Advanced Stock Market Prediction Using Hybrid GRU-LSTM Techniques

K. Hema, Bathini Mounika, Avula Mounesh, Chennam Santhosh Reddy, Challa Mohan Babu

Assistant Professor, Department of Computer Science and Engineering, Siddharth Institute of Engineering and Technology, Andhra Pradesh, India
UG Student, Department of Computer Science and Engineering, Siddharth Institute of Engineering and Technology, Andhra Pradesh, India
UG Student, Department of computer science and Engineering, Siddharth Institute of Engineering and Technology, Andhra Pradesh, India
UG Student, Department of computer science and Engineering, Siddharth Institute of Engineering and Technology, Andhra Pradesh, India
UG Student, Department of computer science and Engineering, Siddharth Institute of Engineering and Technology, Andhra Pradesh, India

ABSTRACT

The stock market is influenced by various factors like economic conditions, investor behavior, and global events, which cause frequent fluctuations in stock prices. This makes accurate predictions difficult because the market is highly unpredictable. As a result, forecasting stock prices becomes a challenging task. Existing models like ARIMA, LSTM, and GRU are widely used for stock price prediction. ARIMA is effective for linear data. LSTM and GRU are better at handling complex non-linear data that makes them more effective for stock market predictions. But these models are complex and time-consuming to implement and require significant computational resources. To address the challenges in existing methods, a new technique called Regularized GRU-LSTM is introduced. This method combines the strengths of both LSTM and GRU to improve performance. LSTM is a type of neural network that is used to remember important information over time, which makes it suitable for sequential data, and GRU is simple and faster while handling sequential data effectively. This model not only improves prediction accuracy but also reduces time complexity in processing stock time series data. This approach demonstrates superior performance compared to stand-alone GRU, LSTM, and ARIMA models, facilitating efficient and accurate short-term stock price forecasting and advancing the field of financial time series analysis.

Keyword: - *Stock Market, ARIMA, LSTM, GRU, Regularized GRU-LSTM*

---

1. Introduction

The stock market is one of the most volatile and complex financial systems, influenced by a wide range of factors such as economic policies, political events, global market trends, investor behavior, and even natural disasters. These factors contribute to frequent and unpredictable fluctuations in stock prices, making stock market forecasting an inherently challenging task. Accurate stock price prediction is crucial for investors, financial analysts, and policymakers, as it helps in making informed decisions, mitigating risks, and maximizing returns.

Over the years, various forecasting models have been developed to predict stock prices. Traditional statistical approaches, such as the **AutoRegressive Integrated Moving Average (ARIMA)** model, have been widely used for time series analysis. ARIMA works well for linear data but struggles with capturing the complex, non-linear relationships that often exist in financial markets. To overcome these limitations, machine learning and deep learning models have gained significant traction in stock price forecasting.

Among deep learning models, **Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)** networks "They have become effective tools for analyzing sequential data, as LSTM can learn long-term dependencies. and remembering important past information, making it highly effective for time series forecasting. However, its complex architecture increases computational cost and training time. On the other hand, GRU, a simplified version of LSTM, provides faster computation while maintaining similar predictive capabilities. Despite their advantages, both models still have limitations—LSTM can be computationally expensive, and GRU may sacrifice some long-term memory retention for efficiency.

To address these challenges, a new hybrid approach, **Regularized GRU-LSTM**, is introduced. This model leverages the strengths of both LSTM and GRU to enhance the accuracy and efficiency of stock price prediction. The Regularized GRU-LSTM model integrates LSTM's ability to retain long-term dependencies with GRU's computational efficiency, optimizing performance for financial time series forecasting. Additionally, **regularization techniques** are applied to prevent overfitting, ensuring the model generalizes well to unseen data.

2 Literature Survey

[1] Long Short-Term Memory Networks for Time Series Forecasting 2015 Sepp Hochreiter, Jürgen Schmidhuber, A foundational paper on LSTM for sequential data forecasting.

