Stock Prediction Using AI

Luciana Alvarado
Jaelen Galindo
Mahya Mottaghi

Research Problem

Project Problem: This project explores AI-driven approaches to enhance stock price forecasting accuracy.

Stock market is notorious for volatility, complexity, and non-linear trends. This creates the task of predicting the unpredictable. This challenge is especially difficult because stocks are influenced by several factors:

- Investors’ sentiment
- Global economic conditions
- Unplanned events
- Company’s financial performances

However, AI has been introduced into the scene in order to help with stock prediction shifting the way stock investing and trading is conducted.

METHODS

Different Models

Traditional Models

Linear Regression (scikit-learn)

- Baseline model fitting a linear relationship between features and price
- Fast and interpretable but limited for complex trends

Long Short-Term Memory (TensorFlow/Keras)

- Recurrent neural network specialized for sequences (financial time series)
- Sequence-based neural network modeling temporal dependencies
- Captures trends and seasonality in stock prices (architecture: 2 LSTM layers with 50 units, dropout 0.2)

Advanced Approaches

Graph Neural Networks (PyTorch Geometric)

- Represents stocks as nodes with edges denoting relationships (e.g., industry)
- Captures market structure alongside price history

Reinforcement Learning (RL Agent, stable_baselines3)

- Models trading as sequential decision-making
- Agent learns to maximize cumulative return through simulated trading environment

Emerging AI

Generative AI (e.g., GANs for Stocks , Keras)

- Generative models learn the data distribution to create realistic new examples (e.g. synthetic price series)
- Generative Adversarial Networks (GANs): Two-part model – a Generator tries to forecast future prices, a Discriminator critiques them against real data
- Generates synthetic price data to test robustness and improve model generalization

Large Language Models (LLMs for Finance, pre-trained BERT model)

- Utilizes advanced NLP models to interpret textual data relevant to stocks
  Analyzes news articles, earnings reports, social media etc. to gauge sentiment and context that influence market moves
- Extracts sentiment signals from financial news and social media to augment numeric models

EXPERIMENT

Design & Set-up

Dataset and Preprocessing

Stock Data Source

- Historical daily price data pulled using the “yfinance" API
- Companies: AAPL, GOOGLE, MSFT, AMZN, TSLA, etc.
- Data Timeframe: January 2020 – January 2024

Preprocessing Steps

- Remove missing values
- Apply Min-Max normalization for scaled inputs
- Create lag/step feature for time series models (e.g., LSTM)
- Format data for use with multiple model types (sequences, graphs, etc.)

Evaluation Metrics

Error Metrics

- **MSE** (Mean Squared Error): Penalizes larger errors more heavily
- **RMSE** (Root MSE): Common scale for comparing to real price values
- **MAE** (Mean Absolute Error): Easier to interpret as average deviation

Behavior & Trend Metrics

- **Directional Accuracy**: Measures whether the model correctly predicted up/down movement
- **Pearson Correlation**: Evaluates alignment of predicted and actual trends
- **Net Worth / Cumulative Return (for RL)**: Evaluates strategy performance over time

Metric Usage

- **Traditional models** (e.g. Linear Regression, LSTM): use error metrics
- **RL agent**: evaluated using profit and final net worth
- **LLM**: sentiment trend correlation with market movement

Implementation Tools & Model Setup

Development Environment

- Developed in Jupyter Notebook using Python
- Tools/libraries: numpy, MatPlotLib, Pandas, scikit-learn, TensorFlow, PyTorch Geometric, yfinance, etc

Code Sources & Adaptation

- Preprocessing feeds into a shared experimental framework
- Each model receives the appropriate input format (e.g., sequences for LSTM, graph structure for GNN)
- Evaluation and visualization scripts will standardize performance comparison across models

Development Sources

- Some models adapted from public Kaggle and GitHub repositories
- Modified and integrated into a unified testing pipeline for consistency

RESULTS

Figures & Data

Long Short Term Memory vs. Apple

In this figure, we see that the results are fairly off

Mean Squared Error:

- Measures average squared difference between predicted and actual values
- 0.0142

Mean Absolute Error:

- Measures the average absolute difference between predictions and actuals
- 0.1143

