International Journal of Research in Engineering, Science and Management
Volume 8, Issue 11, November 2025
www.ijresm.com | E-ISSN: 2581-5792 | RESAIM Publishing (www.resaim.com)

Stock Growth Forecast Through Sentiment Analysis on Social Media and News with Statistical Factors

Ajeyaraj Upadhyaya[1*], Rahul Shinde[2], Taha Shaikh[3], Jyoti Ramteke[4]
[1,2,3]*Student, Department of Computer Science and Engineering, Sardar Patel Institute of Technology, Mumbai, India*
[4]*Professor, Department of Computer Science and Engineering, Sardar Patel Institute of Technology, Mumbai, India*

***Abstract*: The research proposes a comprehensive approach for predicting stock prices over a 30-business-day horizon with an accuracy of 98–99%. By incorporating multiple factors influencing stock prices, the methodology ensures reliable and well-rounded predictions. The base model employs a Univariate Time Series Prediction using a 1D Convolutional Neural Network (CNN-1D), which effectively captures seasonal patterns and general trends. While CNN-1D excels in identifying seasonality, it occasionally struggles with trends. To address this, the research integrates a Kalman Forecaster, which is highly reliable for trend prediction. The combined output of these models provides predictions encompassing both seasonality and trend dynamics, albeit with a slight reduction in accuracy compared to CNN-1D alone. The study further utilizes a Long Short-Term Memory (LSTM) network to predict market indices such as Sensex and Nifty, which, along with CNN-1D and Kalman predictions, inform a Random Forest regressor model trained over 320 days of a 365-day dataset. Predictions for the final 45 days are refined using mean error calculations over the first 15 days, which are then corrected using sentiment scores derived from news analysis powered by a large language model which in this case Gemini Flash Model 1.5. This integrated approach achieves a minimal error margin of 1–3%, successfully combining seasonality, trend, and external market factors for accurate stock price forecasting.**

***Keywords*: CNN-1D, Kalman Forecaster, LSTM, Time Series, News Analysis, LLM (Gemini Flash Model 1.5).**

1. Introduction

The system diagram illustrates the arrangement of models that contribute to the final output. The models implemented are described here

CNN-1D: For Univariate Time Series Forecasting on Stock closing prices

Kalman: For Univariate Time Series Forecasting on Stock Forecaster closing prices (for General Trend Forecasting)

LSTM : For Market Index Prediction using a Univariate Time Series Forecasting manner

Random: Use output of LSTM and (CNN-1D and Forest Kalman Forecaster Combination) to predict Regressor actual Stock Prices

LLM: Gemini Flash 1.5 LLM is used to generate scores out of news, this will be used for correction of Random Forest regressor model values.

The architecture of the solution proposed is as follows:

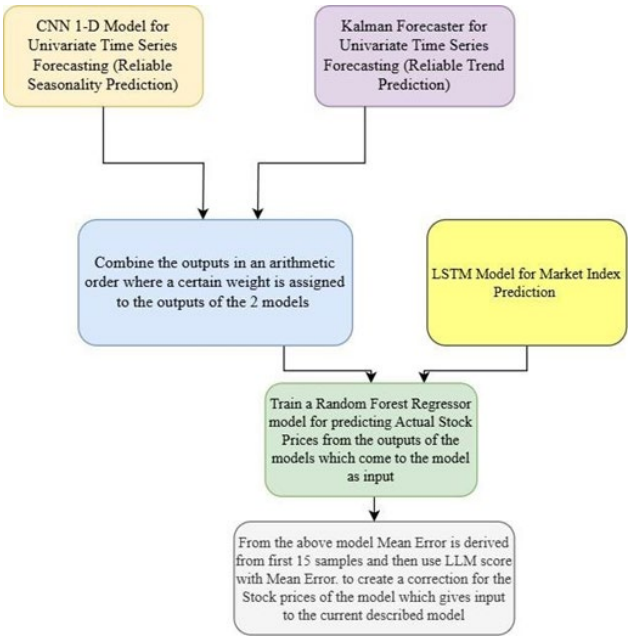

Fig. 1. Architecture

A. CNN-1D (Convolutional Neural Network)

1) Date Pre-Processing

Data pre-processing is a critical step in preparing time series data for modelling, ensuring that the input is suitable for the learning algorithms. In this implementation, the process begins with loading data where we retrieve historical stock price data using the yfinance library, specifically focusing on the closing prices within a specified date range.

