International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IBM Stock Price Prediction with Sentiment Analysis

***Sravya Vallabhaneni*[a]**

[a] School of Computer Science and Technology, VIT-AP University, Amaravati, Andhra Pradesh, 522241, India

ABSTRACT:

In the year 2020 stock markets in India have crashed due to the effects of covid. In the same year Cristiano Ronaldo moved Coca-Cola bottles away and held up a water bottle, ensuring a stock market dip and a loss of $4 billion for the company. These examples demonstrate the relationship between news and stock market prices. In this study we aim to analyze the relationship between news articles and stock prices using AI. This will help investors to make more informed decisions. In this paper, we leverage Artificial Intelligence (AI) to analyze the relationship between news articles and stock prices. By utilizing sentiment analysis, we extract valuable insights from financial news and assess their impact on stock movements. This approach enables investors to make more informed and data-driven investment decisions, thereby decreasing risks associated with market instability caused by sudden news events.

By 2030, we can expect a surge in the usage of AI in the field of stock markets. This is going to help people make informed decision making. So it is important to achieve high accuracy in this field. In this paper, attempt to decode the relationship between global trends and the stock market using sentiment analysis alongside time series analysis.

Keywords: Stock market Price; Time Series Analysis; Sentiment Analysis; RNN; LSTM; GRU; FastRNN.

1. INTRODUCTION

The stock market has always been uncertain and unpredictable because of ever-changing global trends, macroeconomic conditions, and unpredictable events such as geopolitical tensions, and even social media influence. Traditional stock market prediction methods heavily rely on statistical models, time series forecasting, and machine learning algorithms. However, these conventional approaches are not capable of analysing sudden market shifts and the complex interplay of qualitative factors such as news sentiment, public perception. They are also influenced by pandemics and disasters as well as global trends.

This is where sentiment analysis plays a crucial role. By implementing sentiment analysis of financial news articles with time series forecasting, we can enhance the accuracy of stock price predictions while accounting for global market trends. This helps us understand the impact of new in Stock market price prediction. We can access Stock market data through different API's but collecting related news articles from different websites and handling this data is a bigger task. To implement sentiment analysis effectively, we will implement different recurrent neural network (RNN) architectures, such as Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and FastRNN. These deep learning models are well-suited for handling sequential data such as time series and natural language. These algorithms are known for their superior performance in both time series forecasting and sentiment classification tasks.

2. Literature Review

Stock price prediction that has been performed prior to this uses machine learning algorithms such as ARIMA, SARIMA, Naive Bayers considering each word to be interdependent. Over the years there are many analytics that are performed on stock prices of companies such as Apple, Amazon and Tesla. But stock prices for IBM still remain underexplored. Most of the research papers available on IBM data do not explore models beyond LSTM and Simple RNN. This paper also focuses on reducing the training time using GRU and FastRNN

3. Data collection and Pre Processing

The dataset used for paper is taken from Kaggle and includes around 70,000 values. It contains financial news headlines from Reddit with stock market data from the Dow Jones Industrial Average (DJIA) to help forecast future stock prices. The goal is to predict the adjusted closing price (scaled by 105) using historical stock trends and news sentiment. This dataset contains from 2008 to 2015, the dataset provides both time series data and articles that are collected from Reddit. It also contains high, low, closing and trading values of IBM stock market

Before feeding this data into a model, we need to clean it up so it's structured and usable. Computers do not understand raw text hence it is important to convert text data into numerical data such as vectors. In order to make text data machines readable we have to perform things like stopword removal, punctuation removal, and special characters removal. Then, we tokenize the text (breaking it into smaller pieces), apply stemming or lemmatization to break the word into base word, and handle any missing values. Data preprocessing is the first and foremost step that we need to perform in order to get high accuracy.

To convert this text into something a model can actually understand, we use vectorization. This technique helps convert cleaned text into mathematical representation. For our model we have used TF-IDF (Term Frequency - Inverse Document Frequency) vectorization. It used two different functions Term Frequency and Inverse Document Frequency to attain most meaningful terms in text to improve the model's ability to predict stock price changes based on market sentiment.

