STOCK PRICE PREDICTION USING MACHINE LEARNING – AN UNPRECEDENTED APPROACH

Gandhe Sainath[1], Asst Prof. Nandini P[2]

Department of Computer Science and Engineering
The National Institute of Engineering, Mysuru

---***---

Abstract:

In this research, we make an effort to put a machine learning method to stock price prediction into practice. Stock price forecasting uses machine learning effectively. In order to make wiser and more accurate financial decisions, the goal is to forecast stock prices. In order to improve stock forecast accuracy and generate lucrative trades, we suggest a stock price prediction system that combines mathematical functions, machine learning, and other external aspects.

There are two different stock types. You may be familiar with intraday trading through the phrase "Day trading". Interday traders frequently hold securities positions for multiple days up to weeks or months, but at least from one day to the next. Because they have the capacity to store historical data, LSTM's are particularly effective at solving sequence prediction issues. This is significant in our situation since a stock's historical price plays a key role in determining its future price. While predicting a stock's real price is difficult, we can create a model that will predict whether it will rise or fall.

Keywords:

LSTM, CNN, ML, DL, Trade Open, Trade Close, Artificial Intelligence, Stock Market, Machine Learning, Predictions.

Introduction:

Trading in the financial market offers investors the opportunity to buy and sell currencies, stocks, equities, and derivatives through virtual platforms provided by brokers. The stock market allows investors to own shares in public companies, providing the potential for profitable returns without the need for high initial investment or high-risk endeavors like starting a new business or pursuing a high-paying career.

Numerous factors affect stock markets, contributing to the high levels of volatility and unpredictability. Automated trading systems (ATS) powered by computer programs can outperform human traders in submitting orders due to their high momentum. Time-series prediction is a common technique used for stock market prediction, as well as in other applications like weather forecasting. Recurrent Neural Networks (RNN), particularly the Long-short Term Memory (LSTM) model, is currently the most popular algorithm for time-series prediction. Researchers have proposed various models, and this project employs the LSTM model to predict stock prices.

Long short term memory (LSTM) is a model that increases the memory of recurrent neural networks. Recurrent neural networks hold short term memory in that they allow earlier determining information to be employed in the current neural networks. For immediate tasks, the earlier data is used. We may not possess a list of all of the earlier information for the neural node. In RNNs, LSTMs are very widely used in Neural networks. Their effectiveness should be implemented to multiple sequence modeling problems in many application domains like video, NLP, geospatial, and time-series. One of the main issues with RNN is the vanishing gradient problem, and it emerges due to the repeated use of the same parameters, in RNN blocks, at each step. We must try to use different parameters to overcome this problem at each time step. We try to find a balance in such a situation. We bring novel parameters at each step while generalizing variable-length sequences and keeping the overall number of learnable parameters constant.

Existing Methodology:

One of the most significant activities in the world of finance is stock trading. Trying to anticipate the future value of a stock or other financial instrument traded on a financial exchange is known as stock market prediction. The majority of stock brokers employ technical, fundamental, or time series analysis when making stock predictions. Python is the computer language used to make stock market predictions using machine learning. In this article, we suggest a Machine Learning (ML) method that will be taught using the stock market data that is currently accessible, gain intelligence, and then apply the learned information to make an accurate prediction. In this case, this study forecasts stock values using machine learning technology known as LSTM.

Fundamental analysis, which considers a stock's past performance and the general credibility of the company itself, and statistical analysis, which is solely concerned with number crunching and identifying patterns in stock price variation, are traditional approaches to stock market analysis and stock price prediction. Artificial neural networks (ANNs) or genetic algorithms (GAs) are frequently used to accomplish the latter, but they are unable to capture the link between stock prices in the form of long-term temporal dependencies. The phenomena of exploding/vanishing gradient, in which the weights of a big network either grow too large or too tiny (respectively), greatly slows their convergence to the ideal value, is another significant problem with utilizing simple ANNs for stock prediction. This is typically caused by two factors: weights are initialized randomly, and the weights closer to the end of the network also tend to change a lot more than those at the beginning.

