networks in capturing and reproducing the underlying dynamics accurately.

**IDEATION AND BRAINSTORMING:**

***Data Acquisition and Pre-processing:***

Identify and collect historical stock price data from reliable sources such as financial databases or APIs. Pre-process the data to handle missing values, normalize the features, and create sequential input sequences suitable for LSTM training. Explore techniques for incorporating additional data sources such as market sentiment, economic indicators, and news sentiment analysis to enrich the dataset.

***Model Architecture Design:***

Design the architecture of the LSTM-based generative AI model, considering factors such as the number of layers, hidden units, activation functions, and input features. Investigate techniques for enhancing model capacity and capturing complex temporal dependencies in stock price data, such as attention mechanisms, residual connections, and gated recurrent units (GRUs).

Experiment with variations of LSTM architectures, including stacked LSTMs, bidirectional LSTMs, and attention-based LSTMs, to determine the most effective configuration for generating high-quality price trajectories.

***Training and Optimization:***

Develop a training pipeline to optimize the LSTM-based generative model using historical stock price data. Implement advanced optimization techniques such as learning rate scheduling, gradient clipping, and batch normalization to improve training stability and convergence.

Explore regularization techniques such as dropout and weight decay to prevent over fitting and enhance the generalization ability of the model.

***Evaluation and Validation:***

Define evaluation metrics to assess the fidelity and quality of generated stock price trajectories, including statistical measures such as MAE, MSE, RMSE, and qualitative assessments based on visual inspection.

Conduct rigorous validation experiments using held-out datasets or cross-validation to evaluate the model's performance across different market conditions and time periods.

Compare the generated price trajectories against real market data to validate the model's ability to capture key features and dynamics of stock price movements accurately.

***Integration and Deployment:***

Integrate the trained LSTM-based generative AI model into a user-friendly application or platform that allows users to generate synthetic stock price trajectories on-demand. Explore deployment options such as cloud-based services or APIs to make the model accessible to a wider audience of researchers, investors, and financial analysts.

Provide documentation and tutorials to guide users in effectively utilizing the generated stock price for various applications, including back testing, scenario analysis, and risk assessment.

***Iterative Improvement and Feedback Loop.***

Establish a feedback loop to gather user feedback and iteratively improve the LSTM-based generative AI model based on real-world usage scenarios. Continuously monitor and evaluate the model's performance over time, incorporating new data and insights to enhance its accuracy, robustness, and usability.

Collaborate with domain experts and stakeholders to identify emerging trends and challenges in stock market prediction and adapt the model accordingly to meet evolving needs and requirements.

**PROPOSED SOLUTION:**

The project will involve the development of an LSTM-based neural network model for stock price prediction. Initially, historical stock price data will be collected and pre-processed to handle missing values and normalize features. The LSTM model architecture will be designed and implemented using Tensor Flow or PyTorch, with configurable parameters such as the number of layers and units.

Training will involve optimizing the model parameters using techniques like stochastic gradient descent and learning rate scheduling. The trained model will then be evaluated using metrics such as Mean Absolute Error and Mean Squared Error to assess its predictive performance.

Ultimately, the goal is to provide accurate forecasts of future stock prices to aid investors, traders, and financial analysts in their decision-making processes.

**REQUIREMENT ANALYSIS**

***Functional Requirements:***

|  |  |  |
| --- | --- | --- |
| **S.No** | **Metrics** | **Description** |
| FR1 | Data Acquisition | The system shall allow users to input historical stock price data |
| FR2 | Data Pre-processing | The system shall pre-process the input data to handle missing values and normalize the features. |
| FR3 | LSTM Model Architecture | The system shall design LSTM-based predictive model architecture with configurable parameters |
| FR4 | Model Training | The system shall implement a training loop to train the LSTM model on historical stock price data |
| FR5 | Prediction Generation | The system shall generate predictions for future stock prices using the trained LSTM model. |

**Non-Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Requirement** | **Description** |
| NFR1 | Scalability | The system shall be capable of processing large volumes of data efficiently. |
| NFR2 | Security | The system shall ensure the confidentiality and integrity of sensitive data. |
| NFR3 | Reliability | The system shall operate reliably under normal and peak load conditions. |
| NFR4 | Performance | .The system shall be capable of processing large volumes of data efficiently. |
| NFR5 | Usability | The user interface shall be intuitive and user-friendly for both novice and expert users. |
| NFR6 | Robustness | The system shall handle noisy and incomplete input data gracefully. |

**PROJECT DESIGN**

***Briefing:***

The project aims to implement Long Short-Term Memory (LSTM) based Generative AI models to generate high-resolution stock price trajectories. This project design outlines the objectives, methodologies, and key milestones for achieving this goal.

***Solution:***

The solution involves the implementation of LSTM-based Generative AI models to generate high-resolution stock price trajectories using historical market data.

**DEVELOPMENT: PART 1**

In the first phase of development, foundational components of the project will be implemented:

**Data Acquisition and Pre-processing:** Obtain historical stock price data from reliable sources and pre-process it to handle missing values, normalize features, and create sequential input sequences suitable for LSTM training.