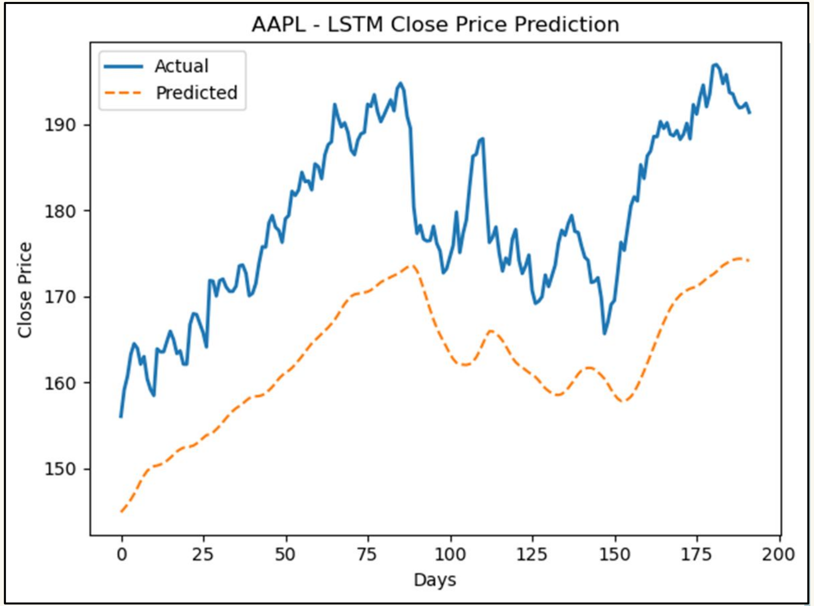
Coefficient of Determination:

- Measures how well the model explains the variance in the target
- -1.8908

Overall: poor model

Why:

- AAPL was more volatile or noisy, making it harder for the model to find patterns.



Long Short Term Memory vs. Google

In this figure, we see that the results are fairly close

Mean Squared Error:

- Measures average squared difference between predicted and actual values
- 0.0039

Mean Absolute Error:

- Measures the average absolute difference between predictions and actuals
- 0.0525

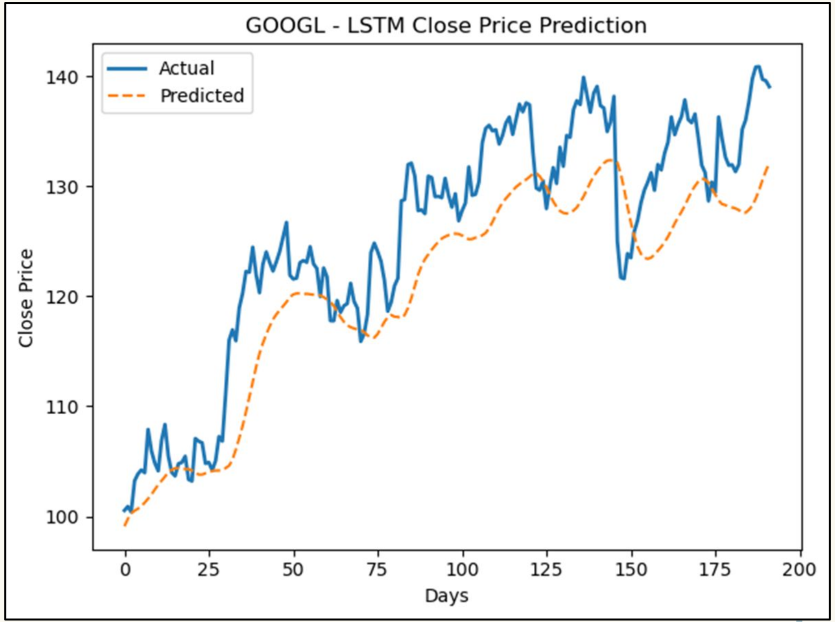
Coefficient of Determination:

- Measures how well the model explains the variance in the target
- 0.6823

Overall: good model

Why:

- GOOGLE could have smoother trends or clearer seasonality.



