Predicting Stock Market Trends

Youssef Briki
youssef.briki@umontreal.ca

Yuan Li
yuan.li@umontreal.ca

Xintian Xu
xintian.xu@umontreal.ca

Abstract

This project aims to build robust predictive models capable of forecasting stock prices or market trends over short and long-term periods by using advanced machine learning techniques on historical data and social media and news data.

1 Introduction

The stock market is influenced by many aspects, such as government and economic policies, company and consumer behaviours, public opinions, and more. These sentiments about the future trend of the market are often reflected in general and financial news, company earning calls, and public discussions on social media. This project aims to compare different approaches to building a multimodal deep learning model that can predict the short-term stock market behaviour. It focuses mainly on sentiment analysis on textual data such as news, and social media posts, and is supplemented by analysis on other types of data such as historical stock price data.

2 Related Work

2.1 Sentiment Analysis with FinBERT

Although there exist models that can perform well in general sentiment analysis tasks, they may be more lacking in domain-specific tasks due to specialized vocabulary and data of that domain. In stock prediction, an understanding of finance language is crucial and one of the main challenges for sentiment analysis. Therefore, one of the models that we will utilize is FinBERT Araci (2019), a version of BERT Devlin et al. (2019) pre-trained on financial corpus and fine-tuned for sentiment analysis for finance. It was found that FinBERT performed better than Vanilla BERT and other language models on finance tasks. This paper also discusses the uniqueness of sentiment analysis in finance compared to general sentiment analysis tasks, namely that in finance the goal is to utilize sentiment scores to forecast the market. A large part of this project makes use of FinBERT to acquire sentiment score prediction for financial news and social media data, and the goal is to compare the performance of stock price prediction against another general purpose LLM, namely DeepSeek-R1.

2.2 Financial Sentiment Analysis: Techniques and Applications

In this paper by Du et al. (2024), a review of studies on techniques and applications of financial sentiment analysis is conducted. It discusses the interaction between financial textual sentiments, investor sentiments, and other market information with the market itself and stock prices. Textual sentiment information can be classified as subject or objective, in which the objective is often official information such as macroeconomic data to which investors later react, providing subjective data. Together, these two types of data provide sentiments that can influence stock market behavior. Various techniques used for financial sentiment analysis are reviewed, including machine learning, deep learning, and pre-trained models. Benchmarks such as PhraseBank, FiQA Task 1, SemEval 2017 Task 5 etc. are used for comparison. It provide a comparison on how different techniques and models perform on the described benchmark datasets, it provides an overview and knowledge foundation for which we leveraged and considered in order to build our project.

2.3 DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

This paper presents DeepSeek-R1 DeepSeek-AI and et al. (2025), an LLM which we fine-tuned for sentiment analysis on financial data specifically.

DeepSeek-R1 incorporates multi-stage reasoning and cold-start data to its large-scale reinforcement training approach. In the paper, the results of DeepSeek-R1 show that it performs particularly well on reasoning tasks, comparable to OpenAI-o1-1217. However, despite its powerful performance, DeepSeek-R1 is a general purpose LLM, this means that unlike FinBERT, it has not been fine-tuned on financial text data and may not perform as well on finance domain sentiment analysis tasks. Therefore, our project builds upon the work of DeepSeek-R1 and attempts to fine-tune it to provide better results on financial tasks.

3 Method

In Figure 1, the diagram shows an overview of our models architecture and the workflow we used for our experiments. In short, we first preprocess the textual data, and input them into FinBERT or DeepSeek-R1 fine-tuned on financial datasets as sentiment analysis regression task. Once we obtained the numerical sentiment scores, we combined them with the historical stock price, while matching each text data sample to historical price point by date and stock ticker. Finally, we train the LSTM on this dataset. With the LSTM, we can make inference calls by providing a specific ticker, for example AAPL for Apple or NVDA for Nvidia, and get the stock price prediction for that company for the next business day relative to the latest date available in the training data. Therefore, our model is a short-term stock price prediction model.

