# **Stock Price Prediction Using Recurrent Neural Networks With RNN, LSTM and GRU Models**

***Abstract****:  
Stock market forecasting is one of the significant functions of finance management that has considerable advantages for investors, traders, and other institutions. This study focuses on forecasting the 'Close' price of Alphabet Inc. (Google) stock using three Recurrent Neural Network (RNN) architectures: There are Vanilla RNN, Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) networks. By performing data preprocessing, feature engineering and model training on the data set of historical stock data from January 2010 to December 2023 derived from the Kaggle dataset Google Stock Price Dataset, the performance of all these models is assessed. To measuring the accuracy of prediction, MAE, RMSE and MAPE have been employed. These evaluations show that LSTM networks perform relatively better in modelling intricate temporal dependence than those of the GRU model, while the Vanilla RNN model is established as the baseline. The results of this comparative analysis would be useful for understanding the applicability of the various RNN designs to actual financial forecasting tasks.*

**Objectives:**

The primary objectives of this study are:

1. To evaluate and compare the performance of Vanilla RNN, GRU and LSTM models in predicting stock ‘Close’ prices
2. To identify the strengths and limitations of each RNN variant in the context of financial time -series forecasting.
3. To provide insights into the suitability of these models for real world stock market predictions.

**Introduction**

Stock performance prediction is essential in investment analysis especially on prominent stocks such as Alphabet Inc. (Google). Simple univariate models such as ARIMA or ETS are effective with linear, short-memory data while financial data are non-linear and have long memory. This is the reason that more complex architectures such as Recurrent Neural Networks (RNNs), including Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) networks are more appropriate. This paper focuses on these deep learning architectures for the predictions of Google’s closing stock prices to present the high competitiveness of stock price predictions as well as to analyse different approaches to stock price forecasting. The key competitors include:

**Support Vector Machines (SVM):**

SVMs are learning models used in the category of supervised learning being employed in classification and regression tasks such as financial forecasting. The basic independent variables and dependent variable relationships can be non-linear in nature due to the kernel functions and is highly resistant to overfitting in many dimensions. The limitations are non-adaptive to big data, high computational time required for model training, and overview of having a Received Time that is characteristic of time series data.

Random Forests: A learning procedure in which a number of decision trees are-individual models are developed and combined to form a new model that provides better and more stable results. The advantages are Less sensitivity to the number of parameters and high accuracy, better performance in training data overfitting, better performance with a large feature space, and feature importance scores. The drawbacks are small analogies to temporal dependencies and sequences, making them still less suitable for time series prediction than RNNs.

**Convolutional Neural Networks (CNNs):** Originally developed for analysing spatial data, these CNNs were extended to the time series forecasting problem by treating temporal data as spatial features. Advantages: In this sense it is good as making local extractions, it is efficient with computations with Convolution layers. Limitations: Due to the absence of inherent temporal modelling capability, they are less capable of capturing long range dependencies as compared to RNNs.

**Recurrent Neural Networks (RNNs)**

Vanilla RNN, GRU, and LSTM are specific types of RNNs which are particularly effective at modelling temporal dependencies while at the same time, do not involve such heavy computation as to make them impractical for most applications. Unlike earlier statistical models which restrict themselves to modelling disturbances with linearity, RNNs have the advantage of unearthing non-linear relationship which makes them excellent models for analysing financial time series data. Such RNN variants are at the centre of this study to establish their performance difference in predicting Google’s stocks price and thus laying down the competitors.

**Method Description:**

The work aims at predicting Google’s stock prices using Recurrent Neural Networks (RNNs), appropriate for sequential data such as time series. Google's stock prices by leveraging Recurrent Neural Networks (RNNs), which are particularly suited for sequential data like time series. Three variants of RNNs are explored:

•Vanilla RNN

• Gated Recurrent Unit (GRU)

• Long Short-Term Memory (LSTM)

**1.Method implementation of Google stock dataset**

Google stock price data set used in this project is obtained from Kaggle from the contributor Rahul Sah Google Stock Price Dataset. Kaggle also is a platform which hosts many datasets which are good for data science and machine learning activities. This dataset is curated to provide comprehensive historical stock price information for Google (GOOGL), facilitating various predictive modelling tasks. The dataset comprises five key features, each representing a different aspect of Google's stock performance daily. For the achievement of predictive models & measured by theoretical reasoning and logical per se is not quite adequate, but rather it also requires highly scrupulous approach in terms of technical detail within a stringent methodological framework. This section focuses on the details of the strategies employed when deploying the proposed RNN models namely Vanilla RNN, GRU, and LSTM for the ‘Close’ price of Google’s stocks. The implementation includes the specifications of development environment, deployment of numerous model architectures, protocols of training and recordability measures.

* Open: The opening price of Google's stock on a given trading day. The importance reflects the stock's initial trading value at the start of the trading session, influenced by overnight news, market sentiment, and other external factors.
* High: The highest price at which Google's stock traded during the trading day which Indicates the peak value reached by the stock, providing insights into intraday volatility and investor enthusiasm.
* Low: The lowest price at which Google's stock traded during the trading day which represents the minimum value the stock reached, highlighting periods of potential bearish sentiment or market corrections.
* Close: The share price of Google computed at the close of the business day. It is used as a measure of evaluating daily performance and it is applied in technical analysis in the preparation of trading.
* Volume: The cumulative share volume processed in the day; an important indicator of market presence and volumes of trading activities of the specific equity as high volumes are often associated with price fluctuations or special events.

