Stock Price Prediction using CNN and LSTM with leading indicators

Department Of Mathematics,
Indian Institute Of Technology Madras

Supervisor : Dr. Priyanka Shukla

Team 4

Deepak Singh - MA23M006

Hareesh P Nair - AE20B027

Mousina Barman-MA23M011

Snehal - MA23M020

Sonu Kumari - MA23M021

Abstract

The recent advancements in the field of neural network designs in the era of rapidly changing financial market prediction domain shows a great scope and promises for increase in improving the accuracy in stock price predictions . This project proposed the development of a new neural network based framework Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) i.e. combining CNN and LSTM to improve stock market forecasts. The proposed model works by first representing the input data, then uses convolutional layers to get some continuous features and then finally makes predictions using LSTM units. This approach ensures efficient extraction of both temporal, non-temporal, and deals with the hurdles which arises by non-linearity as well as noise in the data.

Extensive experiments have been carried out which aimed at verifying whether CNN-LSTM perform better than other models like traditional models as SVM and Random Forests. The results obtained shows that the newly proposed approach in the project surpasses them a great deal with a better track record in forward trend prediction of stock values.The combination of CNN-LSTM framework have proven to be a very effective method for making complex patterns adapted in financial data, hence leading to this boost.

The predictive accuracy of stock markets is not only improved but also it battles the natural differences and unpredictability of the stock market through the propsed CNN-LSTM framework.

Contents

1 Introduction **4**

2 Objective **5**

3 Literature Survey **5**

4 Background **6**

4.1 Convolutional Neural Network 6

4.1.1 Convolutional Layer 6

4.1.2 Pooling Layer 7

4.1.3 Fully Connected Layer 8

4.2 Long Short-Term Memory Networks 8

4.2.1 Forgetting Gate 9

4.2.2 Input Gate 9

4.2.3 Output Gate 9

5 Data Preprocessing **10**

5.1 Data Set Description 10

5.1.1 Features Description 10

5.1.2 Features Visualization 11

5.2 Normalization Function 12

6 CNN-LSTM Framework **12**

6.1 Input Data 12

6.2 Continuous Features Extraction 13

6.3 Prediction 13

6.4 Proposed CNN-LSTM Process 13

6.4.1 The Input Data 13

6.4.2 Extraction of Continuous Features 13

6.4.3 Prediction 13

6.5 Example Configuration of CNN-LSTM 15

7 Experimental Results **15**

8 Conclusion **20**

9 Code link **20**

1 Introduction

Modern day financial services comprises of multi-strategic ways to manage and optimize financial well-being of individuals, businesses, institutions etc. Detailed processes which includes research, analysis and optimization is carried out by individuals and organizations on a daily basis with a long-term objective of maximizing their wealth over time. Some of the most common services in the financial service industry are financial advisory, asset management, investment banking, trading, portfolio optimization etc. A lot of these process rely on complex mathematical tools and methods for analysis and prediction of best possible outcome one can achieve from the perspective of achieving long-term financial goals. These services helps one reduce the risk-to-reward ratio and hence provide financial stability.

A popular choice of wealth management strategy that most prefer to choose is investing in stocks. Stocks are securities that represents the ownership of company. When a company decides to go public, it offers a percentage of its ownership to the public for a certain price, a strategy which can benefit both parties. But the price that is set is subjected to changes according various market conditions, which cannot be predetermined as investment decision varies from person to person, resulting in a very dynamic price setting. Investing in stocks is one of the most favourable forms of wealth management as it has the potential for high return of investment but also the same potential for losses as well. Hence, identifying the optimal approach to stock investment that balances high returns with proper risk management emerges as an exciting field within the realm of wealth management. Such an approach not only seeks to maximize returns but also prioritizes the preservation and sustainable growth of wealth over time, aligning with the core principles of comprehensive financial management strategies.

Our work aims to develop a mathematical model that can predict the movement of stock prices with high accuracy. There have been numerous attempts in the past few decades to predict the direction of stock prices by various individuals and organizations. These attempts include statistical methods like time series analysis, volatility modeling, etc., and due to the rise in computational capability in the past two decades, various machine learning algorithms such as Random Forest, Support Vector machines, and Deep learning methods have also been used in computational finance domain to predict stock prices. Our method aims to combine two models, namely: Convolutional Neural Network (CNN) and Long-Short-Term-Memory (LSTM) Network, to create a hybrid model that can predict stock price fluctuation with high accuracy.

In our project, we aim to enhance stock market prediction methodologies by integrating advanced mathematical models and machine learning techniques. By combining CNN and LSTM Networks into a hybrid model, we seek to enhance the accuracy and reliability of stock price prediction. The combination of machine learning and stock prediction has a lot of potential for research and application as well which can revolutionize the financial industry.

2 Objective

The objective of this project is to develop a hybrid CNN-LSTM model that can predict the variation of stock prices of Indian Companies, listed on the National Stock Exchanges, up to high accuracy. The objective is achieved through the following sub-tasks:

1. Data Analysis: Stock market historical data is a very complex and large dataset which needs to be processed before it is used to train any machine learning model. Our project focuses of analyzing the data and choose the specific stock market indices that has higher impact on the price of a stock.

2. Hyper-parameter tuning and model training: The analyzed data will be split into training and testing set which is fed into the CNN-LSTM model which can identify the hidden pattern among the training data and develop a relationship for between the input and output. Different sets of hyper-parameters will be used for the model to obtain the best fit one for the data.

3. Prediction of the stock price and benchmarking: The trained model will be used to predict the stock price for different length of days and the prediction accuracy of the CNN-LSTM model will be benchmarked with other machine learning models - Random Forest and Support Vector Machines.

