Project report : Enhancing Stock Price Prediction Using Social Media Sentiment Analysis and Advanced Machine Learning Models

Hamza Boukhriss, 3543099
Nicolas Bieszczad, 3542726
Diego Dalsasso, 3549305
Federico Ugolini, 3542882
Giannantonio Sanrocco, 3532542
Project Group 6

Abstract

This report explores the improvement of stock price predictions using sentiment analysis on social networks, an area where artificial intelligence tools are profoundly transforming the financial world. While recent research has demonstrated the value of incorporating social media sentiment into stock market prediction models, there is no widely adopted open-source solution that reliably predicts stock prices using this data.

Our study aims to bridge the gap in understanding how sentiment expressed on Twitter influences US stock market prices. We leverage RoBERTa, a powerful pre-trained language model, to perform fine-grained and contextualized sentiment analysis. To enhance stock market prediction, we embed the extracted sentiment features into our models, integrating them with other potential predictive variables.

For stock price forecasting, we employ time-series models such as Gated Recurrent Units (GRUs), which effectively capture long-term dependencies in sequential data. Additionally, we explore the incorporation of technical indicators to identify the optimal feature set for each model. To benchmark performance, we compare GRUs with an ensemble learning approach using Random Forest. Furthermore, we introduce a Generative Adversarial Network (GAN) architecture with GRUs to investigate whether a more complex model can outperform both traditional time-series and ensemble-based approaches.

The performance of the models will be evaluated using standard metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics will allow us to assess the accuracy and reliability of our predictions across different models.

For data sources, we will use two datasets from Kaggle: one containing tweets related to the 25 most followed stocks on Yahoo Finance and another providing the corresponding stock market data. These datasets will be combined to analyze the impact of online sentiment on stock price movements, enabling us to explore the relationship between sentiment dynamics and market fluctuations.

To evaluate the effectiveness of our innovative approach—combining GRU models with sentiment analysis—we will compare its performance against related studies that have utilized Long Short-Term Memory (LSTM) networks for stock market prediction. LSTMs have been widely employed in financial forecasting due to their ability to capture long-range dependencies in time-series data. By benchmarking our GRU-based sentiment-enhanced model against LSTM-based approaches, we aim to assess whether GRUs provide a more efficient and accurate framework for integrating sentiment-driven market signals. This comparison will help determine the added value of incorporating sentiment analysis into stock prediction models and provide insights into the most suitable deep learning architectures for this task, including whether GANs can outperform traditional recurrent models in leveraging social media sentiment for financial forecasting.

1 Introduction

The exponential growth of social networking platforms, while fostering global communication and social interaction, has also revolutionised the way financial information influences the markets. These platforms allow investors, analysts, and companies to instantly share their opinions and analysis, creating a valuable source of information on investor sentiment. However, the complexity and volatility of stock markets make predicting share prices particularly difficult.

This study aims to address this gap by investigating the impact of Twitter sentiment on stock price movements in the US market. We leverage RoBERTa, a powerful pre-trained language model, to perform fine-grained and contextualized sentiment analysis. Unlike traditional approaches that rely solely on historical price data, our methodology integrates sentiment-driven features into predictive models to enhance forecasting accuracy. By embedding sentiment signals within our predictive framework, we aim to better capture the relationship between market sentiment and stock price dynamics.

For stock price prediction, we focus on time-series models, specifically Gated Recurrent Units (GRUs), which are well-suited for capturing long-term dependencies in sequential data. Additionally, we explore the incorporation of technical indicators to identify the optimal feature set for each model. To evaluate the performance of our approach, we compare GRUs with an ensemble learning method, Random Forest, assessing their respective strengths in leveraging sentiment data. Furthermore, we introduce a more complex model by combining a Generative Adversarial Network (GAN) with GRUs to examine whether this architecture can outperform both traditional time-series and ensemble-based models.

This research is motivated by key studies that have demonstrated the potential of social networks in stock market prediction. For example, the study *Harvesting Social Media Sentiment Analysis to Enhance Stock Market Prediction Using Deep Learning* highlights the value of combining sentiment analysis with machine learning models, although the approaches used are often based on traditional techniques such as Naive Bayes or SVM.