**Dataset Splitting:**

* **Training Set**: Constitutes 80% of the dataset and is used to train the models. This large proportion allows the models to learn as much as possible about the data’s underlying patterns.
* **Validation Set**: Accounts for 20% of the training data. This set is crucial for tuning the hyperparameters and for implementing regularization strategies to mitigate the risk of overfitting.
* **Testing Set**: This is a distinct subset comprising 20 consecutive days of data, separated from the training and validation sets. It is used to evaluate the model's performance on new, unseen data, providing an indication of how the model might perform in real-world scenarios

**Data preprocessing:**  
Data preprocessing occupies a significant place in the preparation of raw dataset for processing with machine learning models including the neural networks such as RNN. The following steps outline the comprehensive preprocessing pipeline employed in this project:

* Handling Missing Values: To increase the quality of data by making corrections to or filling gaps in entries in the dataset. The approach has been employed data examination to check for missing values in features across all the six features.
* Handling Strategies for Missing Data: To manage missing values effectively, we either omit rows with minimal missing entries to prevent skewing the data or employ imputation techniques for larger gaps. Common statistical methods like mean, median, or mode, as well as more advanced approaches such as K-Nearest Neighbour imputation, are utilized. Addressing these missing values is crucial as they can lead to biased or inaccurate predictions if ignored, thereby ensuring the model receives comprehensive and accurate input data.

**Feature Selection:**

Feature selection is a critical step in our stock price prediction project, as it equips the RNN models with a robust dataset containing key stock performance indicators: Open, High, Low, Close, and Volume. This comprehensive dataset allows the models to discern complex patterns and temporal dynamics in stock market data, enhancing prediction accuracy. Effective feature selection not only boosts model accuracy but also sets a foundation for future enhancements. Additionally, the design includes measures to prevent potential issues like feature duplication and overfitting through thoughtful design considerations and the implementation of regularization techniques. This strategic approach to feature selection and utilization is fundamental for the success of predictive models in stock price forecasting.

**Performance Metrics:**

* **Mean Squared Error (MSE)**: This metric calculates the average of the squares of the differences between the predicted and actual values. It emphasizes larger errors, making it particularly useful in scenarios where large deviations are undesirable.
* **Root Mean Squared Error (RMSE)**: RMSE is the square root of MSE and is expressed in the same units as the target variable, making it more interpretable. It provides a clear measure of how accurately the model predicts the target variable, with lower values indicating better performance.
* **Mean Absolute Error (MAE)**: MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It’s less sensitive to outliers than MSE, providing a more robust overview of model accuracy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **MODEL** | **MSE** | **RMSE** | **MAE** |
| 0 | VANILLARNN | 67.875318 | 8.238648 | 6.223219 |
| 1 | GRU | 403.307587 | 20.082519 | 18.592156 |
| 2 | LSTM | 423.603593 | 20.581632 | 16.669814 |

**Visualizations**:

This provides crucial insights into each RNN variant's performance in predicting stock 'Close' prices, illustrating their accuracy and highlighting areas for improvement.

* **Actual vs. Predicted Prices for Vanilla RNN**: The Vanilla RNN's predictions generally track the trend of stock prices but falter during high volatility, showing lag and inaccuracies. This behaviour underlines the model's limitations in capturing complex temporal dynamics due to its simpler structure.

A graph with a line and a line

Description automatically generated

A graph showing a line of loss

Description automatically generated with medium confidence

**Actual vs. Predicted Prices for GRU**: The GRU model aligns more closely with actual prices than the Vanilla RNN. Its gating mechanisms help it adapt to changes in the market more effectively, showcasing its ability to handle nuanced patterns and market dynamics.

A graph with a line and a line

Description automatically generated

A graph of a graph

Description automatically generated

**Actual vs. Predicted Prices for LSTM**: The LSTM displays excellent performance, closely mirroring actual 'Close' prices with minimal lag. Its complex architecture allows it to capture both short-term fluctuations and long-term trends, outperforming the other models.

A graph with a line and a red line

Description automatically generated

A graph of a line

Description automatically generated

**References**

1. Hochreiter, S., & Schmidhuber, J. (1997). **Long Short-Term Memory.** Neural Computation, 9(8), 1735–1780.
2. Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). **On the Properties of Neural Machine Translation: Encoder-Decoder Approaches.** arXiv preprint arXiv:1409.1259.
3. Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). **Learning to Forget: Continual Prediction with LSTM.** Neural Computation, 12(10), 2451–2471.
4. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). **LSTM: A Search Space Odyssey.** IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2222–2232.
5. Graves, A. (2013). **Speech Recognition with Deep Recurrent Neural Networks.** In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6645–6649.
6. Brownlee, J. (2017). **Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python.** Machine Learning Mastery.