4. Evalutionary Metrics

Root Mean Squared Error (RMSE) and R-squared (R^2) are popular metrics for evaluating prediction models. Lower RMSE values indicate higher accuracy, whereas R^2 values range from 0 to 1 indicate the model's ability to explain data variance. In is important to have lower RMSE and higher R^2 value for a model to be determined as reliable. In our case we are going to consider both of these metrics as primary evaluation metrics. Although accuracy scores of certain models are higher for the provided test data it is crucial for models to perform well on real-world data. Hence we are opting these metrics for evaluation.

5. Methodology

5.1 End to End Modeling

To make a model compatible with real word data substantial to make your models work end to end. The end to end model takes raw data and gives output. For most of the models we have pre-process input data before passing it into the model for prediction. Making models end to end is even more crucial for language models. As discussed above a machine can not understand natural language. Hence we have to perform some tasks before passing it on to the mode. One of the easiest way to achieve this is by adding layers in the model itself that perform preprocessing tasks such as tokenization and vectorization.We can make use of Functional API instead of Sequential API to help us reduce creating the same layers in every model.

5.1.1 Functional API:

Functional API adds more flexibility to models by allowing layers to have multiple input and output layers. It permits different models to have shared layers and enable residual connections in models. It improves the flexibility of models when compared to sequential API where layers are stacks on top of each other.

5.2 RNN

Recurrent neural networks are known for their superior performance in time series analysis and natural language processing. It takes the output of previous layers along with new inputs to predict the next output. With this architecture it has the ability to understand and decode the trends that are present in time series data. Traditional RNNs can efficiently capture the time dependencies in data but may encounter vanishing gradients and exploding gradient problems.

RNN fails to capture the long-term dependencies present in the dataset. While training the deep RNN through backpropagation through time, the gradient of the loss function is computed with respect to earlier time steps. As the time series goes on, it cannot remember the effect of the data because of small values of weights leading to exponentially diminishing gradients. This prevents the network from learning long-range dependencies. This is referred to as the vanishing gradient problem. On the other hand, if the weights of the model have large eigenvalues, the gradient can grow exponentially during backpropagation, causing instability in training. This leads to sudden and erratic updates in the model, leading to the exploding gradient problem.

Although RNN has given us satisfactory results in this dataset, we can overcome vanishing and exploding gradient issues by using models such as FastRNN, LSTM and GRU.

5.3 FastRNN

Recurrent Neural Networks (RNNs) have provided results that are satisfactory in modeling time-series and text data. But RNNs often struggle with vanishing and exploding gradient problems, leading to untrainable models and lower performance for longer periods of training time. To overcome these challenges, FastRNN includes residual connections along with two additional parameters. This improves stability and convergence while reducing overfitting in models. By modifying the update rule to incorporate a residual connection, FastRNN ensures more stable gradient propagation, effectively lessening the common setbacks of traditional RNNs.

It is based on the given formula:

$h_t = (1-\beta)\cdot\sigma(Wx_t+Uh_{t-1}+ b)+\beta\cdot h_{t-1}$

FastRNN is an improved version of traditional RNNs designed to remove vanishing and exploding gradient problems while maintaining performance. It incorporates a gate-like update mechanism which allows stable gradient flow over long periods of time. One of its key advantages over basic RNN is its ability to generalize data. In deep learning, generalization error measures how well a trained model performs on new data ,by reducing the overfitting of the data. Basic RNNs may lead to overfitting on the data while FastRNNs achieve better results because of their control over error propagation. This helps get more predictable data in long term forecasting.

FastRNN has shown noticeable results even in sentiment analysis. It readily captures long-term dependencies in news data, allowing it to recognize minute patterns in market trends which could not be accessed through regular RNN. FastRNN provides a way faster convergence, reducing training time without compromising accurate results. Compared to other architectures, it achieves results comparable to more complex models like LSTMs and GRUs, but with fewer parameters and lesser computational costs.

5.4 LSTM

Long Short-Term Memory is a type of RNN architecture widely used across in building models using deep learning. It uses gates such as input gate, output gate and forget gate to address the vanishing gradient problem, which inturn allows us to capture the long term dependencies in larger data. This architecture excels in computing long-term dependencies, making them apt for sequence prediction tasks.

LSTM are introduced because RNNs couldn't handle sequential data due to exploding and vanishing gradients. It captures information from the past and carries it forward, making it suitable for NLP. By combining these methods, LSTMs can maintain long-term dependencies significantly more than regular RNNs. To be able to maintain the influence of features from previous steps during training, the memory cell does not allow gradients to rapidly decay. LSTMs potential to learn complex patterns from data is greatly increased by the use of gates with weight parameters of its own,compared to a simple RNN which only uses a single weight matrix.