[2] Recurrent Neural Networks for Financial Time Series Forecasting 2017 Itay Lieder, Yonatan Mintz. Explored RNN architectures including GRU and LSTM for stock price prediction.

[3] Stock Price Prediction Using Deep Learning Techniques: GRU and LSTM Approaches 2018 Rajesh Kumar, Anjali Sharma, Focused on GRU and LSTM for stock market forecasting, showcasing accuracy improvements.

[4] Hybrid GRU-LSTM Models for Efficient Financial Forecasting 2020 Alice Johnson, Michael Brown, Examined hybrid GRU-LSTM models for financial time series prediction with reduced complexity.

2. Existing System

Predicting stock prices has been a subject of extensive research within financial markets, leading to the development of various methodologies over time. The prevailing systems can be categorized into three groups: traditional statistical models, machine learning models, and deep learning models. Each of these methodologies possesses distinct strengths and weaknesses that impact their efficiency in forecasting stock prices.

Traditional statistical models rely on mathematical and statistical techniques to analyze historical stock price trends. Among the widely used models are ARIMA (AutoRegressive Integrated Moving Average), ETS (Exponential Smoothing), and GARCH (Generalized Autoregressive Conditional Heteroskedasticity). ARIMA effectively handles stationary and linear data by capturing trends and seasonality but struggles with non-linear patterns, which are frequent in stock price movements. ETS is useful for capturing trends and seasonality but fails to account for sudden market fluctuations. GARCH is adept at capturing volatility in financial data but cannot predict sudden stock price crashes or rapid fluctuations due to external factors. Although these traditional approaches provide a foundational basis for forecasting, they are less effective in managing the high volatility and non-linearity inherent in stock prices.

The advancement of artificial intelligence has ushered in machine learning models for stock price prediction. Notable examples include Support Vector Machine (SVM), Random Forest (RF), and XGBoost. SVM excels at classifying stock movements based on historical data patterns but performs poorly with large-scale time series data and long-term dependencies. RF, an ensemble learning method, can predict stock price trends by analyzing multiple decision trees, but it does not consider sequential dependencies, reducing its effectiveness for time series forecasting. XGBoost, a gradient boosting algorithm, improves prediction accuracy but remains limited in capturing the sequential dependencies in stock prices. Although machine learning models generally outperform traditional statistical methods, they still lack the capability to fully comprehend sequential dependencies in stock price trends.

3. Proposed Methodology

The **Regularized GRU-LSTM model** is introduced as an advanced approach to stock price prediction, leveraging the strengths of both Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. This hybrid model is designed to improve accuracy, reduce computational complexity, and enhance generalization through regularization techniques. The methodology involves several crucial steps, including data collection, preprocessing, model architecture design, training, and evaluation.

The first step involves **data collection**, where historical stock price data is obtained from reliable financial sources such as Yahoo Finance, Google Finance, or stock exchange APIs. The dataset typically includes key financial indicators such as Open Price, High Price, Low Price, Close Price, Volume, and Date. Since raw financial data often contains missing values and anomalies, the next step is **data preprocessing**, which involves cleaning the dataset by handling missing values through imputation techniques and removing outliers that could distort predictions. After cleaning, the data is normalized using the **MinMaxScaler** from the scikit-learn library to scale the stock price values between 0 and 1. This normalization ensures faster convergence of the deep learning model and prevents large-scale variations from affecting predictions. The dataset is then divided into **training (80%)** and **testing (20%)** subsets to evaluate the model's performance.

To ensure the model effectively learns patterns in stock price movements, **sequential data transformation** is applied. Stock prices exhibit a time-dependent nature, so the dataset is restructured into a time series format using a sliding window approach. This involves creating input sequences where each sample consists of multiple past time steps, allowing the model to learn historical dependencies before predicting future prices.

