International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 6, Nov-Dec-2023, ISSN (Online): 2395-566X

Neural-Market Dynamics: Unveiling Future Trends with CNN-LSTM Ensemble for Stock Price Forecasting

Madhur Narang, Kushagra Sahani,
Asso. Prof. Dr. Neha Agrawal, Asst. Prof. Ms. Meenu Garg
(Department of Information Technology, Maharaja Agrasen Institute of Technology)

Abstract- The stock market is the platform where anyone can buy and sell or trade shares of public companies, and for that predicting the stock price helps us to forecast the future value of the company shares, derivatives, and mutual funds. So, while doing predictions of the stock market we have to keep some key points in our mind such as No one can accurately predict the future movement of the stock market because the stock market is a composite and volatile system, and many factors can affect its performance.

To evaluate a company's financial stability and performance, fundamental analysis is used. On the other hand, for reviewing historical price and bulk data, technical analysis has been carried out to recognize tendencies and patterns. Risk management, while investing in the stock market carries inherent risks, and to mitigate those risks, it is crucial to spread out investments and establish stop-market orders, and other techniques.

The aim of this paper is to suggest deep learning techniques in order to predict the stock prices of different companies such as AAPL(Apple), BAM (Brookfield Asset Management), and UBER and using two different models such CNN (Convolutional Neural Network) in CNN the paper uses One -Dimensional CNN (1D CNN) and LSTM (Long Short-Term Memory) uses Bidirectional LSTM.

Index Terms- Deep Learning, CNN, LSTM

I. INTRODUCTION

There has been significant research on predicting future stock prices, although supporters of the efficient market hypothesis argue that it is not feasible to predict them. However, various researchers have shown that reliable statistical, econometric, and deep learning models can produce precise predictions based on the careful selection of variables and suitable functional forms. Stock market prediction involves using various analytical techniques to forecast the future price movements of stocks, and it is a complex and challenging task. No can accurately predict the future movement of the stock market. It is necessary to understand that the stock market is a complex and volatile system, and many factors can affect its performance. Therefore, no one can accurately forecast the future stock market movement with complete accuracy. Fundamental analysis involves analysing the financial health and performance of a company, such as its earnings, revenue, assets, liabilities, and management [1]. It is possible to forecast future stock prices using this knowledge. In technical analysis, trends and patterns that may be predictive of future price movements are found by examining historical prices and bulk data. Professionally equipped technical analysts use tools such as charts, plots, and graphs to visualize the trends in data in order to predict the future prices. Researchers have created complex yet sophisticated algorithms that can analyse enormous bulks of data to find patterns and forecast future prices, thanks to the advancement of machine learning and artificial intelligence. Stock market investing has inherent risks, which should be managed by investment diversification, stop-market orders, and other techniques.

II. STOCK PREDICTION METHODOLOGY

1. CNN

In 1998 Leun et al. proposed a kind of deep neural network known as Convolutional Neural Networks (CNNs). CNN is

© 2023 IJSRET

International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 6, Nov-Dec-2023, ISSN (Online): 2395-566X

primarily used for image and video classification tasks. The key idea behind a CNN is to use convolutional layers to automatically learn feature hierarchies from raw input data, such as images. It is effective in forecasting of time series applications. CNNs typically have a sequence of convolutional layers, then pooling layers, and finally fully linked layers. The input image is subjected to filters (kernels) by convolutional layers, which generate feature maps emphasising certain characteristics of the image. The feature maps are then down sampled by the pooling layers, which reduces their size while keeping the crucial data. The output of the final pooling layer is then fed into a typical neural network architecture for classification by the fully connected layers.

The neural network type used in this paper, known as one-dimensional convolutional network (1D CNN), works with 1-dimensional data like time series or signals.

In a 1D CNN, convolutional layers apply filters to a sequence of input data, creating a set of feature maps that capture local patterns in the data. The next step is to send these feature maps through a pooling layer, which lower their dimensionality while retaining crucial data. At the end, one or more fully connected layers that handle classification or regression tasks are fed the output of the pooling layer.

2. LSTM

The traditional RNNs have some limitations in handling the long - term dependencies in sequences. To overcome this, Hochreiter and Schmid Huber introduced Long Short-Term Memory (LSTMs), in 1997.

Depending upon the input and the internal state the LSTMs selectively forget or remember information that is store into their memory cell. The gates control the memory cell, which are composed of sigmoid neural network layers that determine how much of the input should be remembered, how much should be forgotten, and how much should be outputted. These gates provide LSTMs the ability to selectively recall or forget information depending on how relevant the past events are to the current task.