Proposed Methodology:

To offer effective stock price prediction, we suggest using the LSTM (Long Short Term Memory) algorithm. Long-Short Term Memory refers to a particular kind of RNN that has the capacity to learn long-term dependency (LSTM). RNN can remember long-term inputs thanks to LSTM. contains data in memory, much like a computer's memory. It has the capacity to read, write, and delete data from its memory. This memory can be thought of as a closed cell with a closed description that makes decisions about what information to store or erase. Three "gate" structures make up the unique network structure known as LSTM. An LSTM unit has three gates: an input gate, a forgetting gate, and an output gate. Information can be chosen by rules when it enters the LSTM network. Information that does not comply with the algorithm will be erased by the forgetting gate, leaving only the data that does. The historical data that were collected from the Internet and used as experimental data in this study. The experiments made use of three data sets. It is necessary to find an optimization algorithm with a quicker convergence rate and fewer resource requirements.

• Used the LSTM neural network with automatic encoder and the embedded layer of LSTM.

• To prevent gradients from exploding and vanishing, LSTM is employed in place of RNN.

• In this project, the model is trained in Python, and the input dimensions are reduced using MATLAB. Data is stored and retrieved using MySQL as a dataset.

• The historical stock data table contains the information of opening price, highest price, lowest price, closing price, transaction date, volume and so on.

The ability of LSTM to read intermediate context is its key benefit. Without specifically using the activation mechanism contained within the repeating components, each unit retains information for a lengthy or brief amount of time. A crucial point is that any cell state is only replicated upon forget gate release, which ranges from 0 to 1. In other words, the hardware and operation of the cell state activation are both handled by the forgetting gateway in the LSTM cell. As a result, instead of explicitly increasing or decreasing in each step, the data from the preceding cell can flow through the unmodified cell, allowing the instruments to adjust to their proper values over a finite period of time.

System Design:

Algorithm of LSTM =

Input: Historical stock price data.

Output: Prediction for stock prices based on stock price variation.

1. Start.
2. Stock data is taken and stored in a NumPy array of 3 dimensions (N, W, F), where:

• N is number of training sequences,

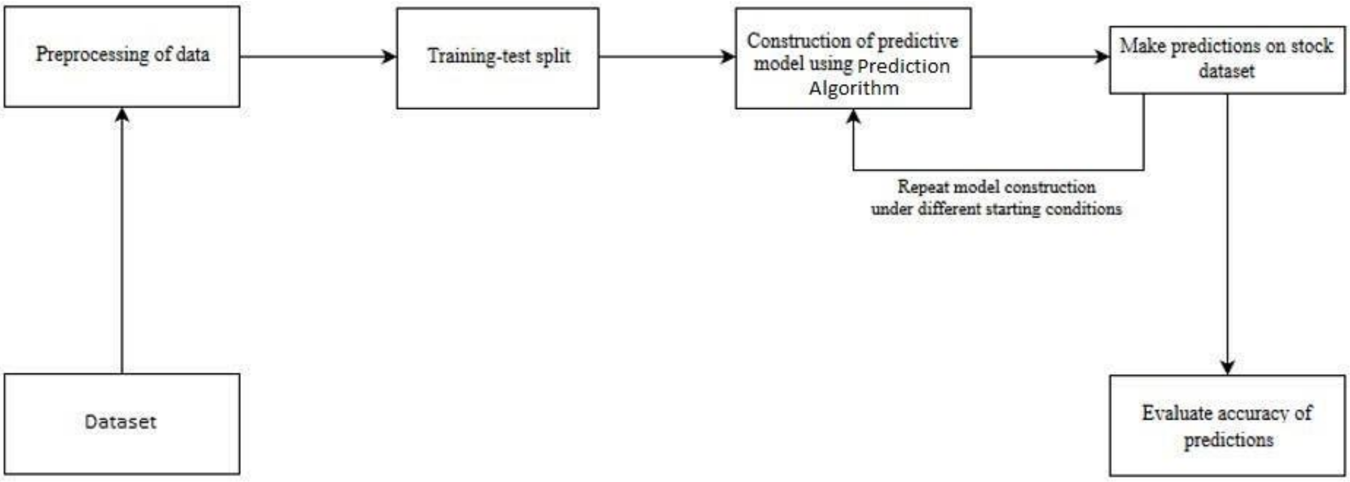
• W is sequence length,

• F is the number of features of each sequence.