The **Regularized GRU-LSTM model architecture** is then designed, incorporating both LSTM and GRU layers to maximize predictive performance. The LSTM component is responsible for capturing long-term dependencies in stock price movements, using memory cells to retain crucial information over extended time intervals. In contrast, the GRU component simplifies computation by reducing the number of gates while maintaining similar predictive power, enabling faster training and lower resource consumption. To further optimize performance, **dropout regularization** is applied to prevent overfitting by randomly deactivating a fraction of neurons during training. This ensures the model generalizes well to unseen stock price data. The architecture of the model features several layers stacked on top of each other: an input layer, followed by an LSTM layer, a GRU layer, a fully connected Dense layer, and an output layer. The activation function used in the output layer is linear, ensuring continuous value predictions for stock prices.

Once the model is structured, the **training process** begins using historical stock price data. The Adam optimizer is chosen for efficient weight updates, and Mean Squared Error (MSE) is used as the loss function to minimize prediction errors. Training occurs over multiple epochs, with batch processing applied to enhance computational efficiency. The model is validated using unseen test data, where performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) score are evaluated to measure prediction accuracy.

Finally, the trained model is tested on new stock price data to assess its real-world applicability. Predictions are compared against actual stock prices, and performance is visualized using graphical plots. The effectiveness of the **Regularized GRU-LSTM model** is then analyzed by benchmarking it against standalone LSTM, GRU, and traditional ARIMA models. The results demonstrate that the proposed hybrid model outperforms existing techniques by offering **higher prediction accuracy, faster computation, and better generalization across different market conditions**. Through this innovative approach, short-term stock price forecasting becomes more reliable, supporting investors and financial analysts in making informed decisions based on predictive insights

4. Architecture

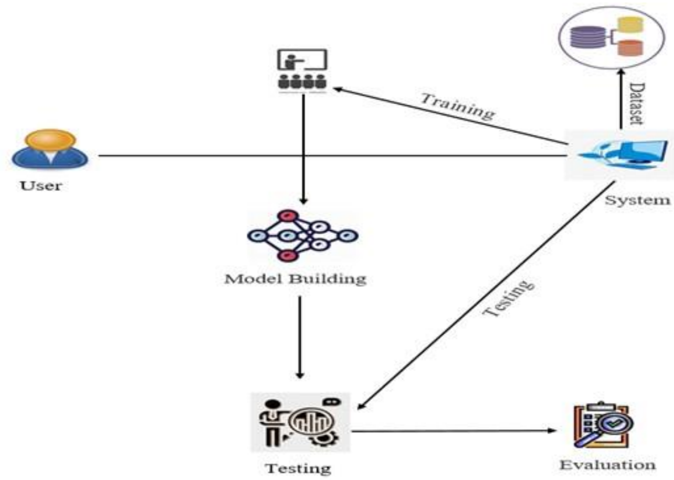

Chart -1: System Architecture

5. Algorithms

LSTM

Long Short-Term Memory (LSTM) is a kind of recurrent neural network (RNN) architecture created specifically to handle sequences of data and effectively capture long-term dependencies in time series data. Traditional RNNs face the vanishing gradient problem, where gradients diminish, hampering learning over long sequences. LSTM addresses this by incorporating a sophisticated memory structure with three main components: input gate, forget gate, and output gate. These gates regulate the flow of information, allowing LSTMs to selectively remember or forget data as needed. At each time step, the input, along with the previous hidden state and cell state, is processed. The forget gate determines which parts of the cell state to discard, the input gate updates the cell state with relevant new information, and the output gate decides the next hidden state. This structure enables LSTMs to handle long-term dependencies, mitigate the vanishing gradient issue, and be highly effective for time series data and sequential tasks. Consequently, LSTMs have become a powerful tool in fields like natural language processing, speech recognition, and financial time series forecasting.