Bidirectional LSTMs are recurrent neural network architecture that processes input sequences in both forward and backward directions. In a standard LSTM, the information flows only in the forward direction, i.e., from the past to the future. But in a bidirectional LSTM, the input sequence is processed in both directions simultaneously. This enables the model to capture not only the past information but also the future information of the input sequence.

3. Data

In this paper, the different datasets that have been taken are AAPL, BAM and UBER. AAPL and BAM are static datasets, while for UBER dynamic data has been fetched from Yahoo Finance [10]. Table 1 shows some of the data, which is the trading data of UBER of 5 days starting from 24th April,2023 to 28th April,2023, starting from when the market opens that is 09:30 am to 15:59 pm. All the data is of 5 days within an interval of 1m. The dataset contains eight columns, open, high, low, close, adjusted close, volume, date, and time. The first 1476 data is taken to be the training dataset and the last 369 data is taken as the test dataset.1983 to 12th December 2022. All the data is within an interval of 1 day. The first 7773 trading days data is used as training dataset and the last 1944 trading days data is used as test dataset. This BAM dataset is taken from Kaggle. Same thing has also been done in AAPL dataset data which is also taken from Kaggle [11]. These dataset does not contain the time column as they are static historical data taken day wise.

4. Implementation of CNN-LSTM

According to the problem statement of the paper, we are working on a time series data and the problem under consideration is a regression analysis problem. For this, a convolutional neural network is first defined followed by long short-term memory model. The input data is assumed to be a 3D tensor. This 3D tensor is defined using 3 parameters namely:

The CNN layers are used for feature extraction from the time series stock data. The LSTM layers function in order to collect and investigate the temporal trends in the data. The function of Time Distributes layer, can be explained as, it is used for the application of the same Convolutional operation to each time step independently. MaxPooling1D is applied after each convolutional layer to reduce the dimensionality of the output. After the CNN layers, the Bidirectional LSTM layers are used to capture the forward and backward temporal dependencies in the data. Dropout is applied to avoid over fitting, and for the regression analysis, a single output is produced by the final denser layer with linear activation function.

5. Implementation of Ensemble Model

To cater to the problem statement, a function is first created to implement a single instance of the model architecture. The ensemble model architecture framework comprises the CNN and LSTM layers. As described earlier, the CNN layers are used for feature extraction and the LSTM layers are used to collect and investigate the temporal trends in the stock data. After the data is passed through the above layers in order to produce a final single continuous value, a final dense layer is employed which uses a “tanh” linear activation function in this project then, a list of Num models’ instances of the ensemble is created.

Each instance is trained separately on the same training data, with same hyper parameters. After this, the fit method is employed for each instance for training, using the training data, in order to predict by calling the method for each instance, for predicting the results.

© 2023 IJSRET

Finally, predictions generated by each model instance are combined using the np mean function, which calculates the average of the predicted values across all the models.

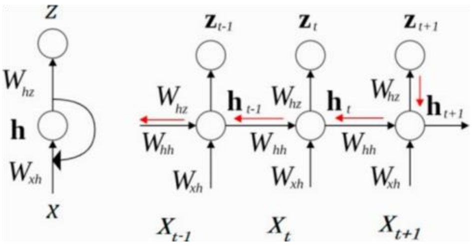


Fig.1 Representation of Ensemble Model

III. ALGORITHM

1. Data Collection and Pre Processing
- **Data Acquisition:** Gather a dataset scans, including both positive and negative cases.
- **Data Labelling:** Label the images as positive or negative based on expert diagnosis.
- **Data Pre Processing:** Resize images, normalize pixel values, and perform other necessary Pre Processing steps to prepare the data for model training.

2. Dataset Splitting
- **Train-Validation-Test Split:** Divide the dataset into training, validation, and test sets to train, tune, and evaluate the model, respectively.

3. Model Architecture
- **CNN Architecture:** Design a CNN architecture suitable for image classification. It might involve convolutional layers, pooling layers, and fully connected layers.
- **Output Layer:** Ensure the output layer matches the number of classes (positive and negative).

4. Model Training
- **Initialization:** Initialize the CNN model.
- **Loss Function:** Define an appropriate loss function (e.g., Cross-Entropy).
- **Optimizer:** Choose an optimizer (e.g., Adam, SGD).
- **Training Loop:** Train the CNN on the training dataset, adjusting model weights to minimize the loss.