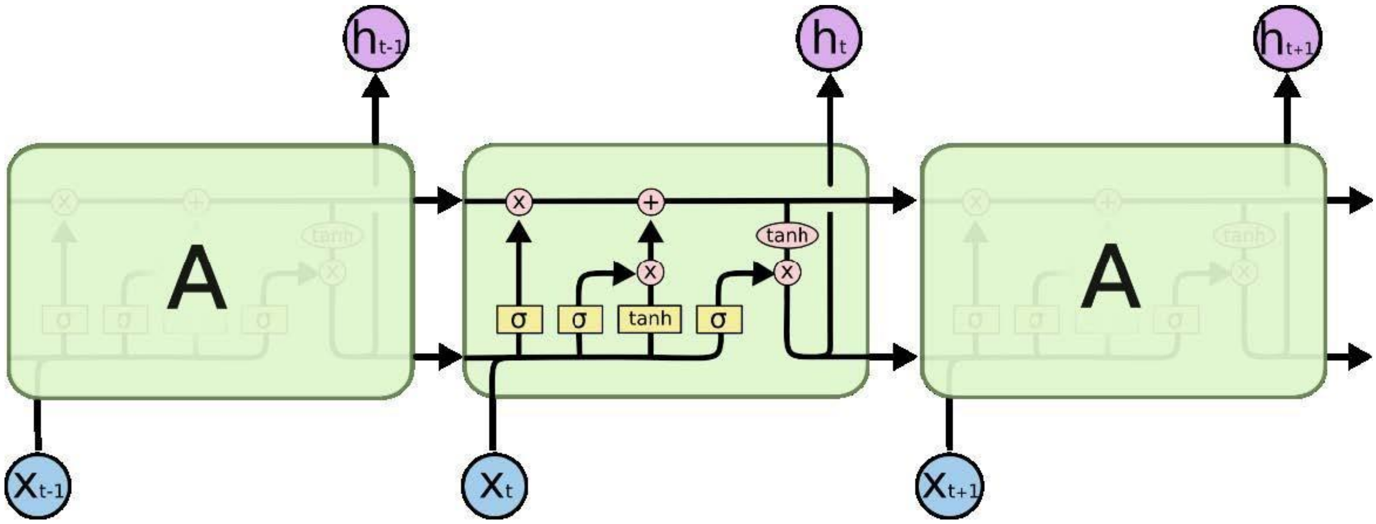
3. A network structure is built with [l, a, b, l] dimensions, where I is the input layer, a neurons in the next layer, b neurons in the subsequent layer, and a single layer with a linear activation function.
4. Train the constructed network on the data.
5. Use the output of the last layer as prediction of the next time step.
6. Repeat steps 4 and 5 until optimal convergence is reached.
7. Obtain predictions by providing test data as input to the network.
8. Evaluate accuracy by comparing predictions made with actual data.
9. End.

Preprocessing of data =

Overall Architecture =



If we pay attention to a real-world event, we notice that in many circumstances, our final output depends not only on the external inputs but also on earlier output. In a classical neural network, final outputs rarely operate as an output for the following stage. Recurrent neural networks (RNN) are designed to get around this constraint. RNNs are networked with internal feedback loops to enable information persistence. The RNN first produces an output of ht for some input Xt (at time step t). The RNN uses two inputs, Xt+1 and ht, to produce the output ht+1 at the subsequent time step (t+1). A loop enables data to be transferred from one network phase to the next.



Recurrent neural networks are in the form of a chain of repeating modules of the neural network. In standard RNNs, this repeating module has a simple structure like a single tanh layer as shown above. LSTMs adhere to this chain-like structure, but the repeating module is structured differently. There are four levels instead of just one, and they interact in a highly unique way as demonstrated above. Each line represents a complete feature vector, from a node's output to other nodes' inputs.