3 Literature Survey

Many research papers explore the use of various machine learning techniques to predict stock prices. An important method is to use random forests to predict market prices. Although some studies focus on specific areas (such as daily load forecasting in the energy industry), others propose new algorithms for weather forecasting. Genetic algorithms can also be used in mining operations. The use of deep learning models such as recurrent neural network and LSTM networks has become popular in stock price prediction. In addition, hybrid models and equally heterogeneous metaheuristics have also contributed to the field. Together, these studies provide insight into the use of machine learning techniques to predict stock prices.

[1] In this paper, stock price predictions are done using Random Forest algorithm. The authors propose a model to predict stock price direction based on historical data. Random Forests are known for their effectiveness in forecasting stock prices and returns.

[2] A study was conducted on forecasting daily stock market returns using dimensionality reduction. They applied artificial neural network (ANN) classifiers to predict

the daily return .Their research contributes to understanding how machine learning techniques can enhance stock market prediction

[3] This paper introduces the Stock Sequence Array Convolutional Neural Network (SSACNN) for feature extraction and stock price movement prediction.SSACNN constructs an input image from historical data and extracts feature vectors using CNN layers.

[4] In this paper, LSTM is used to predict the stock prices specifically for Indian share market.

4 Background

In this section, We will consider CNN and LSTM which are the main components of the proposed algorithm:

4.1 Convolutional Neural Network

Through the progression of DNN, convolutional neural network, which is one of the most prominent algorithms today, was proposed. It has been used successfully in many ways such as search and segmentation. CNN provides better interpretation compared to machine learning algorithms. It consists only of convolution layer, pooling layer and fully connected layer. Detailed information about each episode on CNN is as follows:

4.1.1 Convolutional Layer

The function of the convolution layer is to collect the features of the input data and consists of several convolution kernels. Each unit of the convolution kernel is associated with a bias vector and a weight coefficient, similar to the neurons of a feedforward neural network. Each neuron in the convolutional layer is associated with a single neuron in the region of the previous layer. The size of this field depends on the size of the convolution kernel, called the "receptive field". Its importance is comparable to the receptive field of the cerebral cortex. When the convolution kernel runs, it scans the input features, adds the input features, and multiplies them by matrix elements in the receptive field to superimpose the deviations:

$$Y_{l+1}(c,d) = [Y_l * w_{l+1}](c,d) + b \tag{1}$$

$Y_{l+1}(c,d)$: The value at spatial position (c,d) in the output feature map Y_{l+1}.
Y_l: The input feature map.
w_{l+1}: The convolution kernel.
$[Y_l * w_{l+1}](c,d)$: Convolution operation between Y_l and w_{l+1} at position (c,d).
b: Bias term added to each position in the output feature map.

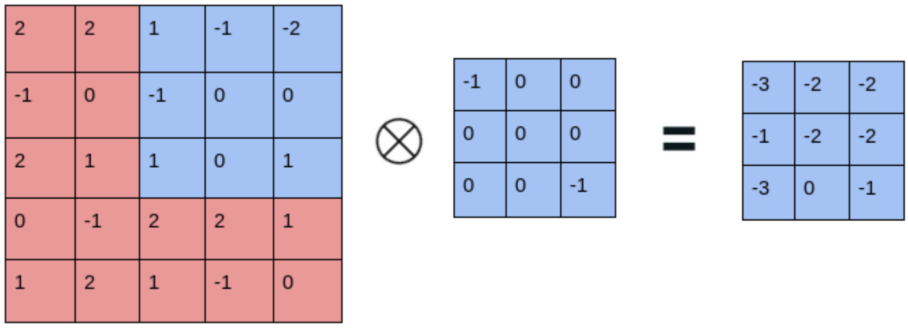

Figure 1: Convolution

4.1.2 Pooling Layer

The output feature maps of the convolutional layer are passed to the pooling layer for data filtering and feature selection. The pooling layer has a preset pooling function; Its task is to replace the result of a point with the statistical data of the feature map of the adjacent area in the feature map. The selection of the pooling area is not separated from the convolution kernel faster feature map in the pooling layer and is controlled by the pooling size, step size and padding. It is generally expressed as:

$$A_l^p(k,c,d) = \left[\sum_{x=1}^{f}\sum_{y=1}^{f} A_l^k(s_0c+x, s_0d+y)^p\right]^{1/p} \tag{2}$$

s_0: Step size or stride of the pooling operation.
A_l: Input feature map to the pooling layer.
$A_l^p(c,d)$: Result of pooling operation at position (c,d) in the output feature map.
p: Pooling size, determining the size of the pooling area.

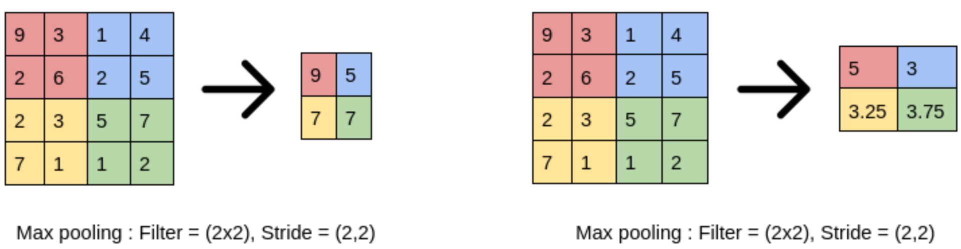

Figure 2: Maximum Pooling

4.1.3 Fully Connected Layer

The fully connected layer is located at the end of the hidden layer and only sends signals to other layers as a whole. The feature map loses the spatial topology in all connected layers, expands to a vector, and switches to the activation function. Convolution and pooling layers in CNN can extract features from input data. The fully connected layer plays the role of classifier in the entire convolutional neural network. Convolutional layers, pooling layers, and activation layers transform the raw data into latent feature space. The function of the entire linking process is to introduce the learning distributed feature representation into the label model. The full integration process does not wait for results to be removed but tries to use existing advanced features to achieve the learning goal.