3.1 FinBERT for Financial Sentiment Analysis

FinBERT is a fine-tuned version of BERT, and because of its exceptional performance on financial text data, we used it for acquiring sentiment scores for financial news and reddit posts and use this model as a threshold for comparing the performance of DeepSeek-R1. Unlike for DeepSeek-R1, we did not further fine-tune FinBERT as it is already fine-tuned with finance specific data.

3.2 Fine-tuning Deepseek for Financial Sentiment Analysis

In this project, we used the parameter-efficient fine-tuning technique, DoRA (Differentially Over-parameterized Reparameterization), to adapt the DeepSeek R1 architecture for the task of financial sentiment analysis. DoRA (Differentially Over-parameterized Reparameterization) is a parameter-efficient fine-tuning (PEFT) technique . Unlike full fine-tuning, which updates all model parameters, DoRA decomposes the weight matrices of the Transformer layers into two components: a low-rank intrinsic component that captures the task-specific aspects and a magnitude component that primarily retains the pre-trained knowledge. During fine-tuning, only the low-rank component is updated, while the magnitude component remains largely frozen. This differential treatment of the weight components allows DoRA to achieve performance comparable to full fine-tuning with a fraction of the trainable parameters.

We selected two distinct model sizes from the DeepSeek R1 family to investigate the impact of scale on performance: the DeepSeek R1 with 1.5 billion parameters and the larger DeepSeek R1 model with 7 billion parameters. Both of these DeepSeek R1 models are built upon the Qwen 2 base model, inheriting its foundational linguistic capabilities.

3.3 LSTM for Stock Prediction

In this project, we employed a Long Short-Term Memory (LSTM) neural network to model stock price prediction across multiple companies. LSTMs are a type of recurrent neural network (RNN) particularly well-suited for time-series forecasting tasks due to their ability to capture long-term dependencies and temporal patterns.

We designed a shared LSTM model that incorporates both sequential market features and company-specific embeddings:

For each time step, the model takes a sequence of 10 days of the following normalized features: close (closing price),news_finbert_score (news sentiment score),social_finbert_score (social sentiment score) volume (traded volume)

Each company is assigned a learnable embedding vector. This allows the model to capture unique company characteristics and trends without training separate models.

LSTM model includes three layers. Embedding Layer transforms company IDs into fixed-dimensional vectors. LSTM Layer accepts concatenated sequences of market features and company embeddings. Hidden size is set to 50. Linear Layer maps the LSTM output at the final time step to a single predicted value (scaled closing price).

The raw data was grouped by ticker (company symbol), and for each company, sequences of 10 consecutive days were created. Corresponding targets were the closing price of the day immediately following each sequence. All features were normalized using MinMaxScaler to stabilize the training process.The dataset was then split into training (80%), validation (10%), and test (10%) sets for each company. The model was trained using a batch size of 64 and optimized with Adam optimizer and MSE loss. Early stopping was employed based on validation loss to prevent overfitting.

The model was trained for up to 20 epochs with a patience of 5. During each epoch, the model was evaluated on the validation set. Loss curves for both training and validation sets were plotted to monitor convergence.

Evaluation metrics included: MAE (Mean Absolute Error),RMSE (Root Mean Square Error),R^2 Score,Directional Accuracy: Measures how often the predicted direction (up/down) matched the true direction.

Predictions were made on training, validation, and test datasets. Results were inversely transformed from scaled values to actual stock prices for interpretability. Visualizations were generated to compare predicted vs. actual trends and to illustrate the model's performance per company.

4 Experiments

4.1 Datasets

For the data, we used a combination of financial news data from various sources, social media data from Reddit, and historical daily stock price from the past 100 days. The textual data is used to acquired sentiment scores that reflects the current market behaviour and investor sentiments, and it will be joined with the stock price data for sentiment-based time-series prediction to get the next-day stock price.

