ISSN:0976-0172

Ch. Sravan Kumar, JBio sci Tech, Vol 13(2),2025, 01-07

Journal of Bioscience And Technology
www.jbstonline.com

Forecasting Global Stock Market Indexes Building Neural Networks (ANNs) using Python

[1] Ch. Sravan Kumar, [2] P. Samatha,

[1]Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Article Info

Received: 30-04-2025 Revised: 16-06-2025 Accepted: 28-06-2025

Abstract—

One of the most effective avenues for monetary growth that need very precise trade prediction is the stock market. To get the most out of this topic, you need to have some technical knowledge and expertise. In order to provide qualified and quantified price predictions with high accuracy and near real-time precision, this article describes a Python console application that is customized using Neural Networks (NN) and Artificial Intelligence (AI). This article introduces new concepts that integrate a security shell that uses voice and PIN authentication with an AI and NN model in a Python console system. It works with several platforms and lets you forecast the price of cryptocurrency. The user may get a copy of the finished data via email using this software. The method that has been suggested shows how AI and ML have affected forecasts for the near future. Any field that makes use of historical databases may benefit from this method.

Index Terms—

Global Stock Prediction using Artificial Neural Networks (ANNs), Long Short-Term Memory (LSTM), and Python.

I. INTRODUCTION

It may be more difficult to forecast the indices in light of recent developments in the stock markets, which have had substantial impacts on finance. These days, almost everyone has some kind of connection to this field, and as technology advances, more and more people will need to understand and anticipate indexes, which will pique their interest in index prediction. Stock price predictions, however, become more difficult as a result of the rapid fluctuations in stock prices. This intricacy has been further amplified by the implications of cryptocurrencies [1]. Traders avoid relying on fundamental research to forecast prices because of these issues and instead employ sophisticated systems. Traders may increase their profit significantly by selling the index before its value drops or buying it before its price increases. Furthermore, traders may find it hard to believe that intelligent systems can supplant their expertise and experience; however, with the abundance of data made possible by algorithms, pattern recognition, and Artificial Intelligence, it appears reasonable to employ and even combine these with human expertise. One of the most effective algorithms and models used to predict how the stock market will act is the Neural Network (NN) [2], which has grown in popularity due to the importance of precise data. One well-liked approach to financial market prediction using technical analysis is the Artificial Neural Network (ANN). Long Short-Term Memory (LSTM) is a very useful strategy in this field [3]. The future recurrent patterns of a stock index may also be predicted using pattern recognition [4]. Having a model like NN that predicts stock prices so close to reality is very respectable in the trading industry. To have an accurate forecast, it is necessary to have

ISSN:0976-0172

Journal of Bioscience And Technology
www.jbstonline.com

expertise and refresh oneself with the process of predicting the stock market. But since it's qualitative, this procedure can't guarantee anything. By transforming it into a scalar number, AI and NN make it amenable to mathematical methods; this, in turn, yields a value with high accuracy and modest Root Mean Squared Error (RMSE) values, making it more trustworthy than qualitative forecasts for the future[5, 6]. The lack of implementation on an OS program means that not all users can take use of this strategy, despite its usefulness. A realistic Neural Network that can store information for later use to improve efficiency and modeling is the Real-Time Recurrent Learning (RTRL) network [7]. The usefulness of RNN-based algorithms for financial market prediction is shown in [8]. This article presents a software that can be dynamic and versatile, allowing the user to modify the mark in a fraction of a minute. Integrating AI with Python programming opens up new possibilities, making it accessible to more people and perhaps more effective than earlier approaches. The NN is a machine learning method that may be used to statistically forecast stock market movements; it learns how to do certain tasks by examining training samples; and it is the foundational case for prediction. This work utilizes this straightforward characteristic to develop intricate methods for precise prediction that approach the actual value. Here is the structure of the paper: Preliminaries and tools are provided in section 2. Section 3 presents the suggested framework. Section 4 presents the outcomes. Conclusion and additional improvements are drawn in section 5.