In this study, we also seek to determine whether GRU models can outperform LSTM models, which have been widely used in related research for stock market prediction. While LSTMs are known for their ability to capture long-range dependencies in sequential data, GRUs offer a more computationally efficient alternative by simplifying the gating mechanisms while still maintaining strong predictive capabilities. By comparing our sentiment-enhanced GRU model against LSTM-based approaches, we aim to assess whether GRUs provide a more effective framework for integrating social media sentiment into stock price forecasting.

Additionally, we investigate whether a GRU model trained without technical indicators can outperform a Random Forest model trained with technical indicators. The underlying assumption is that time-series models like GRUs can inherently capture trends and patterns by remembering past data, whereas a tree-based model like Random Forest lacks this sequential memory and relies on explicit trend-related features. By testing this hypothesis, we aim to understand the relative importance of technical indicators in stock price prediction and whether deep learning models can successfully infer market trends without the need for explicit trend-based input features.

This project focuses on several key research questions:

- What is the influence of Twitter sentiment on daily stock price movements?
- Can ensemble models outperform time-series-specific models in stock market prediction?
- Can the GRU model outperform LSTM in stock price prediction?
- Can a GRU model without technical indicators outperform a Random Forest trained with them?
- Can a GAN-based model achieve better performance than other approaches?

By addressing these questions, we hope to contribute to the development of more robust and accurate stock price prediction models, while exploring the essential role played by social networks in the evolution of financial markets.

2 Related work

Several studies have explored the use of machine learning and natural language processing (NLP) to predict stock prices, incorporating sentiment analysis from social networks in particular. In this section, we present some key works, highlighting their methodologies, datasets, results, and differences with our approach.

- *Harvesting Social Media Sentiment Analysis to Enhance Stock Market Prediction Using Deep Learning (Pooja Mehta et al., 2020)* (1) This study examines how sentiments expressed on social networks can be exploited to predict stock prices. The authors combine classical sentiment analysis approaches such as Naive Bayes, Maximum Entropy (ME), and Support Vector Machines (SVM) with price prediction models such as linear regression, decision trees, and LSTM. Although this study demonstrates the effectiveness of sentiment integration in improving prediction, it relies primarily on traditional sentiment analysis and modeling techniques.

In contrast to this approach, our project uses advanced models such as RoBERTa, a Transformer-based model pre-trained on financial data, to capture specific nuances of financial sentiment. Additionally, we compare the effectiveness of time-series-specific models such as GRUs (Gated Recurrent Units) with ensemble learning approaches such as random forests to determine which method performs best.

- *Stock Market Prediction Using Twitter Sentiment Analysis by S. Mittal and A. Goel (2011))* (2)
  This study applies sentiment analysis and machine learning principles to establish a correlation between 'public sentiment' and 'market sentiment'. The authors use Twitter data to assess collective mood, and then exploit this information, along with past Dow Jones Industrial Average (DJIA) values, to predict stock market movements. To test their approach, they propose a new cross-validation method adapted to financial data. Using self-organising fuzzy neural networks (SOFNNs), they achieved 75.56% accuracy in predicting market movements based on tweets and DJIA values between June and December 2009. However, this method remains limited by the simplification of sentiment into a few fixed categories and by the difficulty of establishing precise causal relationships between public mood and market movements.

Existing research shows considerable potential for using social networks to predict financial markets. However, most studies are limited to traditional machine learning and NLP techniques. Our approach stands out by combining RoBERTa with advanced deep learning models, creating innovative architectures that enhance predictive accuracy and robustness. Additionally, we compare their performance with specialized time series and ensemble learning techniques to achieve more reliable forecasting.

3 Data

For this study, we used 2 datasets found on Kaggle: one composed of tweets related to companies and another one containing stock prices and trading volume for each company.

Dataset Composition

- **Twitter Dataset:**
  - Contains tweets related to the top 25 most-watched stock tickers on Yahoo Finance.
  - Covers a time period from **30 September 2021** to **30 September 2022**.
  - Each record includes the following fields:
    * **Date:** The date the tweet was posted.
    * **Tweet:** The textual content of the tweet.
    * **Stock Name:** The stock ticker (e.g., AAPL for Apple, TSLA for Tesla).
    * **Company Name:** The full name of the corresponding company.
  - Provides a rich source of public sentiment data that can be processed to derive sentiment scores for individual stocks.