5. Model Evaluation
- **Validation Set Evaluation:** Assess the model's performance on the validation set to tune hyper parameters and prevent over fitting.
- **Test Set Evaluation:** Evaluate the trained model's performance on the unseen test set to measure its generalization ability.

6. Model Interpretation and Diagnosis
- **Prediction:** Use the trained model to predict scans.
- **Thresholding:** Establish a threshold (based on probability or other measures) to determine positive/negative diagnosis.
- **Interpretation:** Interpret the model's predictions and assess its accuracy, sensitivity, specificity, etc.

7. Model Optimization and Refinement
- **Hyper parameter Tuning:** Fine - tune hyper parameters (learning rate, batch size, etc.) to improve model performance.
- **Data Augmentation:** Apply data augmentation techniques (e.g., rotation, flipping) to increase dataset diversity and prevent over fitting.

8. Validation and Further Assessment
- **Expert Validation:** Validate the model's findings with medical professionals to confirm the accuracy and reliability of the model's diagnosis.

9. Iterative Improvement
- **Iterate and Refine:** Iterate through the model development, evaluation, and validation steps to refine the model's performance and diagnostic accuracy.

Implementing a CNN-based diagnostic system involves careful data handling, model development, and rigorous validation to ensure its accuracy and reliability in clinical settings. Collaboration with medical experts and ethical considerations regarding patient data are also crucial throughout the process

IV. RELATED STUDIES

Currently, using conventional econometric approaches, it can be challenging to assess the financial market today because it is a dynamic and complicated system. A new approach to forecasting, utilising machine learning techniques, is required for high dimensional and noisy financial series data. The two main categories of stock price forecasting approaches are traditional analysis method and machine learning methods. For the analysis of complicated financial data, traditional econometric methods with parameters are inadequate. Neural networks have gained popularity as a research topic for stock forecasting in recent years. This is due to the fact that they are able to independently extract data features from an enormous amount of high-frequency raw data. White employed neural networks to forecast IBM shares in 1988. However, the outcomes weren't promising [2].

Zhang forecasted stocks in 2003 using neural networks and ARIMA which stands for Autoregressive Integrated Moving Average models. Although neural networks have clear

© 2023 IJSRET

advantages in nonlinear data forecasting, experimental findings indicated that their accuracy still needs to be increased [3]. A time series forecasting technique based on a fusion of the optimum partition algorithm (OPA) and a radial basis function (RBF) neural network was put out by Sun et al. in 2005 [4]. In order to predict four financial time series data, Adhikari et al. devised a method in 2014 that combines Random Walk (RW) and Artificial Neural Network (ANN). The findings indicated that there had been some improvement in forecasting accuracy [5]. In 2018, Zhang and his team suggested a network topology based on the LM-BP neural network for stock price forecasting. This was an improvement over the conventional BP neural network training algorithm's low precision and slow training speed [6]. Convolutional neural networks (CNNs) can be used to solve the problem of time series prediction, according to experimental findings from the same year by HU and his team. Because CNN is more frequently used for feature extraction and image recognition, it had relatively low forecasting accuracy [7]. Kamalov forecasted the stock prices of four significant US public firms for the year 2019 using MLP, CNN, and LSTM.

According to experimental findings, these three styles outperformed comparable studies that read the direction of price change [8]. In 2020, Xue and his team used the LSTM deep neural network to develop a short-term financial market forecasting model. The study argues that such an approach can explain the underlying dynamics and be tailored for real time stock price or stock price movement forecasting. The study further comprises an extensive evaluation of existing research on the prediction and estimation of stock price fluctuations, a discourse on the fundamental concepts, and an outline of the deep learning models employed in the analysis. Moreover, a comprehensive examination of the deep learning models' efficacy is presented, along with a contrast of their outcomes. The CNNLSTM model utilized in this research utilizes CNN for isolating essential characteristics from the temporal input data and LSTM for prognosticating the stock's closing price on the subsequent day. The dataset has been divided uniformly for both kinds of data, as training dataset and test dataset, 80% and 20% respectively. The paper proposes implementation of the CNN-LSTM model on both static and dynamic data. The static data has been taken for different stocks which is the historical data of these stocks of roughly past 5 to 10 years. The dynamic data has been fetched from Yahoo Finance for the related stocks over a period of 5 days within 1 minute interval.

The paper uses 1D CNN as individual units as well as in an ensemble learning model to compare between the accuracy of the learning methods. Along with the use different modelling techniques, we are also considering the datasets that are both static and dynamic.