4.2 Long Short-Term Memory Networks

LSTM is a real-time Recurrent neural network. It is specifically designed to look into the long-term dependency problem of Recurrent neural network.Nowadays it is used in many areas such as machine translation, speech recognition, image annotation, video tagging, and financial time series. Every Recurrent neural network have a repetitive neural network cycle .It usually includes forgetting gate, entry gate and exit gate. A visual representation of this is shown below:

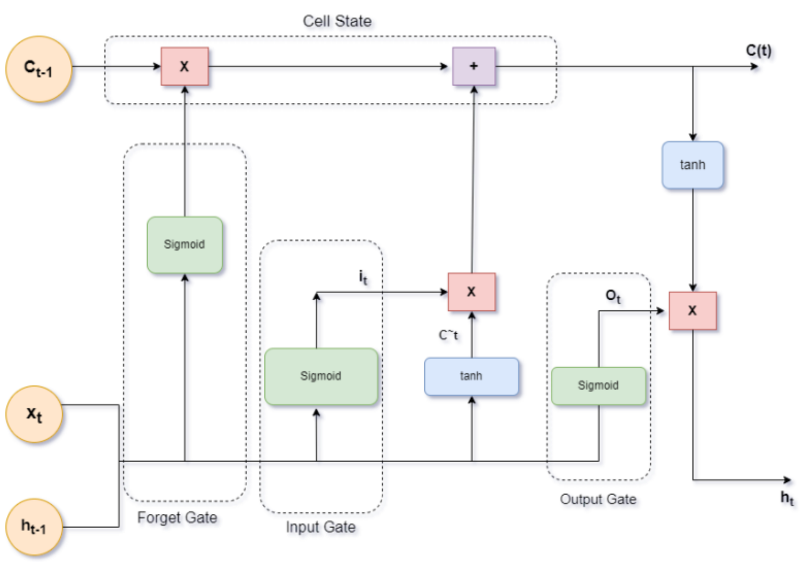

Figure 3: A single LSTM cell in the network

4.2.1 Forgetting Gate

The forget gate in the LSTM network determines which data in the current cell state should be forgotten and which data should be stored and transmitted to the current cell state. It is computed as:

$$z_t = \delta\left(E_f \cdot [h_{t-1}, x_t] + b_f\right)$$

where:

- E_f: Weight matrix associated with the forgetting gate.
- δ: Sigmoid activation function.
- h_{t-1}: Previous hidden State.
- x_t: Current Input.
- b_f: Bias term.

4.2.2 Input Gate

Input section in the LSTM network determine what new data should be added to the current cell state. It has a sigmoid layer and a tanh layer. The output and mean of the sigmoid curve are calculated as follows:

$$j_t = \sigma\left(E_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$B_t = \tanh\left(E_c \cdot [h_{t-1}, x_t]\right) \cdot j_t$$

where:

- E_i and E_c: Weight matrices associated with the input gate.
- σ: Sigmoid activation function.
- b_i: Bias term.

4.2.3 Output Gate

The output gate in the LSTM network controls what part of the current unit state should be output as the output of the current time step. It combines the updated cells with the current hidden state to produce the final output. it is computed as:

$$h_t = \tanh\left(E_o \cdot [h_{t-1}, x_t]\right) \cdot z_t + B_t$$

where:

- E_o: Weight matrix associated with the output gate.
- z_t: Output of the forgetting gate.

- B_t: Intermediate result from the input gate.

5 Data Preprocessing

5.1 Data Set Description

The dataset used in this work includes six different stocks from the leading companies of India markets: Adani port, Tata Motors, SBI, Idea, TCS, Tech-Mahindra. The dataset has features such as Date,High, Open,Previous close, Low, Close, vwap,ltp, 52wH i.e 52 weeks high, 52wL i.e 52 weeks low, volume, value and number of trades.

	Date	series	OPEN	HIGH	LOW	PREV. CLOSE	ltp	close	vwap	52W H	52W L	VOLUME	VALUE	No of trades
0	07-Mar-2024	EQ	1,275.00	1,293.10	1,265.55	1,271.00	1,283.20	1,288.15	1,281.70	1,416.30	981.05	20,73,691	2,65,78,50,269.40	94,123
1	06-Mar-2024	EQ	1,267.50	1,274.35	1,250.10	1,272.50	1,271.00	1,271.00	1,261.48	1,416.30	981.05	20,64,350	2,60,41,34,310.50	94,379
2	05-Mar-2024	EQ	1,274.90	1,279.90	1,259.10	1,280.05	1,270.60	1,272.50	1,271.83	1,416.30	981.05	14,20,379	1,80,64,78,256.30	91,152
3	04-Mar-2024	EQ	1,275.60	1,285.50	1,270.10	1,272.50	1,280.80	1,280.05	1,277.72	1,416.30	981.05	18,43,831	2,35,59,07,739.75	79,537
4	02-Mar-2024	EQ	1,278.00	1,282.00	1,271.00	1,271.80	1,271.90	1,272.50	1,274.88	1,416.30	981.05	2,17,683	27,75,19,800.50	6,538

Figure 4: Dataset

5.1.1 Features Description

Each feature provides crucial insights into different aspects of trading activity, aiding in comprehensive market analysis and decision-making. The features are described as follows:

- **Date**: The specific day when the trades occurred.
- **Series**: A classification system used to categorize different types of securities based on their characteristics and rights. For example, common shares are often categorized as equity (EQ).
- **Open**: It is price at which a particular stock starts trading when the market opens for the day. It represents the first price at which buyers and sellers transact.
- **High**:It is highest price at which a security traded during a specific period, typically within a single trading day. It shows the peak level reached by the security's price during that period.
- **Low**: The lowest price at which a security traded during a specific period, usually within a single trading day. It indicates the lowest level reached by the security's price during that period.
- **Prev.Close**: : It is closing price of a security on the trading day immediately preceding the current one. It serves as a reference point for analyzing price movements and comparing the current day's performance
- **ltp**: The most recent price at which a trade occurred for a particular security. It represents the price at which the most recent buyer and seller agreed to transact.