- **Stock Market Data:**
  - Contains daily stock prices and trading volume for the same set of companies and time period (**30 September 2021** to **30 September 2022**).
  - Each record includes the following fields:
    * **Date:** The trading date.
    * **Open:** The stock price at the beginning of the trading session.
    * **High:** The highest stock price during the trading session.
    * **Low:** The lowest stock price during the trading session.
    * **Close:** The stock price at the end of the trading session.
    * **Adj Close:** The adjusted closing price, accounting for corporate actions like dividends or stock splits.
    * **Stock Name:** The stock ticker corresponding to the company.
    * **Volume:** The total number of shares traded during the session.
  - Enables correlation of sentiment data with stock market performance.

4 Method

4.1 Data Preprocesing

To ensure effective analysis of the financial related tweets, several pre-processing steps were applied to the raw data.

4.1.1 Text Preprocessing

We began by cleaning the tweets to extract only relevant information. This involved:

- Conversion to lower case: all words in the tweets were converted to lower case to ensure consistency and avoid unnecessary variations due to case.
- Removal of URLs: all URLs have been removed to eliminate non-informative elements.
- Removal of mentions and hashtags: mentions (@user) have been removed, and hashtags have been cleaned up by retaining only the keyword without the '#' symbol.
- Removal of punctuation and special characters: all non-alphanumeric characters have been removed, leaving only the words.
- Removal of stopwords: we removed stopwords using the NLTK library to exclude common words in English that have no semantic value.

In order to obtain a standardised version of the words and reduce variability, we applied lemmatisation using NLTK's WordNetLemmatizer tool. This transforms each word into its basic form, or lemma, while preserving the meaning of the original words. This step helps us to avoid duplicating similar words with grammatical variations.

4.1.2 Date and Time processing

To ensure that the temporal information in the Twitter dataset aligns with stock market activity, we processed the `Date` feature as follows. The `Date` field, which initially contained both the date and time of each tweet, was split into two separate features: `Day` and `Time`. This separation allowed us to better distinguish tweets posted during market hours from those posted outside of trading hours. Additionally, the `Time` feature was converted to **Eastern Time** (ET), corresponding to the timezone of the eastern coast of the United States, to ensure alignment with the operational hours of major U.S. stock exchanges such as NASDAQ and NYSE.

4.1.3 Data Filtering: Selection of Companies Based on Tweet Volume

To ensure the robustness of our analysis, we filtered the companies in the Twitter dataset based on their tweet volume over a rolling window. Specifically, we considered only companies that had a sufficient number of tweets on the current day and the two preceding days. The filtering process was performed as follows:

- For each company, we iterated through the dataset to identify days with zero tweets. If a company had less than a specified threshold of tweets (e.g., 3 tweets) over the current day and the previous two days, it was marked for removal.
- Companies failing to meet this condition were excluded from the dataset, ensuring that only those with sufficient tweet activity remained for further analysis.

This filtering step allowed us to focus on companies with adequate data to draw meaningful correlations between social media sentiment and stock market performance. The resulting filtered dataset was saved for subsequent steps, with only the relevant companies retained.

4.2 Incorporation of Technical Indicators

To enhance the predictive power of our model and provide it with a broader context of market trends, we augmented the dataset with various technical indicators. These indicators summarize stock price developments not only for the current day but also over a historical window, such as the past week or month. The added technical indicators are as follows:

- **Moving Averages (MA):**
  - **MA(7):** The simple moving average of the closing price over the past 7 days.
  - **MA(20):** The simple moving average of the closing price over the past 20 days.
- **Exponential Moving Average (EMA):**
  - EMA is a weighted moving average that places greater importance on recent prices. It is calculated recursively using the formula:

    $$\text{EMA}_t = P_{\text{close}} + (\text{EMA}_{t-1} \cdot (1 - P))$$

    where P_{close} is the closing price and P is a weighting factor.
- **MACD (Moving Average Convergence Divergence):**

- MACD is the difference between the 26-day exponential moving average and the 12-day exponential moving average of the stock price. It provides insights into the momentum of the stock price.

- **Bollinger Bands:**
  - These are volatility bands constructed using the 20-day moving average (**MA(20)**) and its standard deviation. The bands include:
    * **Middle Line:** The 20-day moving average.
    * **Upper Band:** $\text{MA}(20) + 2 \cdot \text{stdev}(\text{MA}(20))$
    * **Lower Band:** $\text{MA}(20) - 2 \cdot \text{stdev}(\text{MA}(20))$
  - Bollinger Bands help capture periods of high and low volatility.