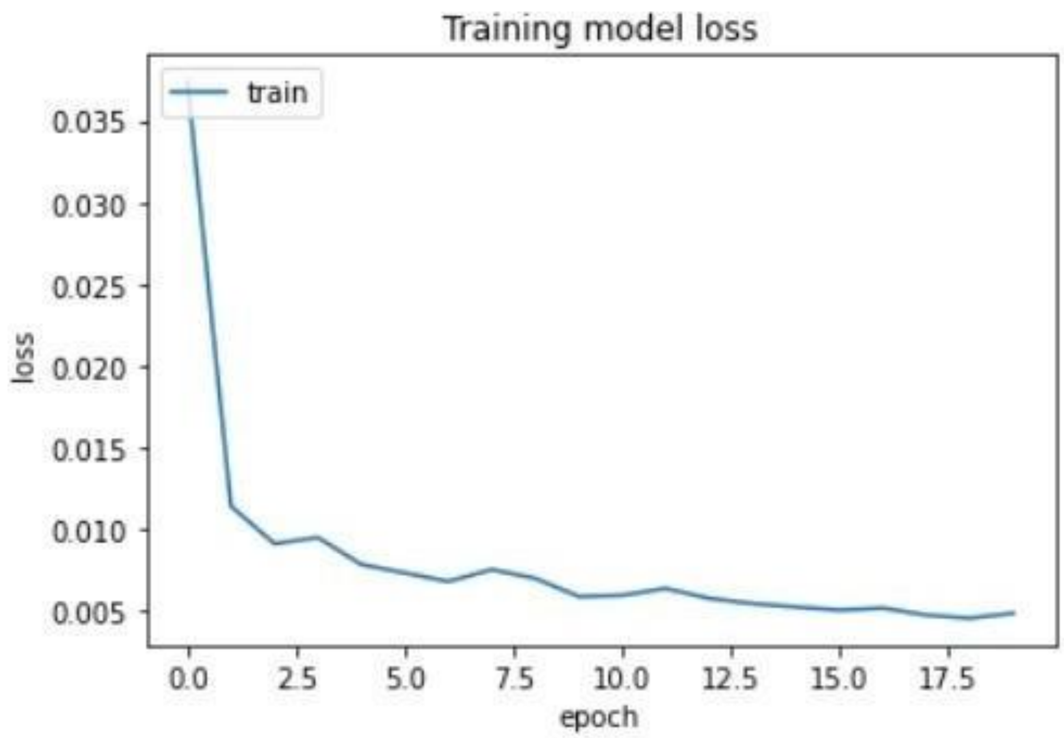
The yellow boxes are learnt neural network layers, whereas the pink circles are pointwise operations like vector addition. Concatenation is indicated by lines merging, but a forked line indicates that its content has been replicated and is being sent to other destinations.

System Testing & Results:

To train and test the machine, we read in the google dataset, preprocess it like we would in any other machine learning task, specifically, we are:

- Choosing which data should be left out of a cell at a given time step is the first stage in LSTM.
- The second layer has two tasks to do. Which values are accepted is determined by the sigmoid function (0 or 1). The tanh function assigns the values passed weight by determining their relevance on a scale of -1 to 1.

- The third phase entails choosing the final product. Run a sigmoid layer first, which chooses which components of the cell state are sent to the output. The cell state must then be multiplied by the output of the sigmoid gate after being passed through the tanh function to push values between -1 and 1.



We can see that the loss starts high, then quickly goes down, and starts approaching zero. This gives usan insight into how the machine learns.

The above graph depicts the actual stock price and the predicted stock price.

Conclusion and Future Enhancements:

In this paper, we examine the development of businesses across several industries in an effort to determine the ideal window of time for estimating future share prices. The crucial inference from this is that businesses in a particular industry have similar dependencies and growth rates. If the model is trained on additional data sets, the forecast may be more accurate. Nevertheless, there may be some room for specific business analysis in the case of share prediction. To improve accuracy, we may examine the various share price patterns of various industries and evaluate graphs with a wider range of time periods. This framework largely aids in market analysis and growth projections for various organizations across various time frames. The accuracy of the prediction may be increased by include additional variables (such as investor sentiment, the outcome of an election, or geopolitical stability) that are not connected with the closing price.

References:

- Jupyter Project documentation : https://bit.ly/3sFi2Ma
- Python documentation : https://bit.ly/37PQnAU
- Simpli learning tutorials : https://bit.ly/3sE9Yvb
- Machine learning Algorithms : https://bit.ly/3yGLl4I
- NumPy documentation : https://bit.ly/3wtKMKl
- Pandas documentation : https://bit.ly/3LmnkTo
- Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, Giridhar Maji, Narayan C. Debnath, Soumya Sen
- S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017
- Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM” in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015.