- **Close**: The final price at which a security trades at the end of a trading session. It is often considered the official closing price for the day and is used to calculate performance metrics and returns.
- **vwap**: A measure of the average price of a security weighted by its trading volume over a specified time period, providing insight into the average transaction price.
- **52 w H**: The highest price at which a security traded over the past 52 weeks or one year. It indicates the peak level reached by the security's price during that time period.
- **52 w L**: The lowest price at which a security traded over the past 52 weeks or one year. It represents the lowest level reached by the security's price during that time period.
- **Volume**:The total number of shares traded during a specified period, representing the level of market activity.
- **Value**: The total monetary value of shares traded during a specified period, calculated by multiplying the volume traded by the price.
- **Number of Trades**:It is total count of trades executed during a specified period, providing insight into trading activity and liquidity levels.

5.1.2 Features Visualization

Figure 5 shows comprehensive visualization depicting the essential data features—open, high, low, close, and 52-week low—across a 250-day timeframe. This graphical representation enables a thorough examination of the trends and fluctuations in the dataset, facilitating informed decision-making and analysis.

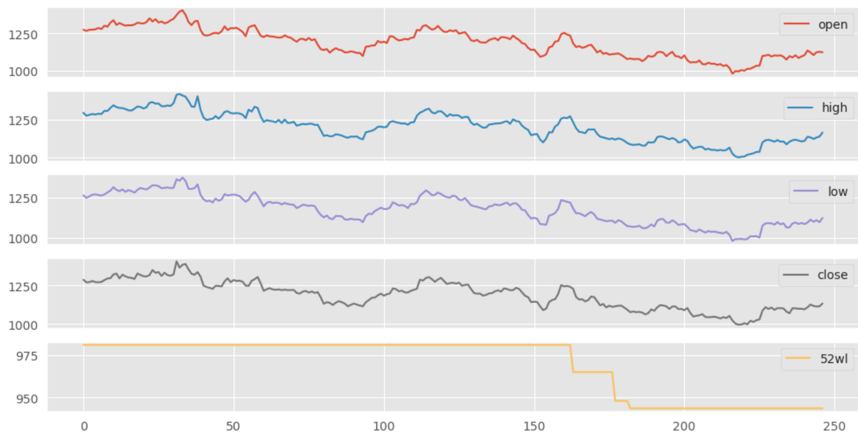

Figure 5: Variation of Tech Mahindra equity price over a year

5.2 Normalization Function

Due to the large size of the data, it is good to scale the data to a certain size, this is called data normalization. This method not only provides the best solution of gradient descent but also improves the accuracy. The normalization function is defined as:

$$Y_t = \frac{Z_t - \text{mean}}{\max - \min}$$

Here, Z_t represents the feature vector for time t, Y_t is the normalized feature vector, and mean, max , and min are the mean, maximum, and minimum values of the feature vector, respectively.

Normalization process is applied to all data before inputting it into the algorithm.

6 CNN-LSTM Framework

We introduced the architecture of CNN-LSTM, a stock market price prediction model based on CNN and LSTM. This model is divided into three main steps: input data representation, extraction of continuous features, and final prediction.

Algorithm CNN-LSTM Algorithm for stock prediction

Input: b is the data of training: c is the data of testing. I is the number of iterations. B is the batch size. Optimizer to be used for stochastic gradient descent
Output : Model evaluation metric : R2-Score, Variance, Max-error
a) Initialize algorithm
b) $b \leftarrow$ Initialize algorithm
c) $P \leftarrow$ (split b in equal parts of B)
d) **for** each round $t = 1, 2, \dots, z$ do:
e) verify, train $\leftarrow P_t, P_{t-1}$
f) tf, vf $\leftarrow$ (generate feature from train and verify)
g) $n_t \leftarrow$ model[Fit($Adam$,tf)]
h) $r_t \leftarrow$ model[Evaluate(n_t,vf)]
i) **end for**
j) $n \leftarrow$ best Model
k) $c \leftarrow$ n
l) $accuracy \leftarrow$ modelEvaluate(n,$test$)

6.1 Input Data

CNN-LSTM aims to predict the future behavior of various businesses by leveraging historical business data for future changes. The model is designed to be applicable across multiple markets, assuming that there exist correct mapping functions from historical data to future market trends. To achieve this, CNN-LSTM collects data from different markets and combines historical market data .In the input data each element represents a piece of information related to the market.

6.2 Continuous Features Extraction

Historical market data, including variables such as close price, open price, lowest price, highest price, and volume, is represented as a series of variables for each day. CNN-LSTM employs a CNN-based approach to extract high-level features from the data. The first convolutional layer merges daily variables into higher-level features, capturing trends and market behavior over time. Subsequent convolutional and pooling layers further aggregate information and generate more complex features to summarize the data.

6.3 Prediction

The advanced features extracted by the convolutional and pooling layers are input into the LSTM unit to extract deeper features. The LSTM unit processes the features and produces a one-dimensional vector, which is then passed through fully connected layers to make the final prediction.