4.1.1 Financial News

The financial news data we acquire is from the Alpha Vantage API . For this part, we used a paid subscription to get more API calls in order to have access to data in larger quantity and quality. With this API, we should be able to acquire live data and create a live prediction model that refreshes the training data each day. However, in order to save costs, we unsubscribed from the API provider and we are working with an already downloaded version of the dataset, and it contains data up to Friday, April 25, 2025. Therefore, our models will demonstrate the prediction capabilities for the next business day, which is Monday April 28, 2025.

In total, we acquired financial news data for 50 stock tickers, 47 of which are individual companies, 3 of which are market indices which represent the market-wide trend, and 8 topics. In the API, each news article has a list of tickers and topics it is related to, and thus for one news article, it may be retrieved several times because it is related to more than one of the 50 ticker or topics. The news supposedly dates back to January 1, 2019, depending on availability.

Although the API provides sentiment scores for each news article, we decided to not utilize them because it is unclear how these scores are generated and with which model. Therefore, we treated the data as raw text and used either FinBERT or DeepSeek-R1 to get sentiment scores for the price prediction.

In total, there are 16473 news articles for the 50 stock tickers.

4.1.2 Social Media - Reddit

To incorporate public sentiment into our stock prediction model, we collected social media data from Reddit.A custom Python-based scraping tool was developed using the `praw` (Python Reddit API Wrapper) library.

The script retrieves a specified number of top posts from selected subreddits related to finance and investing (e.g., `r/stocks`, `r/investing`). For each post, the following information is extracted:

- Post ID, title, and description (self-text)
- Upvote and downvote counts
- Subreddit name
- Top user comments (up to a specified limit)
- Associated images (if available)
- Timestamp of post creation

Posts are filtered based on popularity (hot posts), and top-level comments are collected by flattening the comment tree using `post.comments.list()`.

The collected data is structured using a custom `RedditPost` schema and saved in JSON format for further preprocessing. Timestamps are serialized using ISO format for consistency. Each day's data is saved in a separate file named by date