Long Short Term Memory vs. Tesla

In this figure, we see that the results are fairly close

Mean Squared Error:

- Measures average squared difference between predicted and actual values
- 0.0013

Mean Absolute Error:

- Measures the average absolute difference between predictions and actuals
- 0.0283

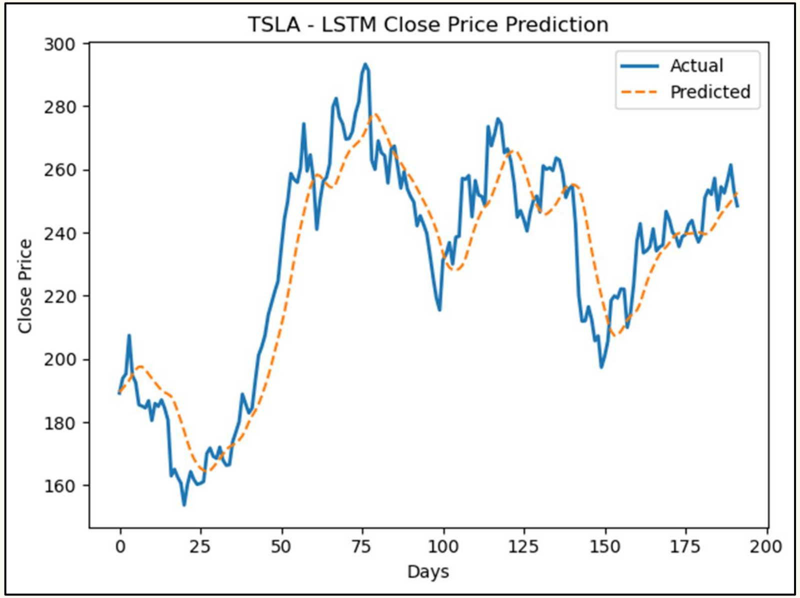
Coefficient of Determination:

- Measures how well the model explains the variance in the target
- 0.8420

Overall: good model

Why:

- TSLA may have clearer patterns, more predictable movements, or more relevant input features.



Reinforcement Learning – APPL

Used PPO (Proximal Policy Optimization) to train an AI agent to make trading decisions based on historical stock data and portfolio state.

RL Environment Highlights:

- Uses 10-day windows of data as input
- Actions: Buy, Sell, Hold
- Includes transaction costs (0.1%)
- Portfolio state (balance + shares held) tracked in each step

Results:

- Trained on AAPL (2020–2024) over 50,000 timesteps
- Final Net Worth: $33,125.62
- Profit: +$23,125.62 from $10,000 initial balance
- Demonstrated adaptive behavior through market ups and downs

Limitations:

- No modeling of slippage, order execution delay, or market impact
- Trained and tested on the same stock, risk of overfitting
- Real-world conditions are more chaotic and data-rich

Figure 5

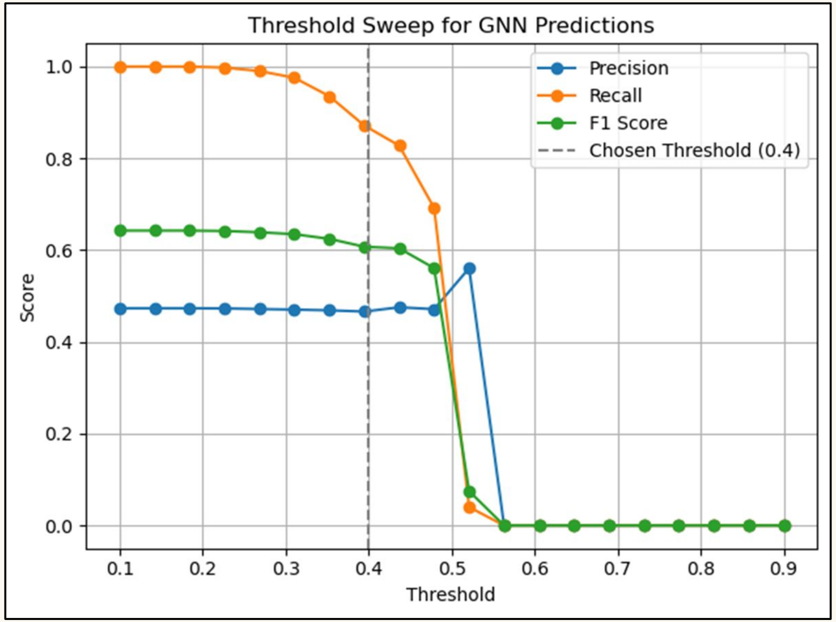
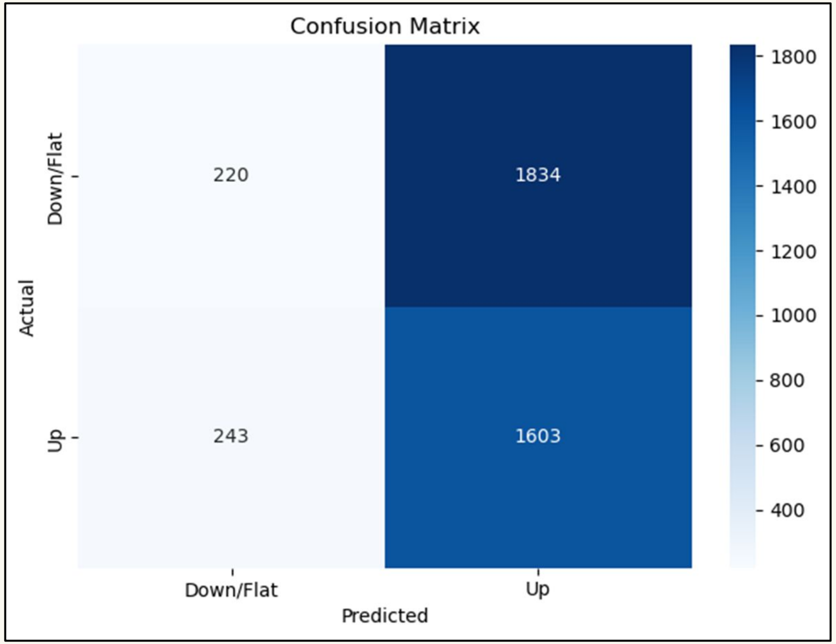