**Architecture Design:** Design the architecture of the LSTM-based generative AI model, specifying parameters such as the number of layers, hidden units, and input features.

Model Training: Train the LSTM-based generative model on the pre-processed stock price data, using appropriate optimization algorithms and loss functions.

**DEVELOPMENT: PART 2**

The second phase of development focuses on fine-tuning and optimizing the LSTM-based generative model:

Hyper parameter Tuning: Fine-tune model hyper parameters such as learning rates, batch sizes, and regularization parameters to improve model performance.

**Regularization Techniques**: Implement regularization techniques such as dropout and weight decay to prevent over fitting and enhance model generalization.

**Advanced Training Strategies**: Explore advanced training strategies such as curriculum learning and adversarial training to enhance model robustness and stability.

**RESULTS**

In the results phase, the performance of the LSTM-based generative model for stock price trajectory generation is evaluated and validated. This entails assessing the fidelity and quality of generated stock price trajectories using statistical measures such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and qualitative assessments

Additionally, visualization techniques are employed to depict the generated stock price trajectories and compare them against real market data, ensuring accuracy and realism. Through performance analysis, the effectiveness of LSTM-based generative

AI models for stock price trajectory generation is thoroughly examined, drawing conclusions based on the evaluation metrics and visualization outcomes. This comprehensive evaluation process provides insights into the model's performance and its potential implications for financial forecasting and decision-making.

**PERFORMANCE METRICS:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Metrics** | **Description** |
| PM1 | Mean Absolute Error (MAE) | Measures the average absolute difference between the generated and actual stock prices. |
| PM2 | Mean Squared Error (MSE) | Measures the average squared difference between the generated and actual stock prices. |
| PM3 | Root Mean Squared Error (RMSE) | Measures the square root of the average squared difference between the generated and actual stock prices. |
| PM4 | Mean Absolute Percentage Error (MAPE) | Measures the average percentage difference between the generated and actual stock prices. |
| PM5 | Pearson Correlation Coefficient | Measures the linear correlation between the generated and actual stock price trajectories. |

**ADVANTAGES AND DISADVANTAGES:**

**ADVANTAGES:**

***Accurate Simulation***: LSTM-based generative AI models have the potential to accurately simulate stock price trajectories, capturing complex temporal dependencies and market dynamics.

***Flexibility:*** These models can generate diverse and realistic stock price trajectories, enabling exploration of various market scenarios and potential outcomes.

***Enhanced Decision Support***: High-resolution stock price trajectories can provide valuable insights for investors, traders, and financial analysts, facilitating informed decision-making and risk management.

***Data Augmentation***: Generative AI models can augment limited historical data by generating synthetic stock price trajectories, enabling more robust and effective training of predictive models.

***Scenario Analysis***: Generated stock price trajectories can be used for scenario analysis, stress testing, and sensitivity analysis, helping investors anticipate potential market fluctuations and plan accordingly.

**DISADVANTAGES:**

***Complexity***: Designing and training LSTM-based generative AI models for stock price prediction can be complex and computationally intensive, requiring expertise in deep learning and financial modelling

.

***Model Uncertainty***: Generative AI models may introduce uncertainties and biases in generated stock price trajectories, potentially leading to inaccurate predictions and decision-making.

***Over fitting:*** There is a risk of over fitting when training generative AI models on historical data, resulting in unrealistic and biased generated trajectories that do not generalize well to unseen market conditions.

***Interpretability:*** Generated stock price trajectories may lack interpretability, making it challenging for users to understand the underlying factors driving the predicted outcomes.

***Ethical Considerations***: The use of synthetic data generated by AI models for financial decision-making raises ethical concerns regarding accountability, transparency, and potential market manipulation.

**CONCLUSION**

In conclusion, the project “Stock Price Prediction Using LSTM-Based Generative AI Models" presents a significant advancement in the field of financial forecasting and analysis. Through the implementation of Long Short-Term Memory (LSTM) based generative AI models, the project aimed to simulate and generate high-fidelity stock price trajectories, providing valuable insights for investors, traders, and financial analysts.

Throughout the project lifecycle, key milestones were achieved, including data acquisition and pre-processing, architecture design, model training and optimization, as well as evaluation and validation of the generated stock price trajectories. Performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and correlation coefficients were utilized to assess the accuracy and fidelity of the LSTM-based generative AI models.

The results of the project demonstrate the effectiveness and potential of LSTM-based generative AI models in generating realistic and diverse stock price trajectories. By capturing complex temporal dependencies and market dynamics, these models offer valuable support for decision-making processes in financial markets, including scenario analysis, risk assessment, and portfolio optimization.

**FUTURE SCOPE**

***Enhanced Model Architectures***: Further exploration of advanced LSTM architectures, such as attention-based mechanisms, memory-augmented networks, and transformer models, to improve the generative capabilities and performance of AI models in capturing complex market dynamics.