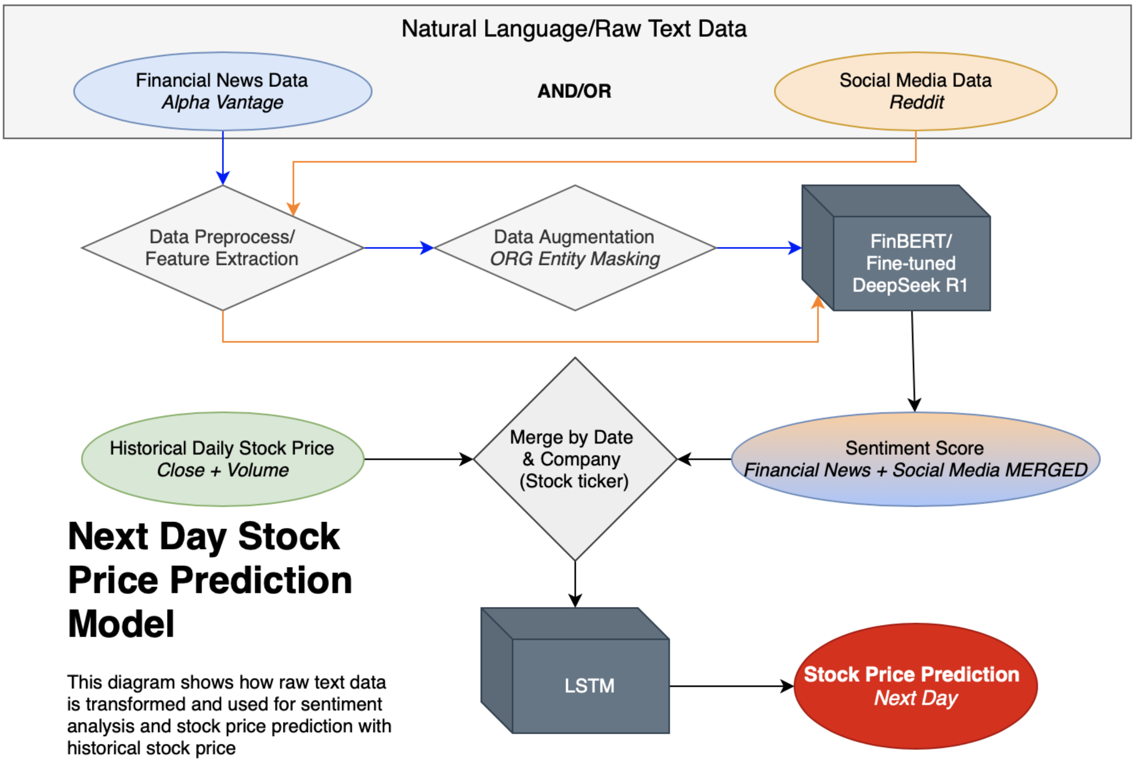

Figure 1: Diagram of the stock prediction model

(e.g., `reddit_2025-04-30.json`) for reproducibility and ease of analysis.

4.1.3 Natural Language Data Preprocessing and Augmentation

Since the raw data we acquire from Alpha Vantage and webscraping Reddit contained a lot of irrelative features, we extracted only the relevant ones for our purpose. For news data, this includes `ticker`, `title`, `summary`, `date`. For reddit, this includes `title`, `description`, `comments`.

In addition, data augmentation was applied to financial news. The techniques include synonym replacing, organization entity masking where the company names or tickers are masked with `[ORG]` in the text, and sentence reordering. Experiments were carried out with each one of the techniques and all three combined. Here is an example of the most effective data augmentation technique, entity masking.

> **`Original:`** `Huawei announced its preparing a new AI chip as an alternative to Nvidia's H20 line.`
> **`Augmented:`** `[ORG] announced its preparing a new [ORG] chip as an alternative to [ORG] 's H20 line .`

Data augmentation was only applied to 30% of the news articles, and the augmented data was added to the data as additional data samples.

4.1.4 Historical Stock Price

The daily historical stock price is also acquired from Alpha Vantage API, and it is acquired per stock ticker. The features that are acquired for each ticker include `date`, `open`, `high`, `low`, `close`, `volume` for each day, with the last 5 being stock price in US dollars. The feature we predict for the next day stock price is the closing price, ie. `close`.

4.2 Baselines

We will use finBERT + LSTM without data augmentation as a baseline and compare it to the models with data augmentation and to the models which use DeepSeek-R1 as sentiment analysis model.

4.3 Evaluation Methods

To assess the performance of the LSTM model for stock price prediction, we use the following evaluation metrics:

- **Mean Absolute Error (MAE):** MAE calculates the average of the absolute differences between predicted values $\hat{y}_i$ and actual values y_i.

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

  The smaller the MAE value, the smaller the model's prediction error and the better the model's performance.

- **Root Mean Square Error (RMSE):** RMSE is the square root of the average squared differences between predicted and actual values.

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

  The smaller the RMSE value, the smaller the model's prediction error and the better the model's performance.

- **R-squared (R^2 Score):** The R^2 score measures the proportion of the variance in the actual values that is explained by the predictions.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

  The closer R² is to 1, the better the model fits the data and the higher the proportion of data variability that the model can explain.

- **Directional Accuracy:** This metric evaluates how often the model correctly predicts the direction of price change (up or down).

$$\text{Directional Accuracy} = \frac{\text{Number of correct direction predictions}}{\text{Total number of predictions}}$$

  Directional accuracy is important if the purpose of the model is to predict whether the stock price will go up or down. However, we noticed that all our models have very low error metrics and high R-squared, which means the models predict stock price very closely to the true price, but very low directional accuracy if the dataset is small. We believe that this is because of tiny fluctuations and noises, but the actual magnitude of fluctuations are negligible and does not impact the actual prediced price. In fact, the directional accuracy is high when we have a dataset with enough historical price points. Therefore, we pick the experiment 5 with FinBERT and entity masking model as the best model.