In terms of forecasting accuracy, firstly talking about the static historical stock data, the maximum residual error is 0.615561, R2 score is 0.950553, mean absolute error is 0.05690 and mean squared error is 0.00576 without the ensemble learning whereas with ensemble learning, the R2 score is found to be 0.94176, the maximum residual error is reduced to 0.23614, mean absolute error is found to be 0.03176 and mean squared error is 0.00176. Talking about the dynamic dataset, the R2 score is 0.940492, maximum residual error is 0.010727, mean absolute error is 0.002011 and mean squared error is 7.5072e-06 without ensemble learning models whereas for ensemble learning, the R2 score is 0.923374, maximum residual error is 0.011855, mean absolute error is found to be 0.002036 and mean squared error is 7.8143e-06.

The outcomes demonstrate how well the CNN-LSTM model performs in case of static data with ensemble and without ensemble and dynamic data with ensemble and without ensemble. Figures 6–9 showcase the comparative analysis of the different methods and data. In terms of forecasting accuracy, in case of static data, the R2 Score, 0.950553, is 0.8% higher in comparison from with ensemble to that of without ensemble. In case of dynamic data, the R2 Score, 0.940492, is 1.71% higher in comparison from ensemble to that of without ensemble. Therefore, the CNN-LSTM model without the ensemble model works better in forecasting accuracy

V. TESTING

1. Unit Testing
Testing of an individual software component or module is termed as Unit Testing. It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. It may also require developing test driver modules or test harnesses.

2. Component Testing
Component Testing is mostly performed by developers after the completion of unit testing. Component Testing involves testing of multiple functionalities as a single code and its objective is to identify if any defect exists after connecting those multiple functionalities with each other.

3. Integration Testing
Testing of all integrated modules to verify the combined functionality after integration is termed as Integration Testing. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

4. System Testing
Under System Testing technique, the entire system is tested as per the requirements. It is a Black-box type Testing that is based on overall requirement specifications and covers all the combined parts of a system.

© 2023 IJSRET

5. Interface Testing
The objective of this Interface Testing is to validate the interface as per the business requirement. The expected interface of the application is mentioned in the detailed design document and interface mock-up screens. Checks if the application correctly connects to the server.

6. Compatibility Testing
Compatibility Testing checks whether the application is compatible with the specified software and hardware requirements and functions efficiently as expected.

7. Performance Testing
Performance testing is used to check for appropriate and efficient performance is shown by the system as per the requirements. The connection requirements are to be maintained to ensure efficient performance evaluation.

8. Usability Testing
Under Usability Testing, User friendliness check is done. The application flow is tested to know if a new user can understand the application easily or not, proper help is documented if a user gets stuck at any point. Basically, system navigation is checked in this testing.