The Raw data which we have loaded is then transformed into a supervised learning format where we creates lagged features (previous time steps) and corresponding target values for prediction.

This transformation allows the model to utilize past observations to forecast future values.

To stabilize the Time series and make it more amenable for

*Corresponding author: ajeyaraj.upadhyaya@spit.ac.in

modelling, the difference function computes the differences between consecutive observations, effectively removing trends and seasonality.

After the described step of differencing is applied, the values are scaled to fit between the values of -1 and 1. This scaling is done using the Minmax Scaler functionality which various libraries provide.

B. Working of the CNN-1D Model

The training data from the data pre-processing steps as described above are fed into the CNN-1D Model.

CNN-1D model is basically for learning from 1-D sequences like Univariate Time Series. The CNN-1D Model is designed as we see to process sequential data. A CNN-1D model is created using convolutional filters across a single dimension. This type of an architecture is particularly effective for time series due to its ability to capture local patterns and relationships within sequences.

The model begins with a convolutional layer that applies multiple filters to extract features from the input sequences. Each filter learns to recognize specific patterns over time, such as trends or seasonal effects.

The flattened output is then passed through one or more dense (fully connected) layers that learn complex representations of the features extracted by the convolutional layers.

Finally, a single neuron in the output layer predicts the next value in the sequence based on learned features.

1) Program Implementation Particulars

In the Data Pre-Processing steps we had described a difference operation to remove seasonality and trends and another operation called Scaling. After the results from CNN-1D Model are out we do perform inverse scaling and difference operations so we retort back to the normal condition.

Also, in the Data Pre-Processing operation we create training and testing datasets. For forecasting we use the last unit of training dataset to predict the first value of stock prediction. This prediction is incorporated in the input to the model for predicting the second value. This is a recursive implementation where the output of the model is incorporated to make the next prediction.

2) Results of the CNN-1D Model

Fig. 2. HDFC bank stock price prediction

The MAPE value for the above forecast is: 0.90%.

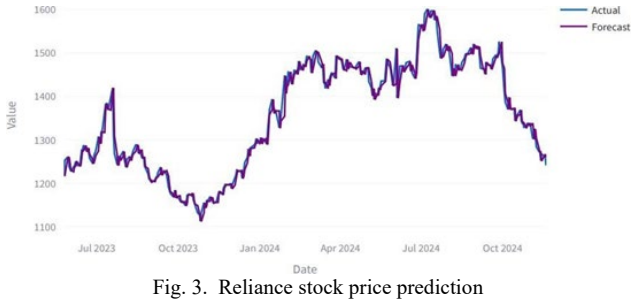

Fig. 3. Reliance stock price prediction

The MAPE value for the above forecast is 0.95%.

The CNN-1D Model makes a prediction over 365 Business Days.

C. Kalman Forecaster

1) Data Pre-Processing

The Data Pre-Processing in Kalman Forecaster is minimal here we just split the data into Training and Testing Datasets Here the last 365 days of the date range are predicted.

2) Working of the Kalman Forecaster Model

The Kalman Forecaster operates through a recursive algorithm that estimates the state of a system over time, balancing predictions and observations.

It consists of two main phases: Prediction and Update. Prediction Phase: The filter uses the previous state estimate and a state transition model to predict the current state and its uncertainty (covariance). This prediction incorporates the expected dynamics of the system, which may include random fluctuations.

Update Phase: Once a new measurement is obtained, the Kalman filter computes the difference between the predicted state and the actual measurement (innovation).

The innovation is weighted by the Kalman gain, which adjusts how much influence the new measurement has on updating the predicted state.

The updated state estimate combines the previous estimate and the weighted innovation, refining it based on new information.

The filter continuously iterates through these phases, allowing it to adapt to changes in the system dynamics while minimizing estimation errors.