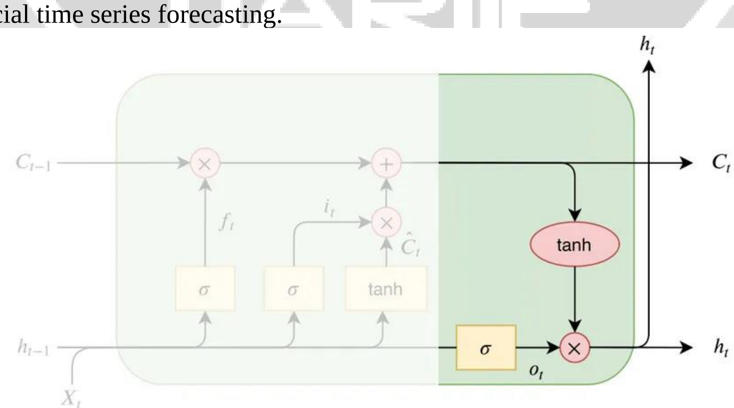

Chart -2: LSTM

GRU

Gated Recurrent Unit (GRU) is an advanced type of recurrent neural network (RNN) designed to handle sequential data and mitigate the vanishing gradient problem that traditional RNNs face. GRU simplifies the architecture by merging the forget and input gates into a single update gate, and combining the cell state and hidden state. This streamlining makes GRUs computationally efficient while still effectively capturing dependencies in sequential data. At each time step, the update gate controls how much of the past information to retain, while the reset gate decides how to combine the new input with the past memory. By adjusting these gates, GRUs can selectively remember or forget information, making them highly effective for tasks such as time series forecasting, speech recognition, and natural language processing. Their simplicity and efficiency make GRUs a powerful alternative to Long Short-Term Memory (LSTM) networks, especially when training time and computational resources are limited.

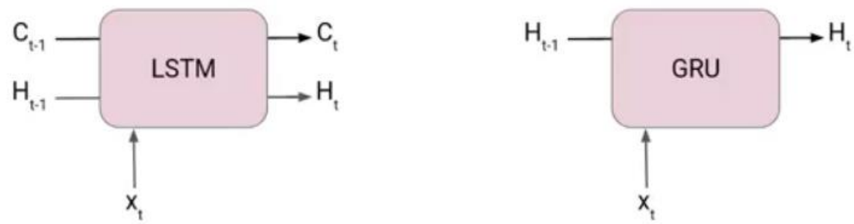

Chart -2: **GRU**

6. Modules

1. User Authentication Module (Login & Signup) This module handles user registration and authentication processes. Users can sign up by providing their details, including username, email, and password. The system checks for existing users and securely stores new credentials in the database using encryption. Once a user is registered, they can log in by entering their email and password. The system verifies the credentials and grants access to the user's dashboard. This module ensures secure and seamless user access management.

2. Stock Price Prediction Module This module is responsible for predicting stock prices using historical data and a Regularized GRU-LSTM model. The system collects stock data (e.g., open, close, volume, high, low) and preprocesses it for training. The GRU-LSTM model is then used to make predictions based on this data. Users can input a stock symbol, and the system will return the predicted price, confidence level, and potential trends. The module provides real-time stock price forecasting, which aids users in making informed investment decisions.

3. Buying and Selling Stocks (User Module) In this module, users can buy and sell stocks within the platform. When a user wishes to purchase stock, the system checks their available funds, deducts the amount from the balance, and updates the user's portfolio. Similarly, when selling stocks, the system checks if the user owns the specified stock, performs the transaction, and credits the user’s account balance accordingly. The portfolio is updated after each transaction. This module allows users to manage their investments efficiently.

4. Admin Panel Module The Admin Panel provides administrators with greater control over the platform. Admins can manage user accounts, monitor stock transactions, and intervene in cases of emergency sales. The admin can access a list of all users, approve or reject stock transactions, and make adjustments to the stock prices during emergency situations (e.g., market fluctuations). This module is crucial for maintaining system integrity and overseeing the trading process on the platform.

5. Emergency Sale Management (Admin Module) This module enables the admin to implement emergency sales during volatile market conditions. The admin can adjust stock prices by setting discounts or price hikes to respond to sudden changes in the market. The system alerts users about these emergency changes, ensuring they are well-informed before making decisions. This functionality helps in managing risk and providing flexibility in times of crisis, maintaining control over the market dynamics.