II. NEURAL NETWORKS AND PROGRAMMING LANGUAGE

The fundamental premise of a forecast is often grounded on facts that have already been examined and verified. Consequently, results may be helpful for forecasting data in the future. Results from predictions made using a neural network are often more accurate estimates of real values since the model may be fine-tuned via model performance by reducing mistakes and tolerances. Python runs our application. That runs a plethora of programs, including those for education, science, and business; software; online and internet development; and desktop GUIs. One such Python package is Keras, which is used for Neural Networks. Fast trial operations using artificial and Deep Neural Networks are part of its architecture, and it is competent to supervise Tensor Flow. SKlearn is a Python package that includes many machine learning functions. It is built to work in tandem with NumPy and other Python numerical science libraries, and it places an emphasis on prismatic techniques for classification, regression, and constellations, such as support for vector machines and database scans. Python programmers may take use of NumPy, a professional math library, to work with complex arrays and matrices that are big and multidimensional. With a large library of mathematical functions designed for use with matrices and arrays at a high level. The Pandas library is a data structure toolkit for the Python programming language. When it comes to object-oriented interfaces and Python's Graphical User Interface (GUI), Tkinter is the gold standard. It finds frequent use in graphical user interface and object-oriented software applications. For the Python programming language, Wx Python provides a wrapper for the cross-platform GUI API (Graphical User Interface Application Programming Interface) toolkit and its C++ widgets. It is a Python extension module (native code) that offers an alternative to Tkinter GUI. Matplotlib is a package that the Python programming language uses to create plots. It is compatible with the scientific and numerical libraries NumPy and SciPy. For developers working with GUI toolkits like Tkinter or wxPython, it offers an OO (Object-Oriented) API that may be used to include plots into their programs. For financial data visualization and analysis, you can't do better than Mplfinance, a Matplotlib utility package. Candlestick charts in Python make extensive use of it. Python programmers may access the operating system with the OS (Operating System) module. It is one of the standard utility modules in Python. You may access features that are reliant on the operating system in a portable fashion using this module. Numerous functions for interacting with the file system and the Python environment are included in the 'OS' and 'OS.path' modules.

III. PROPOSED FRAMEWORK

Every part of this Python-based system is devoted to making predictions about the direction of international stock prices using an attention-based short-term and memory model. There are a total of five levels in the model: the shell, input, hidden, attention, and output layers. User authentication is handled by the shell layer. In order to fulfill the input requirements, the input layer reads data from the input. The LSTM unit establishes a correlation between the hidden layer and the linear network. Based on the predictions made in the hidden layer,

ISSN:0976-0172

Journal of Bioscience And Technology
www.jbstonline.com

the attention layer prepares future quantities. The final measured findings are sent to the output layer so they may be shown to the user. Figure 1 shows the suggested framework, while Figure 2 shows the LSTM technique diagram. Data is retrieved from the database using Pandas and organized using the NumPy package, as shown in Fig. 2, following the user's submission of input criteria. After that, it is processed using the LTMS method for prediction after being rounded and evaluated using the Python AI packages Keras, TensorFlow, and SKlearn. Thus, the user will be able to see the train plot, forecast plot, and future pricing.

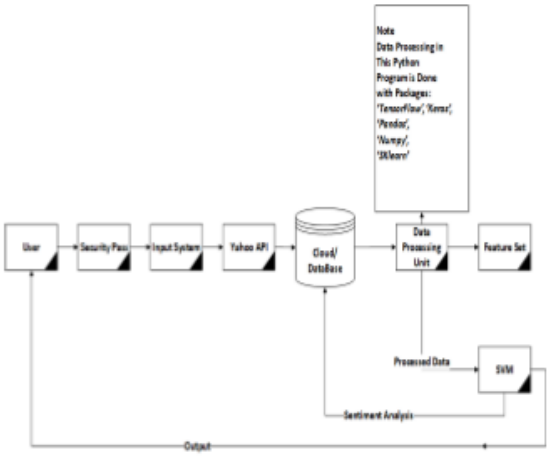

Fig. 1, Proposed Framework that illustrates the process from the initial

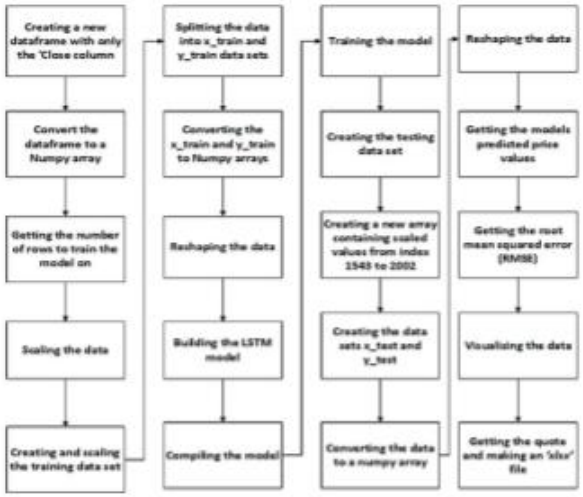

Fig. 2, Diagram of the prediction method based on the LSTM.