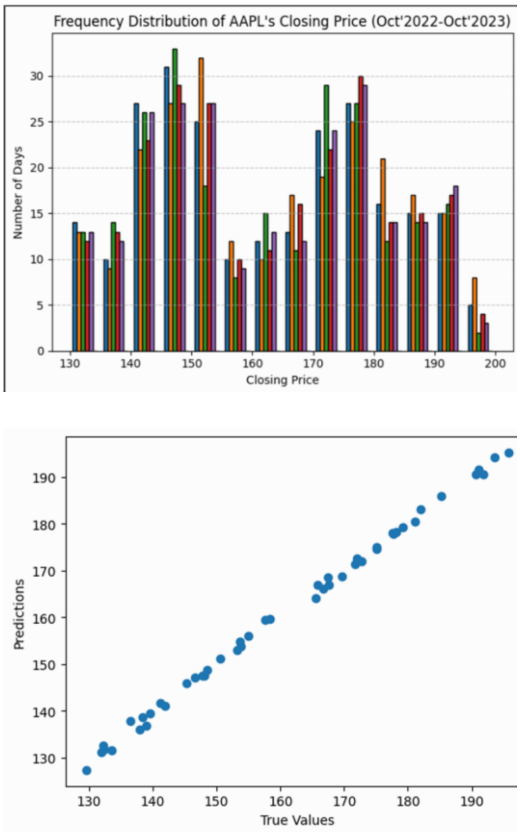


Fig.2 Prediction Graphs

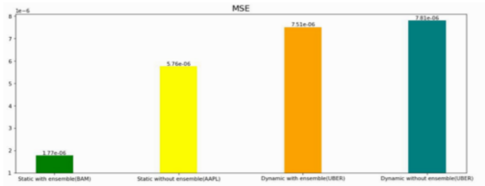
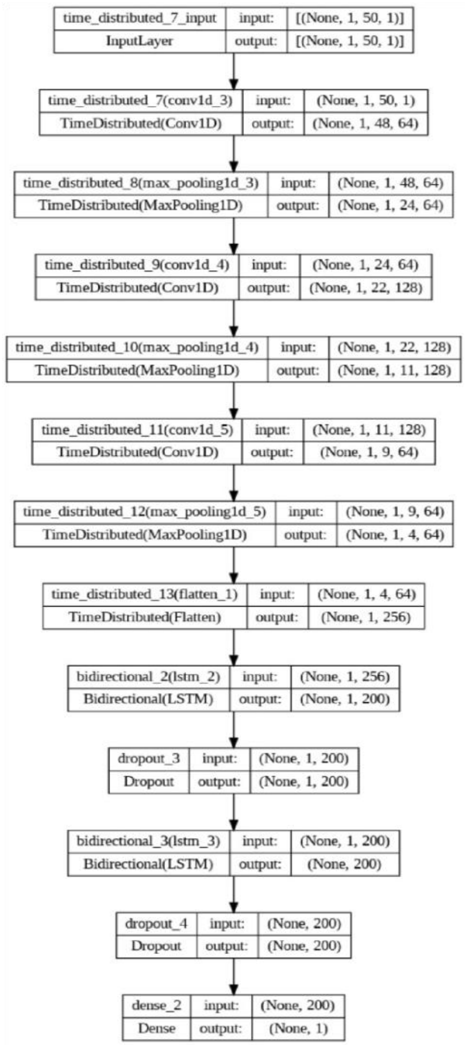


Fig. 9: Showcases the Mean Squared Error(MSE) on different methods and data.



VI. IMPORTANCE

Deep learning applications in particular continue to vary and grow quickly in number. Data augmentation strategies, for example, are data-centric approaches to model creation that might be a useful tool for overcoming obstacles in the artificial intelligence field. By creating additional and distinct instances for training datasets, data augmentation helps machine learning models perform better and produce better results. A machine

© 2023 IJSRET

learning model performs better and is more accurate if the dataset is large and sufficient.

Data collection and labelling may be time-consuming and expensive operations for machine learning models using data augmentation approaches, firms may transform datasets to save these operating expenses.

VII. MERITS AND DEMERITS OF OUR APPROACH

1. Merits

Automatic Feature Learning

CNNs excel at automatically learning hierarchical features from raw data. In the context diagnosis, this means the model can identify relevant patterns and structures images without the need for explicit feature engineering.

Spatial Hierarchical Representation

CNNs are particularly effective in capturing spatial relationships within data. In images, where the spatial arrangement of lesions and structures is crucial, CNNs can discern intricate patterns that might be challenging for traditional methods.

End-to-End Learning

CNNs allow end to-end learning, meaning the entire model can be trained as a unified entity. This simplifies the training process and enables the model to optimize its parameters across the entire architecture.

Robustness to Variations

CNNs are known for their ability to handle variations in scale, orientation, and position. This is advantageous when dealing with medical images that may have variability in terms of lesion size, location, and orientation.

2. Demerits

Large Dataset Requirement

CNNs often require a substantial amount of labelled data for effective training. Obtaining a large and diverse dataset of labelled can be challenging, especially with the need for expert annotations.

Computational Complexity

Training deep CNNs can be computationally intensive, requiring significant computational resources. This might pose challenges in resource constrained environments.

Interpretability

CNNs are often criticized for being "black box" models, meaning it can be challenging to interpret how the model arrives at a particular diagnosis. This lack of interpretability can be a concern in medical applications where understanding the decision-making process is crucial.

VIII. CONCLUSION

As seen, the paper involves using a combination of 1D CNN and Bidirectional LSTM model to predict stock market prices. The models have been trained on past stock market data and then used to predict future prices. It used a dynamic dataset to make the predictions more accurate and timelier.

The results obtained were promising, with the models achieving a good level of accuracy in predicting stock prices. However, there is still room for improvement in terms of refining the models and incorporating additional features that could further enhance accuracy.

IX. FUTURE WORK

Building on the current progress, there are several key areas that warrant further attention and development to enhance the effectiveness and usability of our gesture recognition system:

1. Enhanced Accuracy

Improving the accuracy remains a paramount objective. This involves continuous refinement of the underlying machine learning models, exploring advanced algorithms, and conducting rigorous testing and validation to minimize recognition errors.