6.4 Proposed CNN-LSTM Process

The process of CNN-LSTM is explained below in detail:

6.4.1 The Input Data

Input of CNN-LSTM is a matrix of dimension 2, where each element represents a piece of information related to the market. The size of the input tensor relies on the number of days and number of variables used for data.

6.4.2 Extraction of Continuous Features

To extract features from the input data,CNN-LSTM uses convolutional layers. The first convolutional layer uses 3x3 filters to combine daily variables into higher-level features. Subsequent layers apply additional filters to capture more complex patterns in the data.

6.4.3 Prediction

The features which are extracted by convolutional layers are given as an input into the LSTM unit in order to extract deeper features. The LSTM unit processes the features and generates a final feature vector, which is used to make the final prediction.

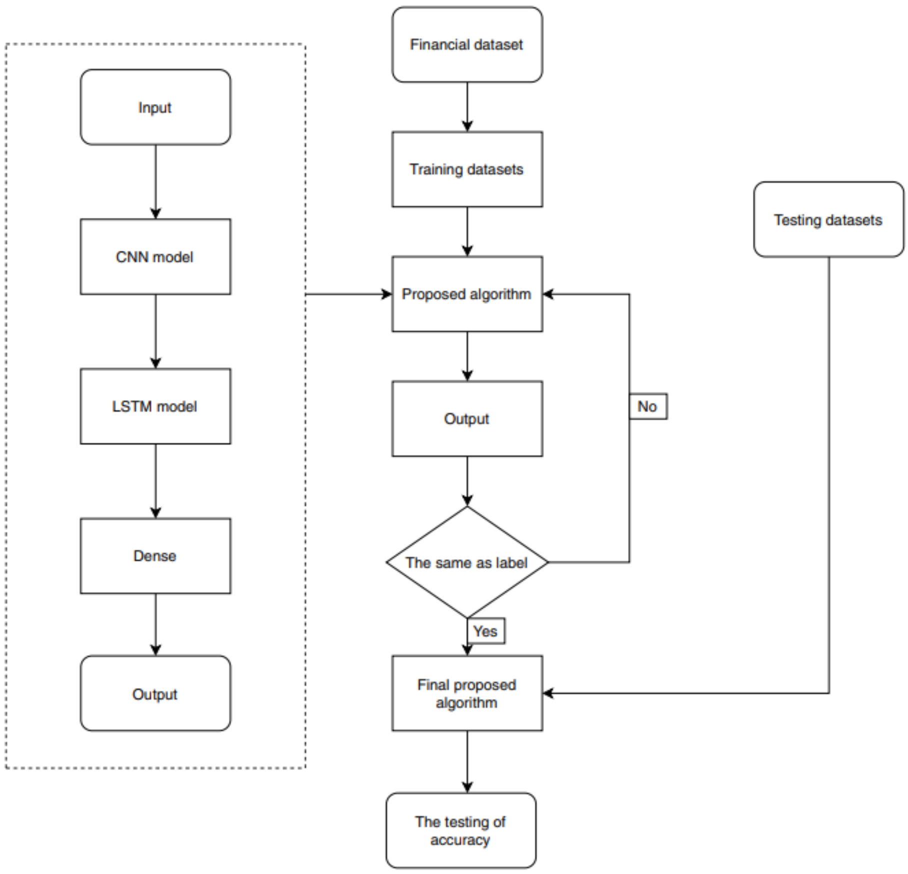

Figure 6: Flowchart of proposed CNN-LSTM

6.5 Example Configuration of CNN-LSTM

The figure 7 below shows how the (CNN+LSTM) process looks like graphically , input Convolutional 3 x 3, Pooling 2 x 2, Convolutional 3 x 3, Pooling 2 x 2, Convolutional 3 x 3, Pooling 2 x 2, Convolutional 3 x 3, Pooling 2 x 2, Output Dense, LSTM, and Pooling 2 x 2 layers.

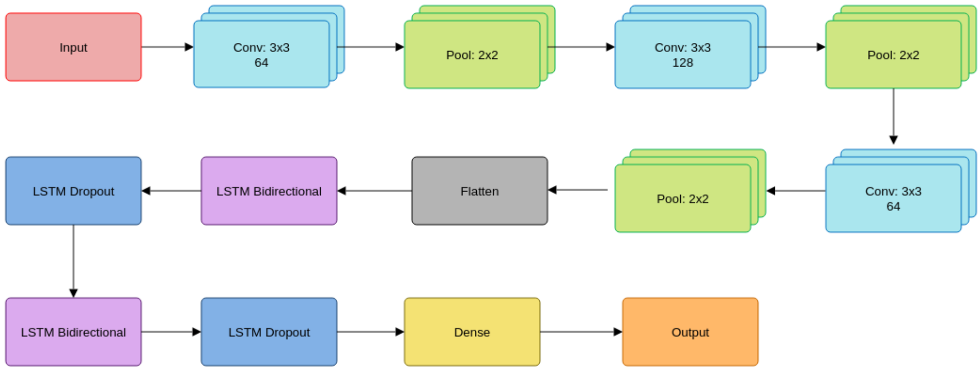

Figure 7: Model line

7 Experimental Results

The project proposed a prediction algorithm for stock market prediction. To validate the performance of the proposed algorithm in the stock prediction market, experiments were conducted to simulate trading predictions using CNN+LSTM. The predictions across six financial stocks are evaluated using three different algorithms (SVM, Random forest and the proposed algorithm). The experiment involves using historical prices as input data and aand predicted using window length that is nothing but the number of days which we are considering for prediction for all comparison algorithms. The proposed framework utilizes the best prediction model, comprising three convolution layers and two fully connected layers.