Figure 6



Graph Neural Network

Classified whether stock price will go Up or remain Down/Flat using graph-based relationships between stocks and features

Training:

- 25 epochs, minimal improvement after ~10 epochs
- Final Avg Loss: 0.7470
- Class Distribution: Up = 4.1, Down/Flat = 53.9%

Performance Metrics (Threshold = 0.4):

- Accuracy: 46.7%
- Precision: 46.6%
- Recall: 86.8%
- F1 Score: 60.7%

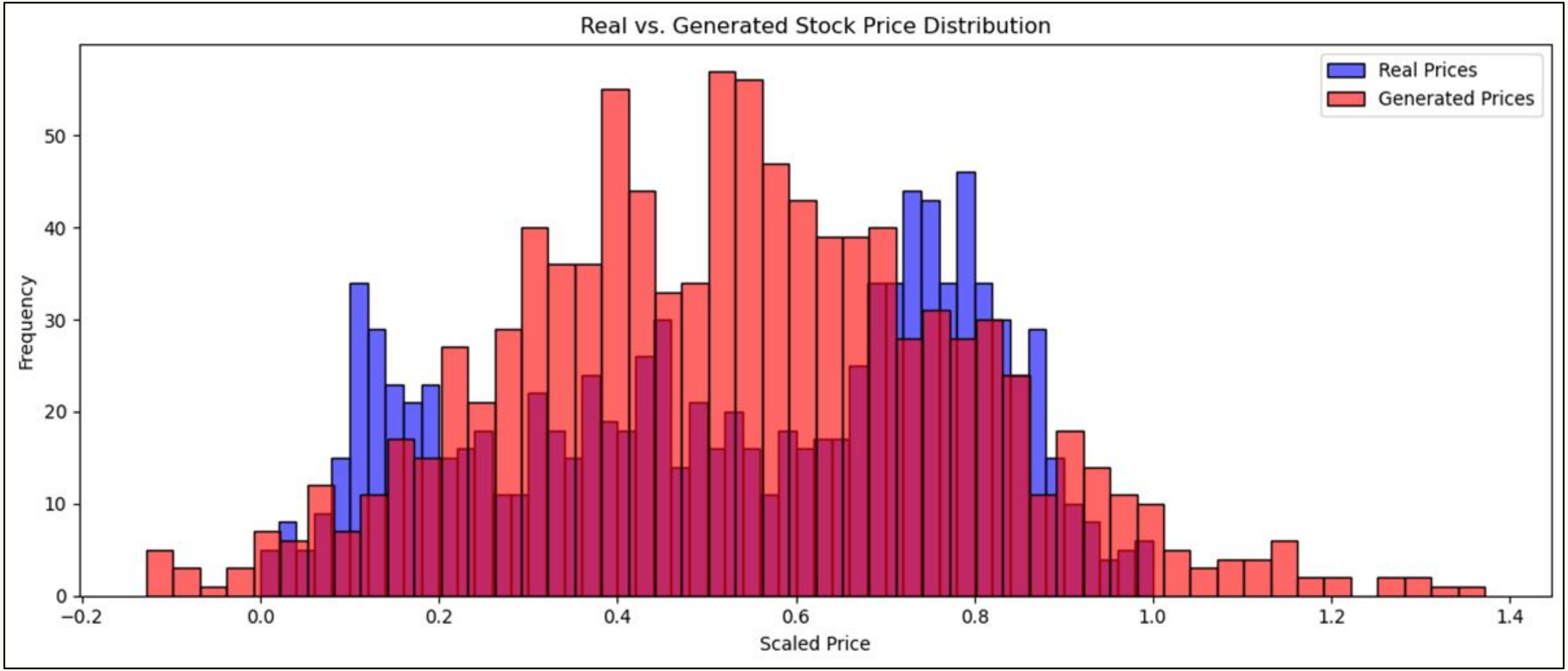
Insights:

- Model is heavily biased toward predicting “Up”
  - likely due to class imbalance and threshold tuning
- High recall suggests the model catches most true positives
  - but it has low precision (many false positives)
- F1 Score peaks around threshold = 0.4, balancing trade-off

```
Class Distribution: Positive=0.4610, Negative=0.5390
Epoch 1/25 - Avg Loss: 0.8230
Epoch 2/25 - Avg Loss: 0.7839
Epoch 3/25 - Avg Loss: 0.7765
Epoch 4/25 - Avg Loss: 0.7649
Epoch 5/25 - Avg Loss: 0.7585
Epoch 6/25 - Avg Loss: 0.7590
Epoch 7/25 - Avg Loss: 0.7501
Epoch 8/25 - Avg Loss: 0.7525
Epoch 9/25 - Avg Loss: 0.7526
Epoch 10/25 - Avg Loss: 0.7475
Epoch 11/25 - Avg Loss: 0.7478
Epoch 12/25 - Avg Loss: 0.7506
Epoch 13/25 - Avg Loss: 0.7494
Epoch 14/25 - Avg Loss: 0.7484
Epoch 15/25 - Avg Loss: 0.7486
Epoch 16/25 - Avg Loss: 0.7479
Epoch 17/25 - Avg Loss: 0.7486
Epoch 18/25 - Avg Loss: 0.7487
Epoch 19/25 - Avg Loss: 0.7473
Epoch 20/25 - Avg Loss: 0.7483
Epoch 21/25 - Avg Loss: 0.7472
Epoch 22/25 - Avg Loss: 0.7479
Epoch 23/25 - Avg Loss: 0.7478
Epoch 24/25 - Avg Loss: 0.7480
Epoch 25/25 - Avg Loss: 0.7470

--- Classification Metrics ---
Accuracy: 0.4674
Precision: 0.4664
Recall: 0.8684
F1 Score: 0.6069
```

Figure 8


Real vs. Generated Stock Price Distribution
Real Prices
Generated Prices
Frequency
Scaled Price
0
10
20
30
40
50
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4

Generative Adversarial Network

The GAN is generating data to compare to real stock prices.

```
Epoch 0 | D Loss: 0.6975, D Acc: 31.25% | G Loss: 0.6954
Epoch 500 | D Loss: 0.6863, D Acc: 52.73% | G Loss: 0.7411
Epoch 1000 | D Loss: 0.6828, D Acc: 52.93% | G Loss: 0.7443
Epoch 1500 | D Loss: 0.6822, D Acc: 52.65% | G Loss: 0.7366
Epoch 2000 | D Loss: 0.6816, D Acc: 52.75% | G Loss: 0.7319
Epoch 2500 | D Loss: 0.6811, D Acc: 53.02% | G Loss: 0.7277
```

Generator:

- Tries to create fake data that looks real
- Starts at 0.6954, rises slightly, then drops to 0.7277