- **Log Momentum:**
  - Log Momentum is defined as the logarithmic difference in the closing price from one day to the next, providing an indication of stock price momentum.

The implementation was performed programmatically to ensure efficiency across multiple companies in the dataset. Using a group-by approach, the technical indicators were calculated for each stock individually. By incorporating these features into the dataset, the model is better equipped to recognize patterns and trends in stock price behavior over time, which is critical for accurate predictions.

4.3 Sentiment analysis

To analyze the sentiment of tweets, we utilized the RoBERTa model (`cardiffnlp/twitter-roberta-base-sentiment`) from the Hugging Face `transformers` library. The sentiment analysis was performed as follows:

- Each tweet was tokenized and passed through the RoBERTa model, which classified the sentiment into three categories: **positive**, **neutral**, or **negative**. The sentiment scores were mapped to numerical values: +1 for positive, 0 for neutral, and -1 for negative.

- The sentiment scores for tweets were then grouped by `Stock Name` and `Date`, summing the scores to compute a **Daily Sentiment Sum** for each stock.

To account for the temporal influence of previous sentiments, the **Daily Sentiment Sum** was modified by incorporating a weighted contribution from the sentiment sums of the previous two days. Specifically, each daily sentiment score was adjusted by adding half of the sentiment values from the prior two days (if available). This modification captures the lingering impact of past sentiment on social media.

Finally, we normalized the **Modified Sentiment Sum** using **Z-score normalization**, which scales the values relative to the mean and standard deviation of the dataset. This step ensures that the sentiment feature is robust to outliers and suitable for analysis.

This sentiment analysis pipeline allowed us to generate a robust sentiment label for each stock on each day, reflecting not only the daily sentiment but also the influence of recent sentiment trends.

4.4 Prediction Models: GRU and Random Forest

In this study, the **GRU (Gated Recurrent Unit)** model plays a central role as the key model for stock price prediction, while the **Random Forest** serves as a simpler baseline model for comparison and performance evaluation.

4.4.1 Random Forest

Random Forest is a machine learning algorithm based on an ensemble of decision trees (5). It combines the predictions of multiple trees to enhance accuracy and reduce the risk of overfitting. While Random Forest is not specifically designed to handle sequential data, its ability to capture non-linear relationships and handle high-dimensional feature sets makes it a suitable baseline model for evaluating the effectiveness of the GRU. By comparing the performance of Random Forest and GRU, we aim to demonstrate the added value of incorporating temporal dependencies and sequence modeling in stock price prediction.

4.4.2 GRU (Gated Recurrent Unit) Model

The GRU model was chosen as the primary focus of this study due to its ability to efficiently capture sequential patterns in time-series data. GRU is a type of Recurrent Neural Network (RNN) optimized for handling long-term dependencies in sequences (4), while mitigating issues like vanishing gradients. Furthermore, we specifically opted for a **Unidirectional GRU**, as the system predicts stock prices based solely on past data, and a Bidirectional GRU would not provide additional benefit in this context.

Comparison with LSTM One of the objectives of this study was to evaluate whether the GRU's simpler structure, compared to the Long Short-Term Memory (LSTM) network, would yield comparable performance while reducing computational complexity. The GRU's ability to achieve similar results with fewer parameters and faster training times makes it a compelling alternative to the LSTM for this task.

Fully Connected Layers The final stage of the GRU model consists of **Fully Connected Layers**, which process a combination of:

- The **context vector** summarizing sequential data.
- Dense embedding vectors for **Stock Name** and **Sentiment Label**.

This design enables the model to capture both temporal dependencies and static relationships, allowing it to learn complex interactions between price trends and categorical attributes. The fully connected layers include:

- A first dense layer with 64 neurons and ReLU activation.
- A second dense layer with 32 neurons and ReLU activation.
- A final dense layer with 1 neuron, without an activation function, to perform the regression task.

Model Comparison By comparing the GRU's performance with Random Forest, we aim to validate the advantage of sequence modeling for stock price prediction. The results of this comparison highlight the GRU's ability to effectively leverage both temporal and categorical data, making it a more sophisticated and accurate tool for this task.