The first layer is the shell. Following successful authentication at the shell layer—which may be accomplished using either a PIN or a passvoice—the user is allowed to proceed to the next tier. See Figure 3 for a schematic of this concept.

Fig. 3. Shell layer algorithm

B. Layer for Input Data such as initial, final, future, and major credit may be entered into the input layer after the shell layer. The user's task is completed at this point; as indicated at the entrance, data is retrieved from the database using the Pandas package and organized using the NumPy package. The data was then divided into a train set and a test set, with the former being sent on to the second layer. In Figure 4, you can see the layer diagram.

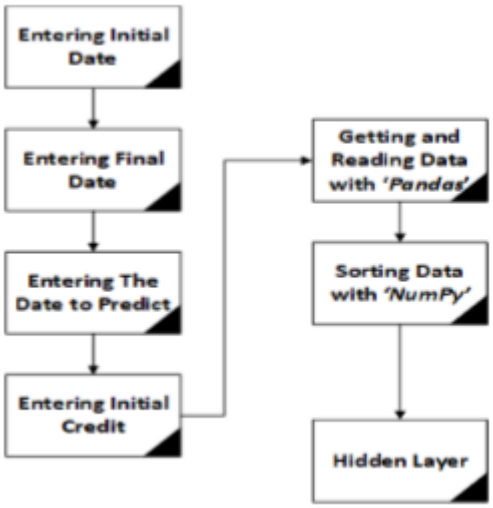

Fig. 4, Input layer block diagram

C. The Secret Layer

ISSN:0976-0172

Journal of Bioscience And Technology
www.jbstonline.com

To put it simply, the Hidden Layer is the program's meat and potatoes. In this layer, you can find all the analysis and predictions made using the LTMS method and the RMSE equation. Figure 5 offers a schematic of the hidden layer.

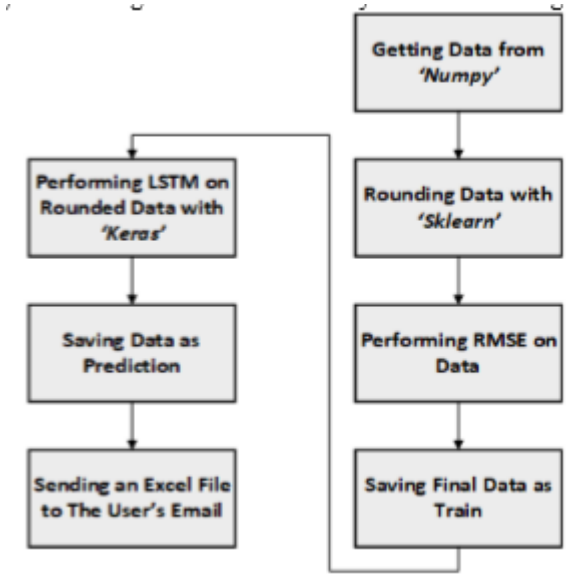

Fig. 5, Hidden layer block diagram

Figure 5 shows that after the calculation is complete, the user's email address specified in Shell Layer is emailed an Excel file with dates, prices, and other relevant information. The concealed layer is enhanced and made more efficient as a result. Figure 4 shows that RMSE is used in error and tolerance prediction. As stated in [9]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(Y_{i,Predicted} - Y_{i,Actual}\right)^2}{n}} \qquad (1)$$

The expected price and the actual price are represented by and, respectively, in Equation (1), where n is the entire sample size of the data. A lower RMSE indicates less error. A zero RMSE is more common when the discrepancy between the inputted beginning data and the outputted final data is large. Consequently, the forecast is more in line with reality. The section on Experimental Results proves this idea. Here is the Python code to compute the RMSE:

```
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
```

Level D: Attention The attention layer is responsible for processing the hidden layer's final computed data. The method works by observing and learning from input data analyzed in the attention layer, which in turn trains the model. The training model gets closer to the goal value as the weights of the data it uses increase. This allows us to forecast the price for a future date. The results of the trials and the weights assigned to the data are used to make predictions that are quite close to the real. Figure 6 is a representation of the attention layer.