It is particularly effective for systems with noise and uncertainty, making it widely used in applications like navigation and time series forecasting.

The Kalman filter's recursive nature means it does not require storing all past measurements, making it efficient for real-time applications.

3) Intuition Behind Kalman Forecaster

The CNN-1D Model is used which captures both trend and seasonality, but in some cases, it may not capture the trends in particular. Also, by only using the CNN-1D Model we cannot consider other features which we may want to consider.

4) Results of the Kalman Forecaster Model

The Model as we will see bellow is good for predicting the general linear trend.

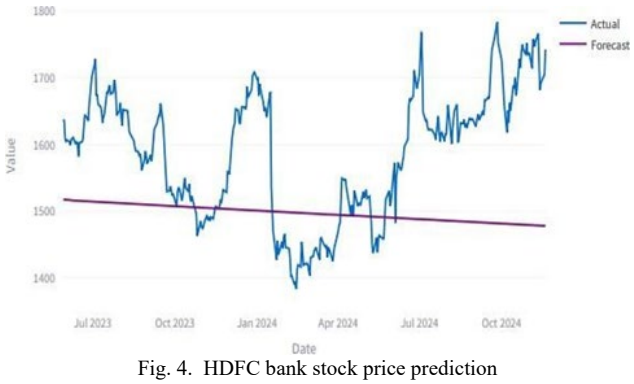

Fig. 4. HDFC bank stock price prediction

The general trend as we see in the above case is linear.

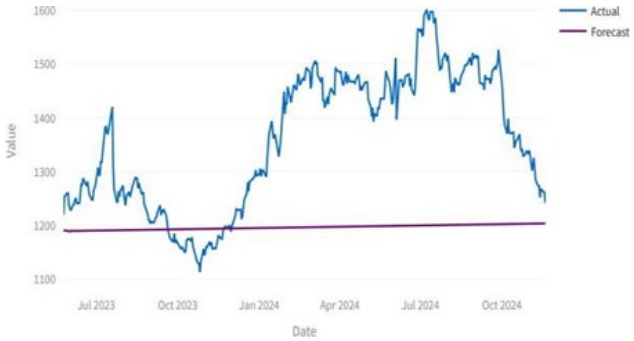

Fig. 5. RELIANCE stock price prediction

The general trend as we see in the above case is linear.

The Kalman Forecaster forecasts over 365 Business Days.

D. CNN-1D and Kalman Forecaster

Forecasts from the CNN-1D model are weighted at $\alpha=0.3$, and those from the Kalman Forecaster at $\beta=0.7$, creating a blended forecast that captures seasonality and trends.

While this combined approach yields lower accuracy than the CNN-1D alone, it facilitates the inclusion of additional features in the time series forecasting process.

These features are the scores which the Gemini Flash Model 1.5 will give after News Analysis and the Market Index prices which are predicted.

The Combination results in the following results as seen below.

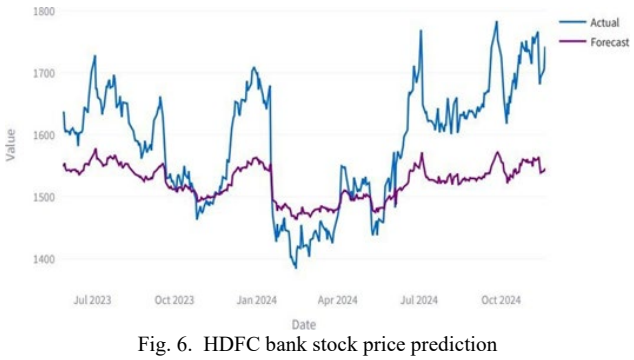

Fig. 6. HDFC bank stock price prediction

The MAPE value for the above forecast is 6.78%.

Fig. 7. RELIANCE stock price prediction

The MAPE value for the above forecast is 10%.

The Combination of CNN-1D and Kalman Forecaster also forecasts over 365 Business Days.

E. LSTM for Market Index Prediction

Market Index is a very powerful metric which can be leveraged for Stock Price Prediction.

Also as compared to other metrics like Volume. This metric is easily predictable using the various Deep Learning algorithms.