4.4 Experiment Setup

4.4.1 Stock Price Prediction with FinBERT as Sentiment Analysis Model

To evaluate the impact of sentiment analysis on stock price prediction, we conducted a series of experiments using a Long Short-Term Memory (LSTM) model. First, we conducted the following experiments on data containing only historical stock prices and news. We then repeated the above experiments after merged social media data.

1. **Experiment 1: (Stock Close Price Only)**
   Only historical stock data (Close price) was used as input to the LSTM model.

2. **Experiment 2: (Stock Close Price + FinBERT Score)**
   Historical stock data (Close price) and sentiment score got by finBERT were used as input to the LSTM model.

3. **Experiment 3: (Stock Close Price + Volume + FinBERT Score)**
   Use more historical stock data (Close price, Volume) and sentiment score got by finBERT to predict.

4. **Experiment 4: Synonym Replacement Augmented Data: Close + FinBERT Score + Volume**
   Use same features as experiment 3, but experiment on synonym replacement augmented data.

5. **Experiment 5: Organization Entity Masking: Close + FinBERT Score + Volume**
   Use same features as experiment 3, but experiment on organization entity masking augmented data.

6. **Experiment 6: Sentence Reordering : close + FinBERT score + volume**
   Use same features as experiment 3, but experiment on sentence reordering augmented data.

7. **Experiment 7: Synonym, Entity Mask, Sentence Reorder : close + FinBERT score + volume**

Use same features as experiment 3, but experiment on data used three methods.

The results indicate that integrating sentiment analysis improves stock price prediction. And compare the results of experiment 4 to experiment 7, we found that the experiment 5 which use organization entity masking augmented method get best performance.

Based on experiment result, we choose FinBERT-entity-masking model to build the subsequent agent flow

4.4.2 DeepSeek-R1

DeepSeek-R1 is an open-source model, based of the Qwen2 architecture. It's famous for it's reasoning abilities. The model achieves SOTA performance in tasks such as math and logic. Here, we procede to do a stock Price Prediction with DeepSeek-R1 as Sentiment Analysis Model. This part is similar to the previous one, except that the fine-tuned deepseek model is used to calculate the sentiment score.

4.4.3 Deepseek fine-tuning

Zero-shot performance: We have run zero-shot experiments using both Deepseek-R1 32b and Deepseek-R1 1.5b. The results are below:

Fine-tuning The low results above for both models made it obvious for us to consider fine-tuning our models. We have conducted two experiments, with two models of different sizes (Deepseek r1 1.5 b and 7 b). We have used a Dora fine-tuning, which is a lightweight, parameter-efficient approach that leverages on-the-fly data augmentation and adaptive regularization to adapt pre-trained language models to new domains, on the FinancialPhraseBank dataset, which is a benchmark corpus of roughly 15000 sentences drawn from financial news and manually annotated with positive, negative, and neutral sentiment labels. We have only selected approximatively 5000 sentences, which were labeled the same by atleast 75 per-cent of the participants.

We notice that the smaller models achieve classification metrics above 80 per-cent. The Deepseek R1 1.5B performs just as well as the 7B parameter model. Moreover, the smaller models even outperform the much larger Deepseek R1 32B on certain metrics, such as the F1 score and precision for both negative and positive classes. **For the experiments, we then be using Deepseek r1 1.5B**