Below are the tables showing accuracy for different algorithm for different window lengths and bar-plots:

	Window Length	CNN+LSTM Accuracy	SVM Accuracy	Random Forest Accuracy
0	20	0.899774	0.488506	0.482759
1	30	0.870160	0.478788	0.490909
2	40	0.819692	0.433962	0.452830
3	50	0.863173	0.466667	0.400000
4	60	0.886486	0.388889	0.312500

Figure 8: Tech-Mahindra stocks accuracy

	Window Length	CNN+LSTM Accuracy	SVM Accuracy	Random Forest Accuracy
0	20	0.869503	0.584795	0.508772
1	30	0.838080	0.551515	0.690909
2	40	0.839800	0.532051	0.500000
3	50	0.849352	0.653333	0.540000
4	60	0.958249	0.581560	0.617021

Figure 9: SBI stocks accuracy

	Window Length	CNN+LSTM Accuracy	SVM Accuracy	Random Forest Accuracy
0	20	0.886800	0.431034	0.482759
1	30	0.798072	0.460606	0.581818
2	40	0.817496	0.490566	0.490566
3	50	0.921629	0.520000	0.580000
4	60	0.947969	0.479167	0.458333

Figure 10: Idea stocks accuracy

	Window Length	CNN+LSTM Accuracy	SVM Accuracy	Random Forest Accuracy
0	20	0.843877	0.461988	0.456140
1	30	0.796264	0.460606	0.381818
2	40	0.816698	0.576923	0.461538
3	50	0.924799	0.473333	0.460000
4	60	0.925551	0.553191	0.446809

Figure 11: Tata-Motors stocks accuracy

	Window Length	CNN+LSTM Accuracy	SVM Accuracy	Random Forest Accuracy
0	20	0.856659	0.391813	0.368421
1	30	0.859218	0.393939	0.490909
2	40	0.760773	0.442308	0.423077
3	50	0.820144	0.486667	0.620000
4	60	0.905907	0.475177	0.553191

Figure 12: TCS stocks accuracy

	Window Length	CNN+LSTM Accuracy	SVM Accuracy	Random Forest Accuracy
0	20	0.789267	0.568966	0.482759
1	30	0.915473	0.551515	0.563636
2	40	0.893336	0.528302	0.509434
3	50	0.833451	0.513333	0.500000
4	60	0.957952	0.562500	0.625000

Figure 13: Adani-port stocks accuracy

Based on our analysis of above graphs of different stocks , we discovered that for certain window lengths, the accuracy from SVM surpasses that of Random Forest, with both achieving 50% or higher accuracy. However, for other window lengths, Random Forest outperforms SVM across all stocks.

Moreover,we observed that the accuracy achieved by CNN-LSTM consistently exceeds 90%. Thus, overall, CNN-LSTM emerges as the superior model, consistently achieving accuracy levels above 90%. Prior to this, SVM showed better performance, followed by Random Forest, which was the least accurate.