4.4.3 Generative Adversarial Network (GAN) with GRU Layers

In addition to the traditional GRU model, we also explore the use of a **Generative Adversarial Network (GAN)** to predict stock prices, incorporating **GRU layers** in both the generator and discriminator networks. The GAN framework consists of two neural networks, the *generator* and the *discriminator*, that are trained simultaneously through a process of adversarial learning (6).

The goal of the generator is to produce synthetic data (in this case, predicted stock prices), while the discriminator's task is to distinguish between real stock prices and the synthetic ones generated by the model. The generator is trained to improve its ability to fool the discriminator, while the discriminator is trained to become better at detecting fake data. This adversarial process continues until the generator produces high-quality synthetic data that is indistinguishable from the real data.

Architecture: GAN with GRU Layers In this experiment, we utilize a 5-layer GRU network for both the generator and discriminator. The use of GRU layers allows the model to capture temporal dependencies within the stock price data effectively, making it well-suited for time-series prediction tasks.

- **Generator:** The generator network takes random noise as input, followed by a sequence of 5 GRU layers. Each layer processes the temporal data and outputs a prediction of the stock price. The output of the generator is designed to mimic the sequence of actual stock prices as closely as possible.

- **Discriminator:** The discriminator is a neural network that evaluates the authenticity of the data generated by the generator. It also uses a stack of 5 GRU layers to process the input sequence (real or generated) and outputs a binary classification (real or fake). The discriminator's role is to help guide the generator by providing feedback on the quality of its synthetic predictions.

Training the GAN The GAN is trained using the traditional adversarial approach:

- The generator is updated based on how well it can fool the discriminator into classifying synthetic data as real.
- The discriminator is updated to become more accurate at distinguishing real from fake stock prices.

The generator and discriminator are updated alternately, and the training process continues until the generator produces stock price predictions that are indistinguishable from the true values, as assessed by the discriminator.

This approach is particularly interesting because it introduces the concept of generating synthetic stock price sequences, which can then be used to augment training data or explore alternative price movements. By incorporating GRU layers, the GAN model is able to maintain a focus on sequential patterns while benefiting from the generative nature of the architecture.

4.5 Evaluation Metrics: MAE and RMSE

To evaluate the performance of the models used in this project, we relied on two widely used regression metrics: the **Mean Absolute Error (MAE)** and the **Root Mean Squared Error (RMSE)**. These metrics provide complementary insights into the accuracy of the predictions, capturing different aspects of error magnitude and variability.

4.5.1 Mean Absolute Error (MAE)

The **Mean Absolute Error** measures the average magnitude of the errors between predicted and actual stock prices, without considering their direction. It is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where y_i represents the actual value, $\hat{y}_i$ represents the predicted value, and n is the total number of predictions.

MAE provides an intuitive measure of prediction accuracy, as it represents the average absolute difference between predicted and actual values in the same unit as the target variable (e.g., stock price in dollars). Lower MAE values indicate better predictive performance. It is robust to the presence of outliers compared to other metrics like RMSE, as it does not square the errors.

4.5.2 Root Mean Squared Error (RMSE)

The **Root Mean Squared Error** evaluates the square root of the average squared differences between predicted and actual values. It is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE penalizes larger errors more heavily than MAE, due to the squaring of the differences. This makes it particularly sensitive to outliers or large deviations in predictions. A lower RMSE indicates a better fit to the data. While RMSE is more sensitive to extreme errors, it provides a clearer indication of how large the residuals are on average.

4.5.3 Comparison of Metrics

The use of both MAE and RMSE allows for a more comprehensive evaluation of model performance:

- **MAE** is useful for understanding the typical error magnitude in a straightforward and interpretable way.
- **RMSE** emphasizes larger errors and provides insights into the variability of the predictions.

By analyzing these two metrics, we can assess not only the overall accuracy of the models but also their ability to handle extreme deviations in stock price predictions. This dual evaluation framework ensures a robust and balanced comparison of the models under study.

5 Experiment and results

5.1 Random Forest

5.1.1 Train/Test Split and Feature Selection

For our experimentation with **Random Forest**, we employed **GridSearch-Cross-Validation** to identify the optimal hyperparameters, ensuring robust model evaluation while minimizing the negative **Mean Squared Error (MSE)**. Initially, we included all available features in the dataset, including technical indicators. However, the model's performance was suboptimal due to the inclusion of too many features, which led to overfitting. To mitigate this issue, we applied **feature selection techniques** such as **Pearson correlation** and **variance threshold** to eliminate less informative variables. As a result, we removed features like **Volume**, **MA20**, **upper_band**, and **lower_band**, which contributed little to the predictive power of the model.