1) Data Pre-Processing

Data pre-processing is a critical step in preparing time series data for modelling, ensuring that the input is suitable for the learning algorithms. In this implementation, the process begins with loading data where we retrieve historical stock price data using the finance library, specifically focusing on the closing prices within a specified date range.

The Raw data which we have loaded is then transformed into a supervised learning format where we creates lagged features (previous time steps) and corresponding target values for prediction.

This transformation allows the model to utilize past observations to forecast future values.

To stabilize the Time series and make it more amenable for modelling, the difference function computes the differences between consecutive observations, effectively removing trends and seasonality.

After the described step of differencing is applied the values are scaled to fit between the values of -1 and 1. This scaling is done using the Minmax Scaler functionality which various libraries provide.

2) LSTM Model Basic Information

Long Short-Term Memory (LSTM) networks are a specialized type of recurrent neural network (RNN) designed to effectively learn from sequential data and address the vanishing gradient problem.

LSTMs address the limitations of traditional RNNs, which struggle with long-term dependencies due to issues like vanishing and exploding gradients.

This makes LSTMs particularly suitable for tasks where context from earlier time steps is crucial for making accurate predictions.

3) Architecture

LSTM has many components as described below.

1. *Memory Cell*: This is the core of the LSTM, capable of retaining information for long periods.

2. *Gates*: LSTMs use threegates to control information:
   - *Forget Gate*: Determines what information to discard from the cell state.
   - *Input Gate*: Decides what new information to add to the cell state.
   - *Output Gate*: Controls what information from the cell state is output as the hidden state.

These gates use sigmoid activation functions to produce values between 0 and 1, allowing the model to selectively retain or discard information.

4) Working

1. *Input Sequence Processing*: At each time step, the LSTM receives input data along with the previous hidden state and cell state.
2. *Information Flow*:
   - The forget gate evaluates which parts of the previous cell state to keep or discard.
   - The input gate updates the cell state with new candidate values.
   - The output gate generates the hidden state for the next time step based on the updated cell state.
3. *Long-Term Dependencies*: By maintaining a cell state that can be modified over time, LSTMs effectively capture long-term dependencies in sequential data, making them ideal for tasks like time series forecasting.
4. *Training*: The model is trained using backpropagation through time (BPTT), allowing it to learn patterns in data sequences efficiently.

5) Program Implementation Particulars

In the Data Pre-Processing steps we had described a difference operation to remove seasonality and trends and another operation called Scaling. After the results from CNN-1D Model are out we do perform inverse scaling and difference operations so we retort back to the normal condition.

Also, in the Data Pre-Processing operation we create training and testing datasets. For forecasting we use the last unit of training dataset to predict the first value of stock prediction. This prediction is incorporated in the input to the model for predicting the second value. This is a recursive implementation where the output of the model is incorporated to make the next prediction.

Fig. 8. Results of the LSTM market index forecaster SENSEX market index predictor

MAPE of the above forecast is 0.54%.

Fig. 9. NIFTY market index predictor

MAPE of the above forecast is 0.54%.

The LSTM Model as described here forecasts over 365 Business Days.

F. Random Forest Regressor

The Random Forest Regressor is used to combine the outputs of the Combination of CNN-1D and Kalman Forecaster and the LSTM Marker Index Model

All the above concerned models predict over a period of 365 days. We take Actual and Predicted values of the First 320 days for the Combination of CNN-1D and Kalman.

Here Combination of CNN-1D and Kalman has Actual Stock Prices as Actual values and Predicted Stock Prices as Predicted values

Also, the LSTM Model will predict Market Indices over 365 Business Days. We will take the predicted market indexes from this model.

Now we will train a Random Forest Regressor Model where the model is trained to predict Actual Stock prices based on the Predicted Stock Prices and Market Indices. All 3 entities are derived from the above described models. The Model is trained on the first 320 days and its corresponding data out of the 365 days. After this we predict for the last 45 days.