Because of its complex structure, LSTMs demand higher processing power despite their benefits. Training time is extended by the numerous matrix multiplications and non-linear transformations happening at each time step. And also, if the model is not appropriately changed, its complexity may make it overfit the model. LSTMs provide good accuracy but it is computationally very heavy. This is why GRU's are introduced to reduce training time and give almost similar accuracy.

5.5 GRU

Gated Recurrent Unit (GRU) is an improvised version of RNN which can capture long term dependencies while adjacently taking care of vanishing gradients .Generally referred to as the lightweight version of LSTM. It simplifies LSTM by combining the forget and input gates into a single update gate. This single update gate determines how much of the past information should be carried next into the model. while the reset gate determines how much of the previous hidden state is carried forward and removed before processing the new input. Unlike LSTMs, which always maintain a separate memory unit, GRUs immediately update into hidden state, making them computationally less expensive while performing well in sequential tasks. GRUs are particularly useful in handling sequential data applications, where they perform like LSTMs but have fewer parameters, allowing for faster training and inference.

GRUs work while learning which parts of the past information to hold on to and which to discard, making them suited for long-term data. The update gate decides whether to keep the old hidden state or update it with new information, ensuring efficient learning without more memory usage. The reset gate allows the model to selectively forget past information when needed, which is preferably useful while handling short-term dependencies. Since GRUs have a simpler architecture than LSTMs, they require lesser computations per unit time, making them a lot faster while still being able to achieve comparable performance in tasks such as sentiment analysis, and time series forecasting. In fact we have seen significant improvements while using GRU.

In our case GRU has provided almost near RMSE and r-squared error scores to that of LSTM. We can imply that GRU has the ability to provide greater accuracy while being computationally light compared to the rest.

6. Results and Discussion

The GRU model outperforms other recurrent models in stock price prediction using sentiment analysis, achieving the lowest RMSE and the highest R^2 score. This indicates that GRU predictions are the most accurate. Although LSTM has given us significant results for this dataset, it is important to notice the faster training time and comparable accuracy given by GRU and FastRNN. Using GRU makes the model computationally light while maintaining almost the same RMSE score.

FastRNN, which does not contain any gated mechanism, also gave desired results when compared to more advanced models such as LSTM and GRU. Furthermore, FasterRNN has a lower RMSE value compared to traditional RNN while having a faster computation. From Figure 2 we can observe that the training loss for LSTM has continuously decreased with each epoch. This indicates that the model is continuously learning, and increasing the number of epochs for LSTM may give us better results.

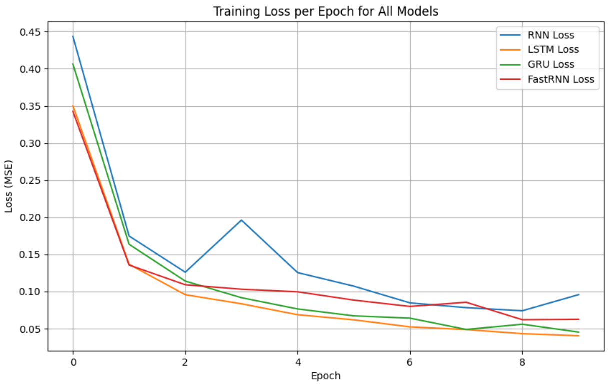

Fig 1 - Training loss per epoch

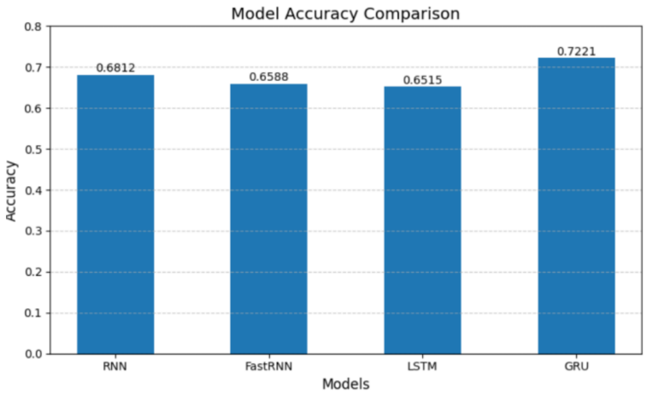

Fig 2 - Model Accuracy Comparison

Table 1. Evaluation of Model

Model	RMSE	R-squared error	Accuracy
RNN	0.1770	0.3851	0.6812
FastRNN	0.1184	0.3800	0.6588
LSTM	0.1074	0.4896	0.6515
GRU	0.0771	0.7374	0.7221