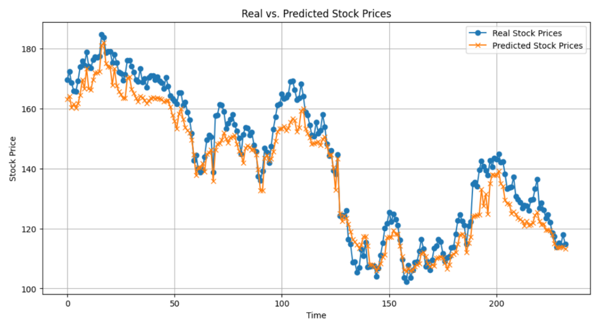

Figure 1: Random Forest Prediction for Amazon Stock without EMA, MA7, logmomentum

Mean Absolute Error (MAE): 5.38

Mean Squared Error (MSE): 38.55
Root Mean Squared Error (RMSE): 6.21

5.1.2 Improving Performance by Adding Technical Indicators and Further Feature Selection

Once the most informative features were selected, we decided to incorporate new technical indicators into our dataset: `EMA`, `MA7`, and `logmomentum`. The inclusion of these features, combined with the application of **feature selection techniques**, significantly improved the model's performance, as these new variables allowed the model to better capture market trends.

Then, to further enhance the model's performance, we utilized **feature selection scores** provided by the **Random Forest** model to refine the feature set even further.

Feature	Importance
High	0.214784
Low	0.204324
logmomentum	0.191997
EMA	0.158470
Open	0.142991
MA7	0.086544
Z_Score_Normalized	0.000634
month	0.000222
day	0.000033

Table 1: Feature Importance

As a result, we excluded the features **month** and **day** from the training data. This ensures that our training dataset is now prepared for use, incorporating the new technical indicators while omitting the less relevant features.

5.1.3 Model Performance Evaluation

The model's performance was evaluated using the following metrics: **MAE**, **MSE** and **RMSE**. For the stock **Tesla**, the results showed a slight improvement by removing the temporal features `month` and `day` :

With `month` and `day`:

- **MAE** : 4.51
- **MSE** : 25.69
- **RMSE** : 5.07

Without `month` and `day` :

- **MAE** : 4.40
- **MSE** : 25.02
- **RMSE** : 5.00

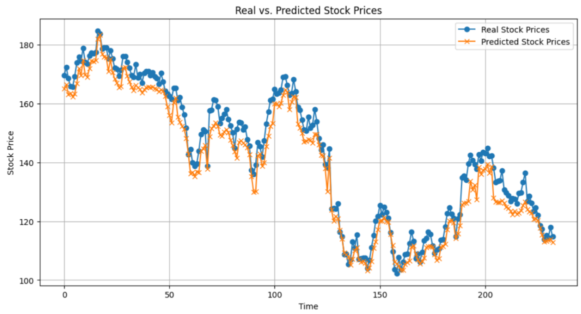

Figure 2: Random Forest Prediction for Amazon Stock with EMA, MA7, logmomentum

The chart and the metrics above illustrate the model's enhanced capability to accurately capture and learn stock market trend patterns.

In conclusion, **Random Forest** showed great efficiency in predicting stock prices, especially with the addition of technical indicators. Careful feature selection and elimination of uninformative variables were crucial to avoid overfitting and improve model accuracy, demonstrating the importance of data preparation in machine learning models for stock price prediction.

5.2 GRU

5.2.1 Architecture of the GRU Model

The GRU approach uses 30-day sequences for each stock, allowing the model to take into account data from the last 30 days for all training stocks. The dataset was divided into 7 companies for training and 1 company for testing. The data was sorted by company and date, with missing values filled in. Numerical features were normalized using `MinMaxScaler`.

To ensure stable training, we set `shuffle=False` in the fit method to maintain the order of the training data. Early stopping was implemented with `patience=50` and `restore_best_weights=True` to halt training when performance starts deteriorating while preserving the best weights. Given the complexity of GRU models and their slower convergence, we set `epochs=250` to allow sufficient training while ensuring computational efficiency through early stopping.