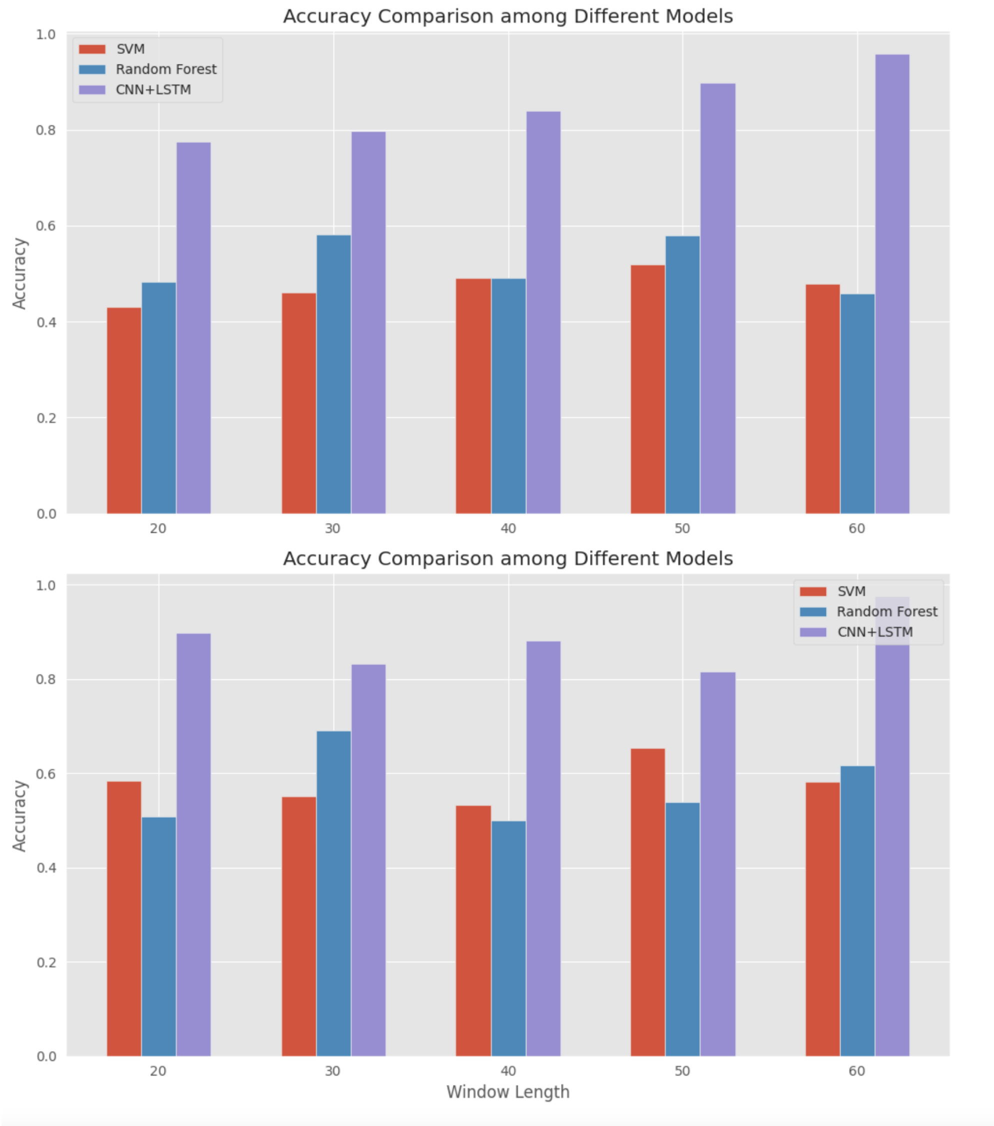

Figure 15: Prediction accuracy of SBI equity

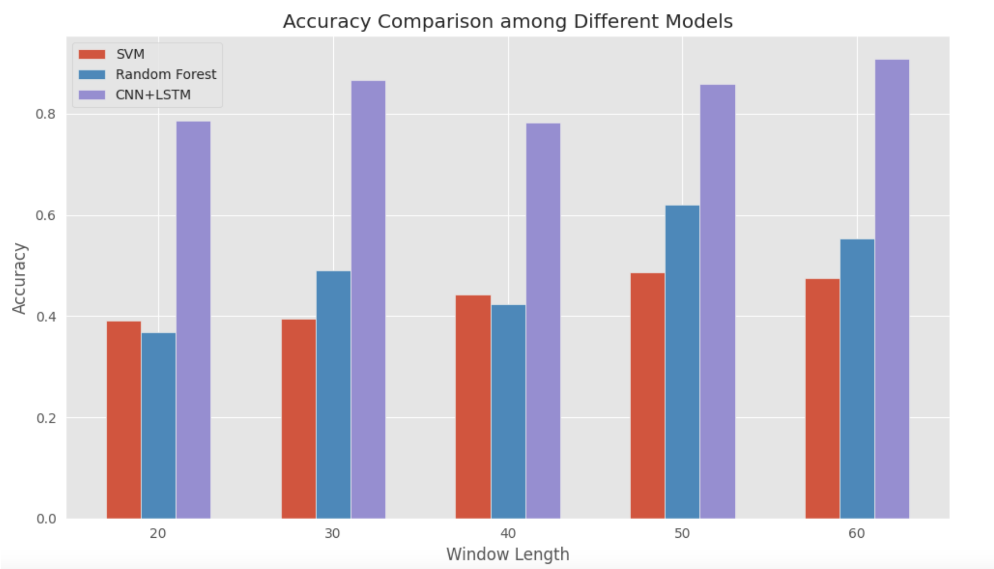

Figure 16: Prediction accuracy of TCS equity

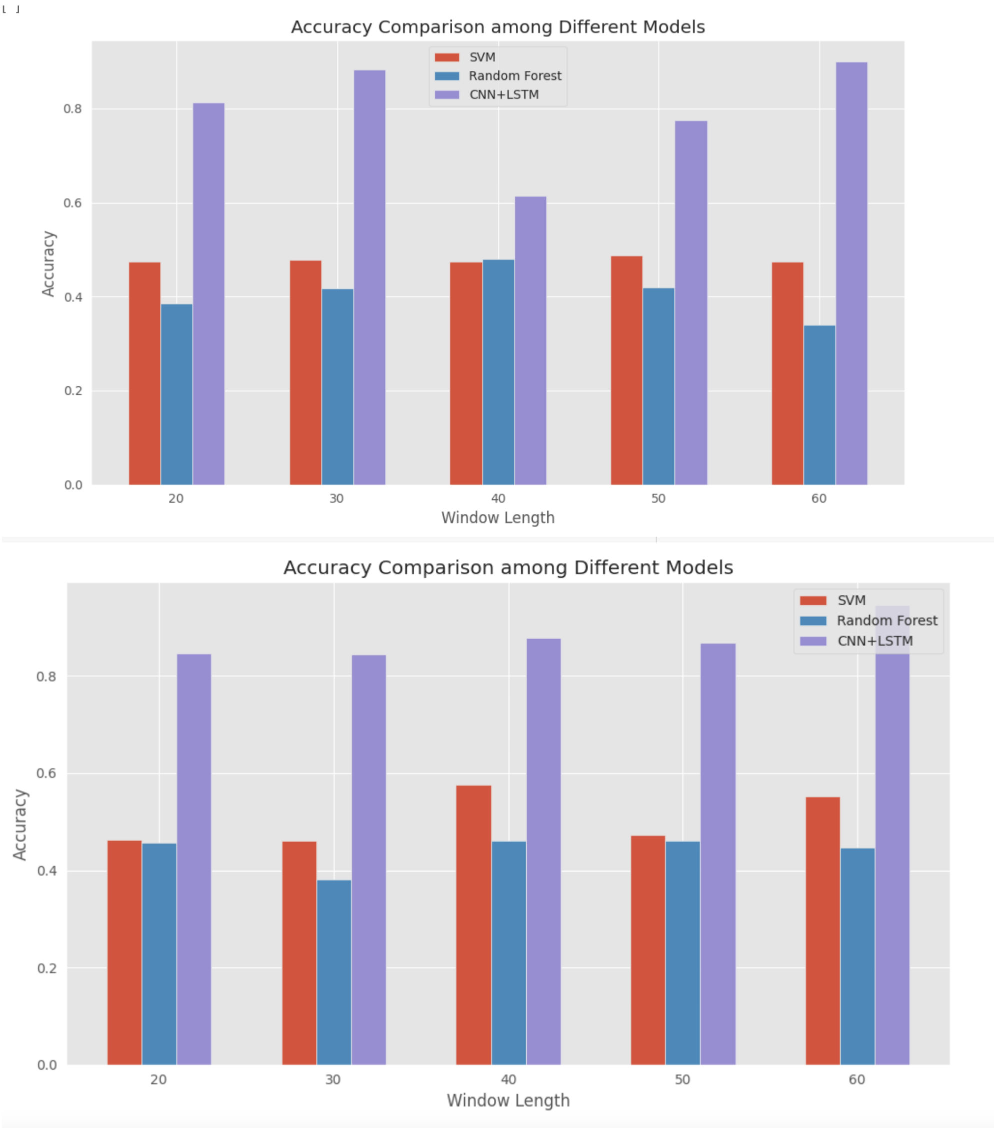

Figure 18: Prediction accuracy of TATA Motor equity

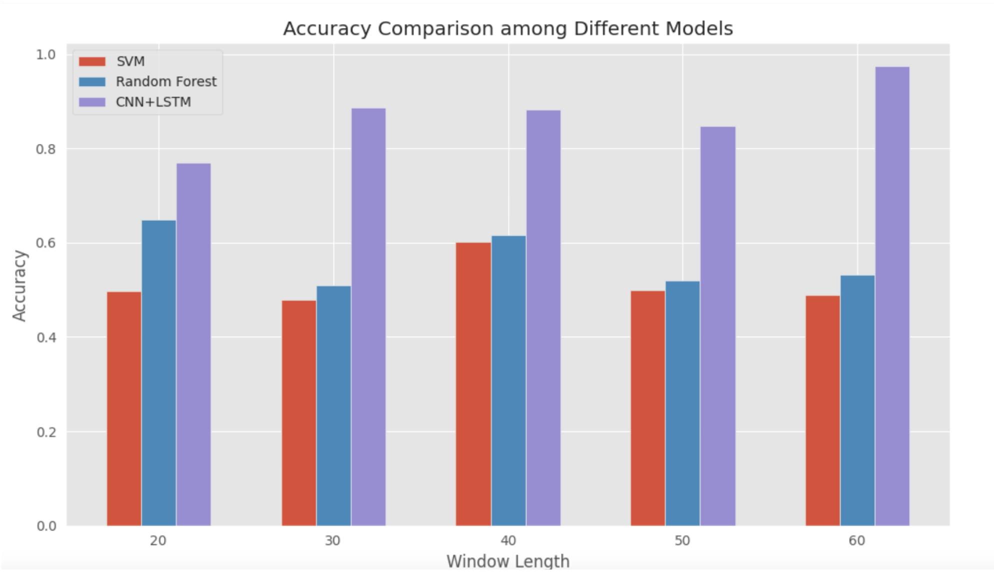

Figure 19: Prediction accuracy of Adani Ports equity

8 Conclusion

The inherent noise and nonlinear behavior of financial markets underscore the complexity of forecasting market trends. This introduces the CNN+LSTM framework, incorporating various news collections, including historical data, to predict stock sequences using convolutional LSTM algorithms. The designed framework extracts financial features using convolutional layers and predicts stock trends through long and short-term memory networks.

The findings from the experiment indicate that the fusion of convolutional and long-short-term memory (LSTM) units within a neural network architecture surpasses conventional statistical techniques as well as traditional CNN and LSTM methodologies in predictive capabilities. Through the integration of data into matrices and employing convolution to capture pertinent features, the CNN+LSTM model mitigates data dispersion and filters out extraneous information, thereby enhancing its efficacy in forecasting stock prices.

Future research directions involve implementing the proposed algorithm into a trading system to further validate its applicability and efficacy. Additionally, the establishment of an expert investment system is envisaged to leverage the predictive capabilities of the proposed framework. Through continued refinement and application, the CNN+LSTM framework holds promise in revolutionizing stock market prediction strategies.