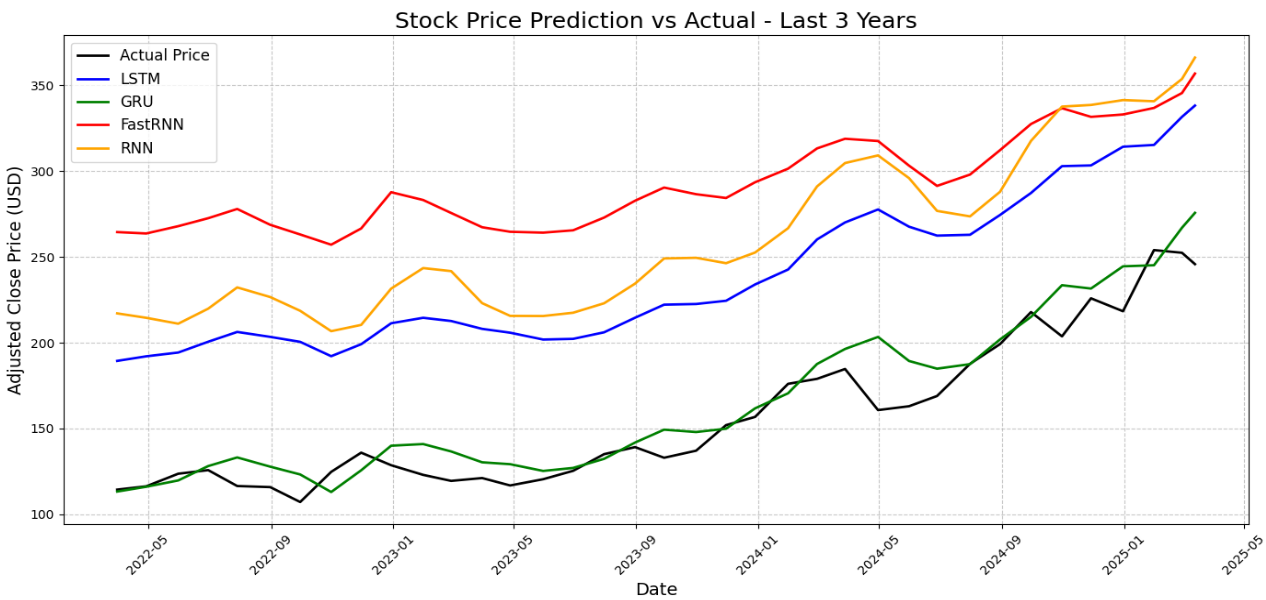

Fig 3 - Prediction of models on real-time data collected through Alpha Vantage API from 2022 – 2025

7. Optimization techniques

Fine Tuning a model requires several optimization techniques and callbacks. For this all the models we have chosen Adam optimizer which gave better results than using SGD. It includes optimization techniques such as Regularization, Dropout layers and callbacks such as early stopping to improve accuracy.

7.1 Dropout Layers

Dropout layer is a technique helping the architecture prevent overfitting of the deep learning models. By adding a dropout layer we can deactivate a part of neurons at random during training. dropout layers force the network to learn more useful features to predict the results than relying on all the neurons.which in turn leads to decreasing the chances of overfitting and improving the models prediction accuracy on real-world data.

7.2 L1 and L2 Regularization

L1 and L2 regularization techniques add a penalty to the loss function to handle the model's complexity. L1 regularization, also known as Lasso regularization, adds the absolute sum of the existing weights to the loss function as a penalty which would help the model to be accurate in feature selection. While L2 regularization, also known as Rigid regularization, adds the square of the magnitude of the weights which helps reduce overfitting of the complex models.

7.3 Early Stopping Callback

Early stopping callback is another technique used to put a stop to overfitting by keeping track of the validation loss or accuracy during training. If the model's performance doesn't improve for a particular set of epochs the training of the model is stopped,so the n=model does not learn patterns which go against the existing ones and cause overfitting. This technique also allows us to restore the weights for best accuracy after pausing the training.