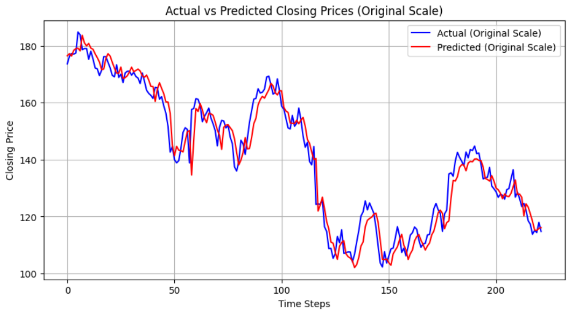

Figure 3: GRU stock price prediction for AMZN

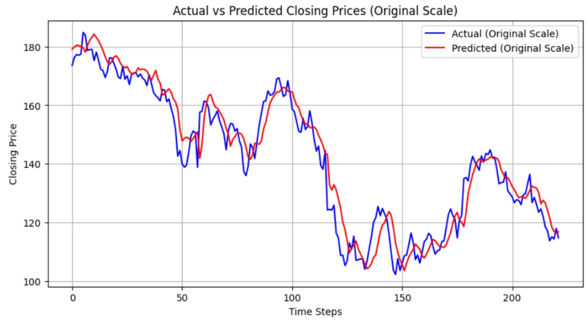

Figure 4: LSTM stock price prediction for AMZN

5.2.2 Results

The following table presents the metric results for various combinations of feature sets. Here, "tech" refers to technical indicators, "sentiment" represents the sentiment feature derived from our sentiment analysis, and "all" encompasses all remaining features in the dataset. The metric results are averaged over 10 runs for the Amazon stock.

GRU Model	MAE	MSE	RMSE
All-tech	4.22	30.76	5.46
All-tech-sentiment	4.14	29.32	5.37
All-sentiment	4.42	34	5.67
All	4.35	32.00	5.58

Table 2: Performance metrics of GRU models.

The results in Table 2 highlight the impact of incorporating sentiment features into the GRU model's performance. When the sentiment feature is removed (as seen in the "All-sentiment"), the error metrics, particularly MAE and RMSE, increase, indicating that sentiment features provide valuable information for improving prediction accuracy. However, the slight decrease in errors across "All-tech" and "All-tech-sentiment" models suggests that the GRU model may struggle to effectively learn from a large number of features. This is likely due to the model's inherent complexity and its sensitivity to feature redundancy or noise. Overall, the results demonstrate that sentiment features enhance predictive performance, but careful feature selection is crucial to avoid overloading the model with too many inputs.

5.2.3 Comparison with LSTM

LSTM performance on the same dataset:

MAE: 4.40
MSE: 30.67
RMSE: 5.51

The results demonstrate that the GRU model outperforms the LSTM model, as initially assumed in our research question. Specifically, the GRU achieved lower error metrics (MAE: 4.14, MSE: 29.32, RMSE: 5.37) compared to the LSTM. This confirms that the GRU's simpler architecture is better suited for the given dataset, providing more accurate predictions while maintaining efficiency.

5.3 GAN with GRU

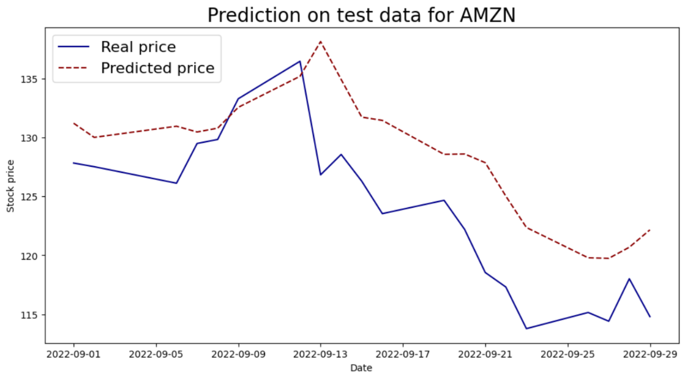

Figure 5: GAN stock price prediction for AMZN

5.3.1 Results

GAN with GRU	RMSE (Average)
All-tech	4.92
All-tech-Sentiment	8.65
All-Sentiment	6.72
All	3.52

Table 3: Performance metrics (RMSE) for GAN (with GRU) model using different feature combinations. Test on AMZN stock

The results presented in Table 3 demonstrate that the GAN model, when combined with GRU, generalizes effectively when utilizing all features. The "All" configuration achieves the lowest RMSE

(3.52), indicating that this complex model is capable of handling a large number of features while maintaining outstanding predictive performance. Furthermore, the significantly higher error observed in configurations without the sentiment feature, such as "All-tech-Sentiment" (RMSE: 8.65) and "All-Sentiment"(RMSE: 6.72), highlights the crucial role of sentiment data in predicting the closing price. This underscores the importance of integrating the sentiment feature to enhance the model's accuracy and its ability to capture market dynamics effectively.