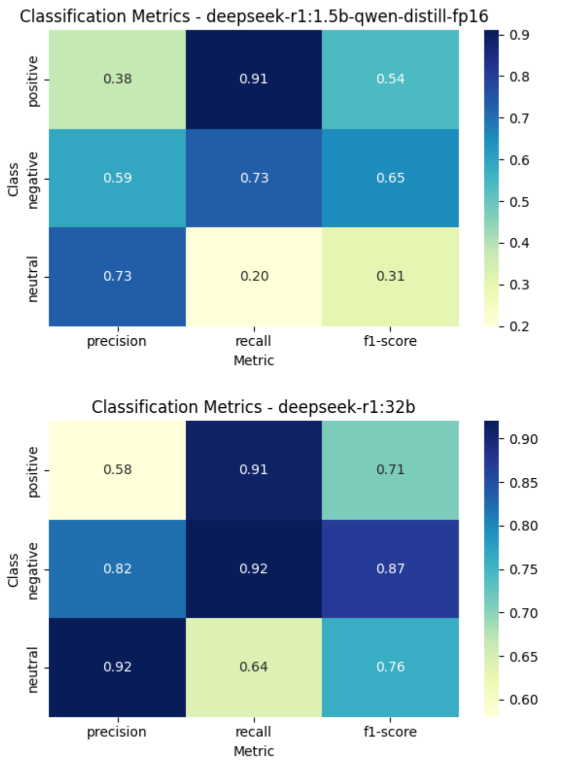

Figure 2: Results of zero-shot for Deepseek R1 1.5b and 32b

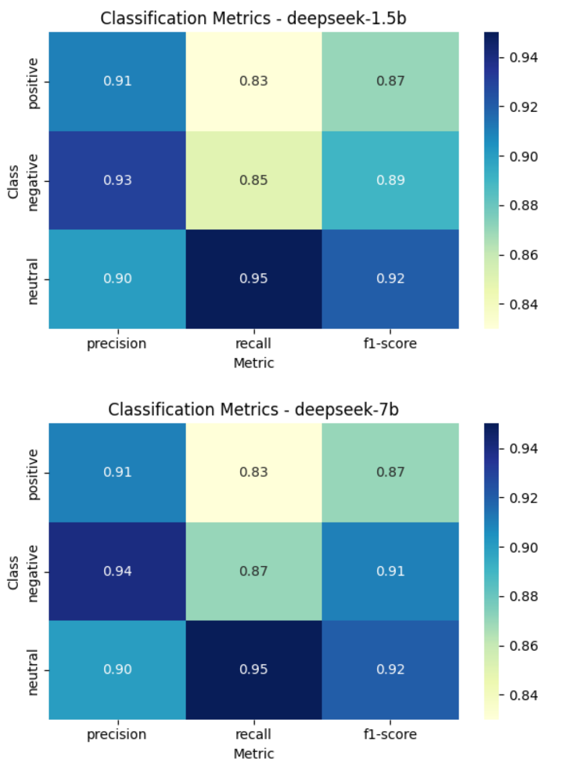

Figure 3: Results of fine-tuned zero-shot for Deepseek R1 1.5b and 32b

as it has the pefect balance between accuracy and ressource effeciency.

Hardware requirements

Training was done on four A100 40 GB GPUs for 12 h for Deepseek 7B. Training was done in 1 h 30 min on four A100 40 GB GPUs for Deepseek 1.5B.

On four A100 GPUs, we were able to achieve incredible results when it comes to inference time. Running our data in parallel on all four GPUs yielded incredibly fast results. Each second we performed around 32 inferences, with 10 GB of VRAM used and 100 percent of computing resources being exploited. We managed to use our resources well.

4.5 Results & Analysis

The results show that out of the FinBERT-based models, the one which uses organization name entity masking as data augmentation for acquiring sentiment scores achieves the best results in LSTM stock prediction. Therefore, it seems that masking the company names is reducing the amount of noise present in textual data. In addition, in all the models, even though the MAE, RMSE, and R^2 are indicating good performance, the directional accuracy is low. This is because the model predicts a very close price to the actual price, but in the time series, the predictions has tiny fluctuations which does not correspond to the direction of the real price movememnt. However, for the purpose of our project, which is to predict short term future price, the predicted price being close to the actual price is more important than predicting the correct direction of the price movement. In addition, the tiny fluctuations causing the low directional accuracy may simply be due to noises, and thus smoothing techniques can be applied either as penalties in training or post-processing.