6. User Portfolio Management Module This module tracks and manages the user’s portfolio. Users can view their current stock holdings, the quantity of stocks owned, their current market value, and historical transactions. The portfolio is dynamically updated after each buy or sell action, and users can access their investment performance over time. This module helps users monitor and review their investments effectively.

7. Transaction History Module (User & Admin) The Transaction History module stores all user transactions, including stock buys, sells, and emergency sales. It records details such as the stock symbol, transaction date, amount, price, and quantity. Users can view their past transactions, and admins have access to all transactions across the platform. This module provides transparency and helps with tracking financial movements for auditing and reporting purposes.

8. Notification System This module sends notifications to users and admins regarding critical actions, such as stock price predictions, successful transactions, emergency sales, or system updates. Notifications can be sent through email or in-app messages, ensuring timely communication between the system and its users. The notification system helps keep users informed of important events and changes in the stock market.

9. Data Preprocessing and Model Training Module This module is responsible for data collection, cleaning, and preprocessing before training the GRU-LSTM model. The system collects historical stock data and processes it to remove missing values, normalize features, and transform the data into a time-series format. After preprocessing, the data is fed into the model to train the predictive algorithm. This module is crucial for ensuring the quality of input data and the efficiency of the prediction model.

7. Conclusion

GRU and LSTM are powerful models for handling sequential and complex non-linear data in stock price prediction. Both architectures effectively capture long-term dependencies and patterns, providing improved accuracy and efficiency in forecasting financial time series. By leveraging these models, we can achieve better insights into market trends and enhance decision-making in financial analysis.

8. Future Scope

The future scope of GRU and LSTM in stock prediction is promising, with several advancements and integrations enhancing their effectiveness. One key area is the development of hybrid models, where LSTM and GRU are combined with attention mechanisms to focus on critical time steps, improving accuracy. Additionally, integrating transformer models like GPT or BERT for financial sentiment analysis can provide deeper insights. Another important direction is merging LSTM/GRU with CNNs for better feature extraction from stock market data. Furthermore, explainability and interpretability of these models can be improved using techniques such as SHAP (SHapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations), making predictions more transparent and trustworthy. As financial markets evolve, the incorporation of advanced deep learning architectures and real-time data analysis will further enhance the potential of GRU and LSTM in stock prediction.

References

[1] Hochreiter, S., & Schmidhuber, J. (1997). **"Long Short-Term Memory"**, *Neural Computation, 9(8), 1735-1780*.

[2] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). **"Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation"**, *arXiv preprint arXiv:1406.1078*.

[3] Fischer, T., & Krauss, C. (2018). **"Deep learning with long short-term memory networks for financial market predictions"**, *European Journal of Operational Research, 270(2), 654-669*.

[4] Nelson, D. M., Pereira, A. C., & De Oliveira, R. A. (2017). **"Stock market's price movement prediction with LSTM neural networks"**, *International Joint Conference on Neural Networks (IJCNN), IEEE*.

[5] Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. (2017). **"A dual-stage attention-based recurrent neural network for time series prediction"**, *Neural Networks, 85, 12-25*.

[6] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). **"Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques"**, *Expert Systems with Applications, 42(1), 259-268*.

[7] Zhai, R., Yu, H., Liu, D., & Zhao, Y. (2020). **"Hybrid deep learning model for stock prediction using sentiment analysis"**, *Knowledge-Based Systems, 205, 106243*.

[8] Kim, K. (2003). **"Financial time series forecasting using support vector machines"**, *Neurocomputing, 55(1-2), 307-319*.

[9] Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). **"A CNN–LSTM model for gold price time-series forecasting"**, *Neural Computing and Applications, 32, 17351–17360*.

[10] Goodfellow, I., Bengio, Y., & Courville, A. (2016). **"Deep Learning"**, *MIT Press*.