5.3.2 Comparison with Related Works

A comparative analysis was conducted against a related study that utilized the same initial dataset, as documented in their publicly available notebook (3). The referenced work implemented a **GAN with LSTM** architecture and reported a test **RMSE of 3.81** for Amazon stock price prediction.

In contrast, our proposed model, which combines a **GAN with GRU**, achieved a **lower RMSE of 3.52** on the same test case. This improvement highlights the effectiveness of the GRU-based architecture in capturing temporal patterns and processing sequential data more efficiently, resulting in enhanced prediction accuracy. The findings confirm that GRU's simpler structure and computational efficiency make it better suited for this specific stock prediction task when integrated with a GAN.

6 Discussion and conclusion

This study addressed several research questions regarding the efficacy of various machine learning approaches for stock market prediction. The results provide valuable insights into the predictive power of advanced architectures and feature combinations.

The incorporation of the sentiment feature, derived using RoBERTa, proved to be a critical factor for achieving better predictions and reducing errors across all stocks. Pre-trained models like RoBERTa demonstrated their ability to better capture specific financial sentiments compared to traditional approaches, enhancing the overall predictive performance.

In comparing ensemble models and time-series-specific models, it was observed that while Random Forest produced satisfactory results, it was outperformed by GRU, a time-series-specific model. The GRU model demonstrated a superior ability to handle temporal data, confirming its advantage over ensemble approaches for this task.

The GRU model further outperformed LSTM, validating our hypothesis about its efficacy in stock price prediction. This improvement can be attributed to GRU's simpler architecture, which made it less prone to overfitting and better suited for our dataset.

A key finding was the performance of the GRU model without technical indicators. Remarkably, it outperformed the Random Forest model trained with technical indicators, underscoring GRU's ability to internally store and leverage past stock market trends without relying on explicit financial trend features.

Finally, the GAN-based model with GRU emerged as the best-performing architecture for this dataset. It achieved outstanding results across all stocks and demonstrated the ability to effectively manage all dataset features without requiring extensive feature selection. Furthermore, this approach outperformed a related study that employed a GAN with LSTM, showcasing the innovative potential of combining GAN and GRU architectures for stock price prediction.

In conclusion, the integration of sentiment analysis, careful feature selection, and advanced architectures like GRU and GAN enabled superior predictive performance. These findings contribute to advancing methodologies for stock market prediction and highlight the potential of combining pre-trained models with innovative machine learning architectures.

7 Additional Notes

The use of AI in this paper has been limited solely to improving the clarity and readability of the text, without influencing the research content, data analysis, or conclusions.

8 References

References

[1] Mehta, Pooja, et al. Harvesting Social Media Sentiment Analysis to Enhance Stock Market Prediction Using Deep Learning. Available at: https://www.researchgate.net/publication/350847987_Harvesting_social_media_sentiment_analysis_to_enhance_stock_market_prediction_using_deep_learning.

[2] Stock Market Prediction Using Twitter Sentiment Analysis by S. Mittal and A. Goel (2011) Available at: https://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf.

[3] A comparative analysis was conducted against a related study that utilized the same initial dataset, as documented in their publicly available notebook Available at: https://www.kaggle.com/code/equinxx/stock-prediction-gan-twitter-sentiment-analysis

[4] Gated Recurrent Neural Tensor Network by Andros Tjandra et al. (2017) https://arxiv.org/abs/1706.02222

[5] The Random Forest Algorithm for Statistical Learning by David J. Cutler et al. (2011) https://journals.sagepub.com/doi/full/10.1177/1536867X20909688

[6] Time Series Generative Adversarial Networks by Jinsung Yoon, Daniel Jarrett, Mihaela van der Schaar https://arxiv.org/abs/1909.01264