After merged social media data,model performance deteriorates. The possible reason is that social media data introduces noise to the model. Unlike structured financial news or reports, social media texts often contain informal language, slang, abbreviations, emoticons, and internet jargon, which may be significantly different from the language patterns learned by the model in the financial field. In addition, the brevity of social media posts may lead to incomplete information and missing key context, which makes accurate sentiment assessment challenging. In the future, we can consider some data cleaning and noise filtering strategies to mitigate the negative impact of social media noisy data.

In addition, experimental results on deepseek show that the fine-tuned DeepSeek model achieves similar performance to finBERT.

Figure 4 is the example prediction graphs for Apple (AAPL) on the training, validation, and test set for the best FinBERT Model, **FinBERT-entity-masking**.

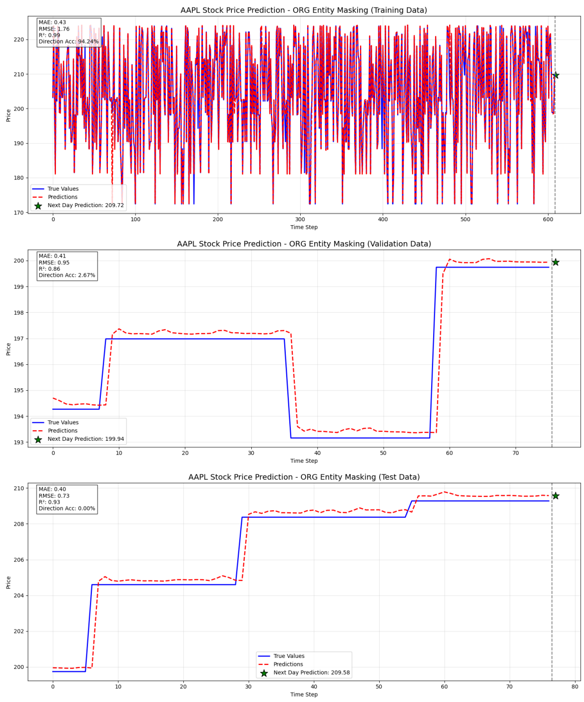

Figure 4: Example of Apple (AAPL) stock price prediction using the news-only-FinBERT model with entity masking augmentation

5 Agentic Workflow

We refactored our models into an agentic style workflow, where the user can query the agent for a specific company's stock price using the ticker, and the agent would train or use an existing model to get the next day prediction for that company's stock. The user can also specify which model or dataset to use for the sentiment analysis, such as texttnews-only-finbert, news-socialmedia-finbert, for FinBERT based model that uses either news only data, or also social media data. Figure 5 is an example of the next day price prediction returned by the agent for Nvidia (NVDA).

Model	MAE	RMSE	R^2	Directional Accuracy
News Only FinBERT Models				
close-price-only	0.0180	0.0453	0.9591	8.70%
FinBERT-close-price	0.0178	0.0450	0.9595	9.63%
FinBERT-close-volume-price	0.0224	0.0461	0.9577	9.01%
FinBERT-synonym	0.0179	0.0381	0.9831	6.96%
FinBERT-entity-masking	0.0135	0.0367	0.9843	7.72%
FinBERT-sentence-reorder	0.0185	0.0416	0.9803	9.08%
FinBERT-all3-augmented	0.0238	0.0404	0.9810	7.01%
News + Reddit FinBERT Models				
FinBERT-close-price	0.0131	0.0326	0.9844	3.54%
FinBERT-close-volume-price	0.0111	0.0320	0.9850	3.75%
FinBERT-synonym	0.0110	0.0320	0.9850	3.70%
FinBERT-entity-masking	0.0140	0.0331	0.9839	3.70%
FinBERT-sentence-reorder	0.0157	0.0332	0.9838	3.62%
FinBERT-all3-augmented	0.0165	0.0345	0.9825	3.65%
News Only DeepseekModels				
Deepseek-close-price	0.0209	0.0460	0.9577	9.38%
Deepseek-close-volume-price	0.0181	0.0450	0.9596	8.95%
News + Reddit DeepseekModels				
Deepseek-close-price	0.0074	0.0190	0.9924	2.59%
Deepseek-close-volume-price	0.0068	0.0187	0.9926	2.55%

Table 1: Prediction models results on test set; all metrics are calculated over all stock tickers aggregated and not for just one ticker.