This project's findings underscore the importance of considering proper variables and employing combination of advanced algorithms to reach accurate stock market predictions. By leveraging a combination of leading indicators and sophisticated neural network architectures, the CNN+LSTM framework represents a significant advancement in stock market prediction methodologies, paving the way for more informed and effective investment strategies.

9 Code link

Click https://github.com/Deepaksinghma23m006/stockprice.git

References

[1] Luckyson Khaidem, Snehanshu Saha, Suryoday Basak, Saibal Kar, and Sudeepa Dey. Predicting the direction of stock market prices using random forest. 04 2016.

[2] Xiao Zhong and David Enke. Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, 67, 09 2016.

[3] Jimmy Ming-Tai Wu, Zhongcui Li, Jerry Lin, and Matin Pirouz. *A New Convolution Neural Network Model for Stock Price Prediction*, pages 581–585. 03 2020.

[4] Achyut Ghosh, Soumik Bose, Giridhar Maji, Narayan Debnath, and Soumya Sen. Stock price prediction using lstm on indian share market. 09 2019.

[5] Jimmy Ming-Tai Wu, Zhongcui Li, Norbert Herencsar, Bay Vo, and Jerry Lin. A graph-based cnn-lstm stock price prediction algorithm with leading indicators. *Multimedia Systems*, 29, 02 2021.

[6] Sangeeta Rani and Geeta Sikka. Recent techniques of clustering of time series data: A survey. *International Journal of Computer Applications*, 52:1–9, 08 2012.

[7] Xueqin Zhang, Yuehui Chen, and Jack Yang. Stock index forecasting using pso based selective neural network ensemble. volume 1, pages 260–264, 01 2007.

[8] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, and Victor Chang. An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, 76, 03 2020.

[9] Luca Di Persio and Oleksandr Honchar. Artificial neural networks architectures for stock price prediction: Comparisons and applications. 10:403–413, 01 2016.