***Incorporation of External Factors***: Integration of additional external factors, such as macroeconomic indicators, industry news sentiment, and geopolitical events, into LSTM-based generative AI models to enhance the accuracy and relevance of generated stock price trajectories.

***Multi-Modal Data Fusion:*** Investigation of multi-modal data fusion techniques to incorporate diverse data sources, including textual data, image data, and social media signals, into generative AI models for a more comprehensive understanding of market behavior and sentiment.

***Dynamic Model Adaptation***: Development of adaptive LSTM-based generative AI models capable of dynamically adjusting model parameters and architectures in response to changing market conditions, enabling real-time adaptation and improved prediction accuracy.

***Interpretability and Explain ability***: Research into techniques for enhancing the interpretability and explain ability of generated stock price trajectories, enabling users to understand the underlying factors driving the predictions and making more informed decisions.

***Ethical and Regulatory Considerations***: Continued exploration of ethical and regulatory frameworks for the responsible use of generative AI models in financial decision-making, addressing concerns related to transparency, fairness, and accountability.

**SOURCE CODE:**

import pandas as pd

import datetime as dt

from datetime import date

import matplotlib.pyplot as plt

import yfinance as yf

import numpy as np

import tensorflow as tf

START = "2015-01-01"

TODAY = date.today().strftime("%Y-%m-%d")

# Define a function to load the dataset

def load\_data(ticker):

data = yf.download(ticker, START, TODAY)

data.reset\_index(inplace=True)

return data

data = load\_data('AAPL')

df=data

df.head()

plt.title("Close Price Visualization")

plt.plot(df.Close)

df

ma100 = df.Close.rolling(100).mean()

ma100

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

plt.plot(df.Close)

plt.plot(ma100, 'r')

plt.title('Graph Of Moving Averages Of 100 Days')

ma200 = df.Close.rolling(200).mean()

ma200

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

plt.plot(df.Close)

plt.plot(ma100, 'r')

plt.plot(ma200, 'g')

plt.title('Comparision Of 100 Days And 200 Days Moving Averages')

df.shape

train = pd.DataFrame(data[0:int(len(data)\*0.70)])

test = pd.DataFrame(data[int(len(data)\*0.70): int(len(data))])

train.head()

test.head()

from sklearn.preprocessing import MinMaxScaler

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

train\_close = train.iloc[:, 4:5].values

test\_close = test.iloc[:, 4:5].values

data\_training\_array = scaler.fit\_transform(train\_close)

data\_training\_array

x\_train = []

y\_train = []

for i in range(100, data\_training\_array.shape[0]):

x\_train.append(data\_training\_array[i-100: i])

y\_train.append(data\_training\_array[i, 0])

x\_train, y\_train = np.array(x\_train), np.array(y\_train)

x\_train.shape

from tensorflow.keras.layers import Dense, Dropout, LSTM

from tensorflow.keras.models import Sequential

model = Sequential()

model.add(LSTM(units = 50, activation = 'relu', return\_sequences=True

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

model.add(Dropout(0.2))

model.add(LSTM(units = 60, activation = 'relu', return\_sequences=True))

model.add(Dropout(0.3))

model.add(LSTM(units = 80, activation = 'relu', return\_sequences=True))

model.add(Dropout(0.4))

model.add(LSTM(units = 120, activation = 'relu'))

model.add(Dropout(0.5))

model.add(Dense(units = 1))

model.summary()

model.compile(optimizer = 'adam', loss = 'mean\_squared\_error', metrics = ['MAE'])

model.fit(x\_train, y\_train, validation\_data = (x\_test, y\_test) ,epochs = 50)

model.save('keras\_model.h5')

test\_close.shape

test\_close

past\_100\_days = pd.DataFrame(train\_close[-100:])

test\_df = pd.DataFrame(test\_close)

final\_df = past\_100\_days.append(test\_df, ignore\_index = True)

final\_df.head()

input\_data = scaler.fit\_transform(final\_df)

input\_data

input\_data.shape

x\_test = []

y\_test = []

for i in range(100, input\_data.shape[0]):

x\_test.append(input\_data[i-100: i])

y\_test.append(input\_data[i, 0])

x\_test, y\_test = np.array(x\_test), np.array(y\_test)

print(x\_test.shape)

print(y\_test.shape)

y\_pred = model.predict(x\_test)

y\_pred.shape

y\_test

y\_pred

scaler.scale\_

scale\_factor = 1/0.00985902

y\_pred = y\_pred \* scale\_factor

y\_test = y\_test \* scale\_factor

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

plt.plot(y\_test, 'b', label = "Original Price")

plt.plot(y\_pred, 'r', label = "Predicted Price")

plt.xlabel('Time')

plt.ylabel('Price')

plt.legend()

plt.show()

from sklearn.metrics import mean\_absolute\_error

mae = mean\_absolute\_error(y\_test, y\_pred)

print("Mean absolute error on test set: ", mae)

**APPENDIX:**

**SOURCE CODE: https://github.com/Isaachrjrj/IBM\_AI-PROJECT/upload**