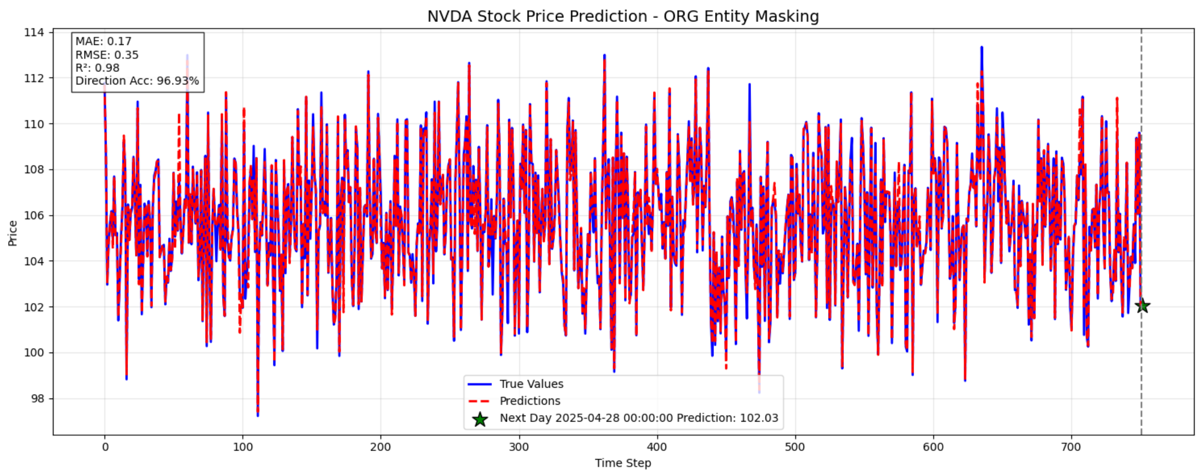

Figure 5: Example of Nvidia (NVDA) stock price prediction using the news-only-FinBERT model

6 Conclusion

This project attempts to use a fine-tuned large language model, DeepSeek, for sentiment analysis of financial news and compares its effectiveness with using FinBERT to extract sentiment scores for stock price prediction. Experimental results show that the fine-tuned DeepSeek model is able to capture more detailed semantic and sentiment information from financial texts, achieving similar performance to finBERT.

7 Contributions of Team Members

- Basic project and model logic design in first proposal (sentiment analysis + LSTM time-series workflow) - **Xintian**, **Yuan** worked in researching datasets, feasibility and completing the design.
- DeepSeek-R1 experiments and agentic workflow proposal - **Youssef**
- Data acquisition: **Xintian** - Alpha Vantage financial news and stock price, **Youssef** - social media Reddit and bluesky, **Yuan** - macroeconomic data
- Data preprocessing and data augmentation: **Yuan** - clean up and merged data from Alpha Vantage and Reddit, **Youssef/Xintian** - data augmentation for DeepSeek and FinBERT respectively
- LSTM and price prediction - **Yuan** developed the training and prediction code for LSTM
- Running FinBERT Experiments - **Xintian and Yuan**, **Yuan** focused on baseline no data augmentation experiments, **Xintian** focused on developing data augmentation pipelines and running experiments on augmented data
- Running DeepSeek experiments - **Youssef**
- Agentic workflow - **Xintian** developed the base logic and code for agent, **Yuan and Youssef** added their experiments and models to the agent.
- Report writing and presentation preparing - all

8 Link to GitHub Repo

https://github.com/youssefbriki1/IFT6289-project/tree/main

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *Preprint*, arXiv:1908.10063.

DeepSeek-AI and Daya Guo et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186.

Kelvin Du, Frank Xing, Rui Mao, and Erik Cambria. 2024. Financial sentiment analysis: Techniques and applications. *ACM Comput. Surv.*, 56(9).