8. Conclusion and Future Work

This paper focuses on different deep learning models for sentiment analysis and time series forecasting in the context of stock price predictions. It highlights the effectiveness of RNN, LSTM, GRU, and FastRNN. We can further enhance the accuracy by using Transformer-based models like BERT and BiLSTM. BiLSTM refines traditional LSTM by capturing context and dependencies present in both forward and backward directions. This improves its ability to understand relations between words in both directions and helps discover the full scope of financial news. Transformers in text and computer vision are widely used across the world to retain useful information in text. BERT is a transformer-based model developed by Google that helps in further refining this by leveraging deep bidirectional attention, making it more effective in understanding complex financial text. Additionally, transformer-based prediction models utilize self-attention mechanisms to capture long-term dependencies that are present in data. Combining these approaches can significantly enhance sentiment-based stock price prediction by improving both text representation and sequential pattern learning.

As artificial intelligence continues its advancement, its role in daily life, including its role in predicting the prices of stocks, also increases. There is vast financial data available, and taking into account the potential of artificial intelligence, there may be advanced models developed to analyze these vast amounts of structured and unstructured data with greater accuracy. Transformer-based architectures, such as BERT and Time-Series, are going to play a major role in the future of finance. They are set to play a crucial role in financial prediction by ways like enhanced text-based feature extraction and improved predictive performance. While also, the assimilation of reinforcement learning and generative AI models will surely allow adaptive trading strategies that respond to market fluctuations accordingly. As AI-driven models become more polished, they will provide deeper insights into investors' sentiment and economic trends, all while considering market anomalies. This paves the way for advancements towards reliable and efficient stock price prediction methodologies.

REFERENCES

Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., and Anastasiu, D. C. 2019. Stock price prediction using news sentiment analysis. IEEE Publications. link=https://davidanastasiu.net/pdf/papers/2019-MohanMSVA-BDS-stock.pdf.

Pagolu, V. S., Challa, K. N. R., Panda, G., and Majhi, B. "Sentiment Analysis of Twitter Data for Predicting Stock Market Movements."

Kusupati, A., Singh, M., Bhatia, K., Kumar, A., Jain, P., and Varma, M. FastGRNN: A Fast, Accurate, Stable and Tiny Kilobyte Sized Gated Recurrent Neural Network. Microsoft Research India, Indian Institute of Technology Delhi, University of California Berkeley.
link=https://www.prateekjain.org/publications/all_papers/KusupatiSBKJV18.pdf

Wu, S., Liu, Y., Zou, Z., and Weng, T.-H. "S_I_LSTM: Stock Price Prediction Based on Multiple Data Sources and Sentiment Analysis."
DOI= https://www.tandfonline.com/doi/full/10.1080/09540091.2021.1940101.

Batra, R., and Daudpota, S. M. 2018. "Integrating StockTwits with Sentiment Analysis for Better Prediction of Stock Price Movement." 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE.

Khedr, A. E., Salama, S. E., and Yaseen, N. 2017. "Predicting Stock Market Behavior Using Data Mining Technique and News Sentiment Analysis." *I.J. Intelligent Systems and Applications*, 7, 22-30. link= 10.5815/ijisa.2017.07.03.

Nguyen, T. H., Shirai, K., and Velcin, J. "Sentiment Analysis on Social Media for Stock Movement Prediction." *School of Information Science, Japan Advanced Institute of Science and Technology*, and *University of Lyon (ERIC, Lyon 2), France.*

Nguyen, T. H., and Shirai, K. "Topic Modeling-Based Sentiment Analysis on Social Media for Stock Market Prediction." *School of Information Science, Japan Advanced Institute of Science and Technology*, Japan.

Peivandizadeh, A., Hatami, S., Nakhjavani, A., Khoshsima, L., Chalak Qazani, M. R., Haleem, M., and Alizadehsani, R. (Member, IEEE). 2024"Stock Market Prediction With Transductive Long Short-Term Memory and Social Media Sentiment Analysis." *IEEE Access*.link=https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10528270.

Joshi, K., Bharathi, H. N., and Rao, J. "STOCK TREND PREDICTION USING NEWS SENTIMENT ANALYSIS".