2. Model Optimization

Model optimization is essential for ensuring the system's efficiency and real-time performance. By fine-tuning the algorithms, reducing computational overhead, and optimizing resource utilization, we can create a more responsive and seamless user experience.

3. Gesture Diversity

Expanding the repertoire of recognized gestures is vital for accommodating a broader range of sign languages, regional variations, and user preferences. The inclusion of additional gestures and signs will make the system more versatile and accessible.

4. User Feedback Integration

Continuous user feedback collection and analysis will guide the system's improvements. By closely collaborating with the deaf and hard of hearing community and other stakeholders, we can tailor the system to better align with their needs and preferences.

© 2023 IJSRET

5. Accessibility Features
Implementing accessibility features, such as real-time translation to text and speech, ensures that the system is inclusive and user-friendly. This step will further enhance the system's utility in diverse communication scenarios.

6. Scalability
Preparing the system for scalability is crucial, as it should be capable of handling increased user demand. Scalability will involve both software and hardware optimizations to support a growing user base.

7. Robustness and Reliability
Ensuring the robustness and reliability of the system in various environmental conditions and user contexts is imperative. Extensive testing and quality assurance procedures will contribute to its dependability.

8. Cross-Platform Compatibility
Enabling cross-platform compatibility will ensure that the system can be seamlessly integrated into various devices and applications, extending its reach and usability.

By addressing these future objectives, we aim to create a gesture recognition system that not only meets the immediate needs of the deaf and hard of hearing community but also remains adaptable and responsive to their evolving requirements. This ongoing development journey underscores our commitment to fostering more accessible and inclusive communication experiences.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of this minor project. This endeavour would not have been possible without the support, guidance, and encouragement from various individuals.

First and foremost, I extend my sincere thanks to Doctor Neha Agrawal & Ms. Meenu Garg, my project supervisors, for providing invaluable insights, constructive feedback, and continuous support throughout the project. Your expertise and mentorship have been instrumental in shaping the direction of this work.

I am grateful to the faculty members of the Information Technology Department for their academic guidance and encouragement. Their commitment to fostering a conducive learning environment has been crucial in my academic journey.

I would like to extend my appreciation to my peers and classmates for their collaborative spirit and willingness to share ideas. The exchange of thoughts and discussions have significantly enriched the project.

I must express my thanks to Maharaja Agrasen Institute of Technology for providing the necessary resources and infrastructure for the successful execution of the project.

Last but not least, I want to acknowledge the unwavering support of my family and friends. Their encouragement and understanding have been a constant source of motivation throughout this project.

In conclusion, I am deeply grateful to everyone who has played a role, big or small, in the completion of this minor project. Your support has been invaluable, and I am truly thankful for the collaborative efforts that made this project a reality.

REFERENCES

1. Halbert White, "Economic prediction using neural networks: the case of IBM daily stock returns," IEEE 1988 International Conference on Neural Networks, vol. 2, pp. 451-458, 1988.
2. G. Peter Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," Neuro computing, vol. 50, pp. 159–175, 2003.
3. Sun, Y.F., Liang, Y.C., Zhang, W.L. et al, "Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting," Neural Computing and Applications, vol. 14, pp. 36-44, 2005.
4. Adhikari, R., Agrawal, R.K., "A combination of artificial neural network and random walk models for financial time series forecasting," Neural Computing and Applications, vol. 24, pp. 1441-1449, 2014.
5. Zhang, L., Wang, F., Xu, B. et al., "Prediction of stock prices based on LMBP neural network and the estimation of over fitting point by RDCI," Neural Computing and Applications, vol. 30, pp. 425–1444, 2018.
6. Hu, Y., "Stock market timing model based on convolutional neural network–a case study of Shanghai composite index," Finance& Economy, vol. 4, pp. 71–74, 2018.
7. Alibasic, E., Fazo, B., i Petrovic, I., "A new approach to calculating electrical energy losses on power lines with a new improved three-mode method," Technik Venki, vol. 26, pp. 405–411, 2019.
8. Yan, X., Welham, W. & Chang, M., "Research on financial assets transaction prediction model based on LSTM neural network," Neural Computing and Applications, vol. 33, pp. 257-270, 2021. Ovidiu Popescu, "Technical Analysis of the Financial Markets," 10 03 2023. [Online]. Available:
9. https://www.morpher.com/blog/technical -analysis.
10. "Pipit, “Available: https://pypi.org/project/yfinan/

11. "Kaggle Stock Market Data," [Online]. https://www.kaggle.com/datasets