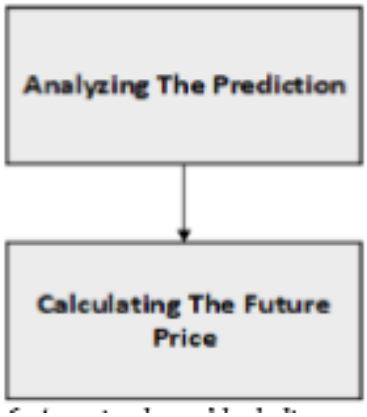

Fig. 6, Attention layer block diagram

E. Layer for Output Finally, the system's Output Layer displays future price, profit, and worth along with associated charts. Achieving a profit and future value is determined by:

$$i = ln\left(\frac{train["close"][int(End\_Date)]}{train["close"][int(Start\_Date)]}\right) \qquad (2)$$

$$F = (i + p)^t \qquad (3)$$

Future worth (F), primary credit (P), compound interest (i), and period count (t) are all given here. Yearly periods are meant to be counted as the number of periods (t) as:

$$t = (Year_{final}) - (Year_{initial}) \qquad (4)$$

The data is prepared to be processed and plotted using the matplotlib tool, as per the code. Figure 7 shows the Output Layer diagram.

ISSN:0976-0172

Journal of Bioscience And Technology
www.jbstonline.com

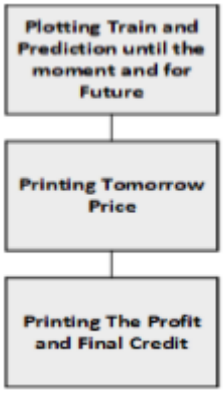

Fig. 7, Output layer block diagram

Besides the plot findings, this layer also displays others, such as interest, future value, quotes, and more.

IV. EXPERIMENTAL RESULTS

There are three iterations of testing the model and software with various data sets for AMZN stock, as shown in Table 1. Nine, ten, eleven, twelve, and thirteen figures show the outcomes.

Table 1

Primary year	Final year	Prediction Date	Range
2010	2020	2020/02/24	10
2018	2020	2020/02/24	2
2019	2020	2020/02/24	1