1) Random Forest Regressor Background and Working

A Random Forest Regressor is a machine learning model that uses multiple decision trees to make predictions for regression tasks. A decision tree works by splitting data into smaller groups based on the values of input features, aiming to minimize errors like Mean Squared Error (MSE). However, a single tree often overfits the data, meaning it performs well on the training set but poorly on unseen data. Random Forest overcomes this by creating an ensemble of decision trees, where each tree is trained on a random subset of the data (using a method called bootstrapping) and only considers a random subset of features at each split. This randomness ensures that the trees are diverse, which helps in reducing overfitting and improving generalization.

During prediction, the Random Forest combines the outputs of all the individual trees to make a final prediction. For regression tasks, it takes the average of all tree predictions. This process makes the model more robust to noise and variations in the data. Random Forest handles large datasets and high-dimensional data efficiently and is less sensitive to missing data compared to many other models. However, because it builds multiple trees, it can require more computational resources and may not perform ell if the trees are too similar to each other.

Overall, Random Forest is a powerful and versatile model widely used in machine learning due to its simplicity and strong predictive performance.

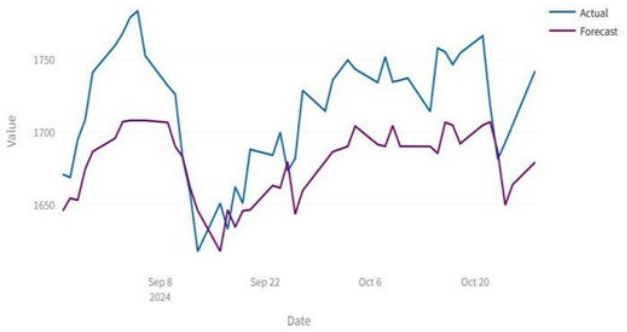

Fig. 10. Results of the Random Forest regressor stock forecast HDFC bank stock price prediction

The MAPE of the above forecast is about 2.26%.

Fig. 11. RELIANCE stock price prediction

The MAPE of the above forecast is about 2.33%.

The Above Forecasts of the Random Forest Regressor are over a period of 45 days.

G. News Analysis Integration

In this Research Paper we get news from Google RSS these are various articles and their headlines. After this we create a custom prompt for the LLM Model Google Gemini Flash.

This result for this prompt would contain Textual Information on various aspects like Sentiment Summarization, Contextualization of Market Trends, Consideration of Historical Data etc.

Here very importantly we get a score which is used for making corrections.

From the Random Forest Model, the first 15 days of the 45 are compared with the actual values and a Mean Error is calculated. The Score by LLM is operated upon and multiplied to the Mean Error.

This Final result is added to the remaining 30 predicted values. This operation reduces the error by about 1.5%. This output is the final output of the research paper, the error is now around 1.5 – 2 %.

Fig. 12. Results of the news analysis integration HDFC bank stock price forecast

MAPE of the Above Forecast is 1.87%.

Fig. 13. RELIANCE stock price forecast

MAPE of the above forecast is 1.20%.

The Above Forecasts are the Final Results and this is a prediction for 30 days.

2. Conclusion

By using various models like CNN-1D, Kalman forecaster LSTM, Random Forest and LLM (Google Gemini Flash 1.5) we have created a stock price predictor which on an average gives an error of 1.5%.

These models have helped integrate 3 crucial factors to predict stock prices one being previous stock prices, the second one being stock market indices which give us an idea of the larger market and its state and the third one being news and the scores provided on its analysis

References

[1] K. Srilakshmi and Ch. Sai Sruthi, "Prediction of TCS stock prices using deep learning models," in *Proc. 2021 7th Int. Conf. Adv. Comput. Commun. Syst.* (ICACCS), Coimbatore, India, Mar. 19–20, 2021, vol. 1, pp. 1448–1455.

[2] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. 2017 Int. Conf. Adv. Comput., Commun. Informat.* (ICACCI), Udupi, India, Sept. 13–16, 2017, pp. 1643–1647.

[3] Y. Yan, X. Nie, M. Wang, and Y. Chen, "LSTM-based stock price prediction model using news sentiments," in *Proc. Int. Symp. Econ., Social Develop. Trade* (ISESDT), 2023.

[4] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in *Proc. 2016 IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci.* (ICIS), Okayama, Japan, 2016, pp. 1–6.