Fig. 8, Periodic Graph for t = 10 years

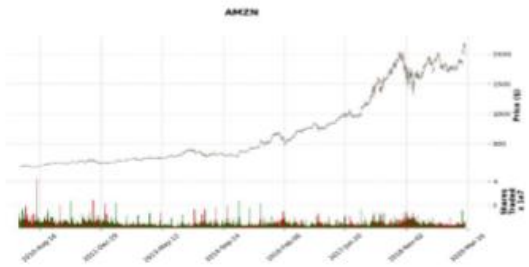

Fig. 9, Candlestick Graph for t = 10 years

Fig. 10, Periodic Graph for t = 2 years

ISSN:0976-0172

Journal of Bioscience And Technology
www.jbstonline.com

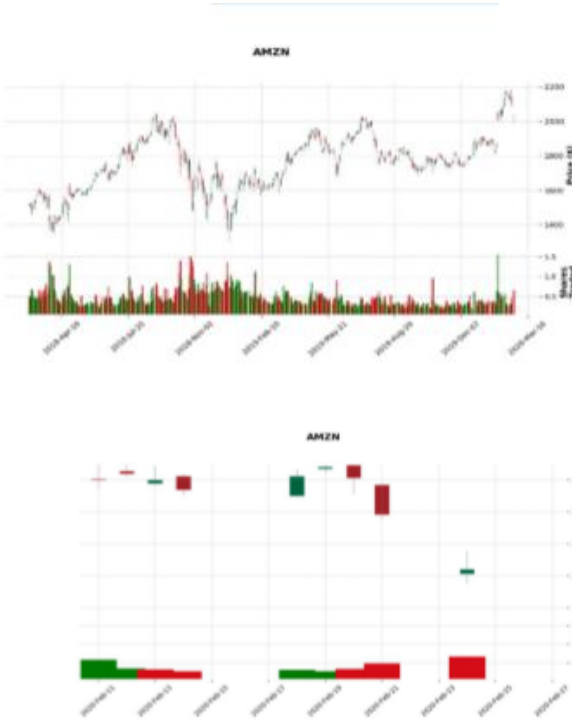

Fig. 11, Candlestick Graph for t = 2 years

Fig. 12, Periodic Graph for t = 1 year

Fig. 13, Candlestick Graph for t = 1 year

The large quantity of received data is responsible for the low RMSE, as seen in Figures 9 and 10. Because of the large gap between the inputted years (2010-2020), the plot of the years before 2020 is not very impressive. Consequently, the LSTM algorithm's forecast is both accurate and very close to reality. This means that the forecast and price are quite close. The RMSE is within the typical range, as shown in Figures 11 and 12, since the quantity of data received is normal. The sum of the years 2018–2020 that were input. Consequently, the plot from 2018 to 2019 is in a precarious position. In most cases, the LSTM algorithm's forecast is quite close to the actual result. The large RMSE is a result of the little quantity of received data, as seen in Figures 11 and 12. This year's plot (2019) is at a very elevated position due to the little difference between the inputted years (2019 and 2020). The LSTM algorithm achieves a lower forecast than the real one. The offered model is more efficient than other approaches and models since it has two more characteristics. In addition, the model may run on any platform since it is built on a Python terminal software. You may find further details in Table 2.

Table 2: Comparison of the proposed approach and the others

ISSN:0976-0172

Journal of Bioscience And Technology
www.jbstonline.com

Item	Proposed Program	Other ones
An email of duplicated data	Sends to user	Not able
Cross-Platform capability	Able and Lighter	Able and Heavier
Operating System	Python 3.7	various
Security Layer	Included	Not Included
An Excel data file	Generates	Do not Generate

V. CONCLUSION

This article presents the results of a study using the deep learning and AI frameworks in Python for stock prediction based on LSTM machine learning, AI, Python programming, and the international stock market. In this experiment, we draw the prediction graph for AMZN stock at three separate periods using the algorithm that we used to forecast its future price. The findings confirm the program's and the prediction's viability and accuracy in this sequence. The experiment confirmed that the suggested Python program was accurate when compared to predictions made using traditional methods. Consequently, this approach has the potential to be expanded to several domains, including those dealing with finance and education. Here are seven ways in which this study is innovative: Cryptocurrencies like Bitcoin (BTC) and Ethereum (ETH) may have their prices predicted by the suggested application. (2) This paper's model is fine-tuned using a versatile approach, so its accuracy improves as the user inputs more varied data. (3) A secure layer, the Shell Layer, is allowed in this paper's approach; so, the user should be confirmed via voice or PIN. (4) Candlestick graphs may be plotted using the provided data using the suggested application. (5) The user's entered email is used to send an Excel file with the 'xlsx' format when the data is processed and analyzed to turn it into information. 6) Anyone, from stock market experts to scientists, may utilize the Python software created using the paper technique. (7) The software that has been suggested may forecast a wide range of topics using past numerical data, like the price of a home or a vehicle.

VI. REFERENCES

[1] D. Wei, "Prediction of Stock Price Based on LSTM Neural Network," in 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019, pp. 544-547.

[2] M. Abdechiri, K. Faez, and H. Bahrami, "Neural network learning based on chaotic imperialist competitive algorithm," in 2010 2nd International Workshop on Intelligent Systems and Applications, 2010, pp. 1-5.

[3] T. Mankar, T. Hotchandani, M. Madhwani, A. Chidrawar, and C. Lifna, "Stock market prediction based on social sentiments using machine learning," in 2018 International Conference on Smart City and Emerging Technology (ICSCET), 2018, pp. 1-3.

[4] L. Sayavong, Z. Wu, and S. Chalita, "Research on Stock Price Prediction Method Based on Convolutional Neural Network," in 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), 2019, pp. 173-176.

[5] M. Billah, S. Waheed, and A. Hanifa, "Stock market prediction using an improved training algorithm of neural network," in 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), 2016, pp. 1-4.

[6] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," Computers & Industrial Engineering, p. 106435, 2020.

[7] M. T. Motlagh and H. Khaloozadeh, "A new architecture for modeling and prediction of dynamic systems using neural networks: Application in Tehran stock exchange," in 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA), 2016, pp. 196-201.

[8] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," Procedia Computer Science, vol. 170, pp. 1168-1173, 2020.

[9] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," Procedia Computer Science, vol. 167, pp. 599-606, 2020.