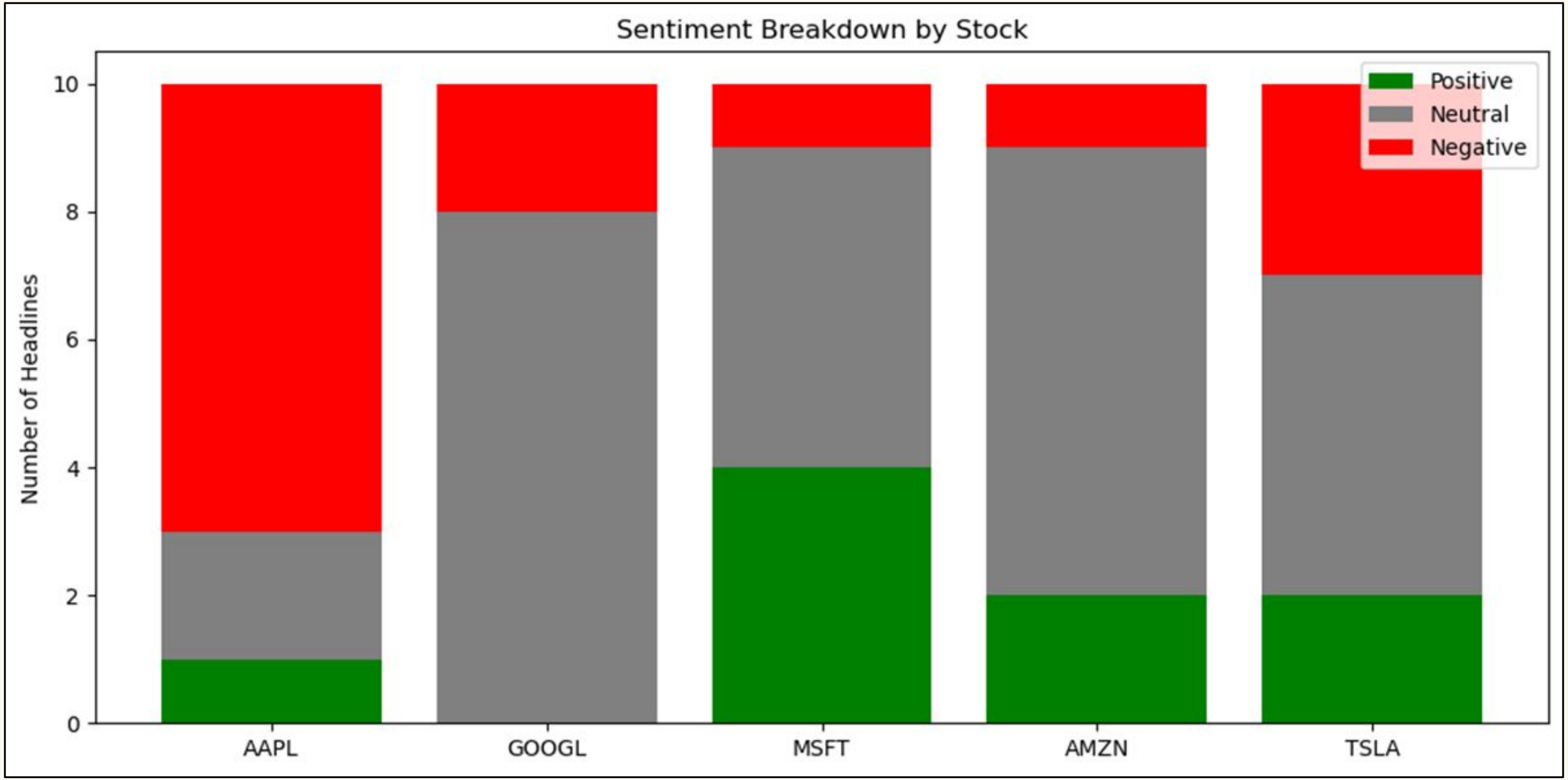
Discriminator:

- Tries to tell real data the from fake data
- **Loss**: measure confidence of predictions
  - Starts at 0.6975 (high), then slowly drops to 0.6811
  - Falling loss means the discriminator is learning -> good
- **Accuracy**: measures frequency of correctness
  - Starts at 31.25%, Climbs to 53.02%
  - Bad start as it is worse than random guessing, but increases to around random guessing
  - Discriminator is getting confused -> shows a balance between the two models

Insights:

- The GAN is successfully confusing the Discriminator (~50% accuracy), which is a good sign in GAN training
- The losses and accuracy stabilizing is a sign that the GAN may be near convergence

Figure 9

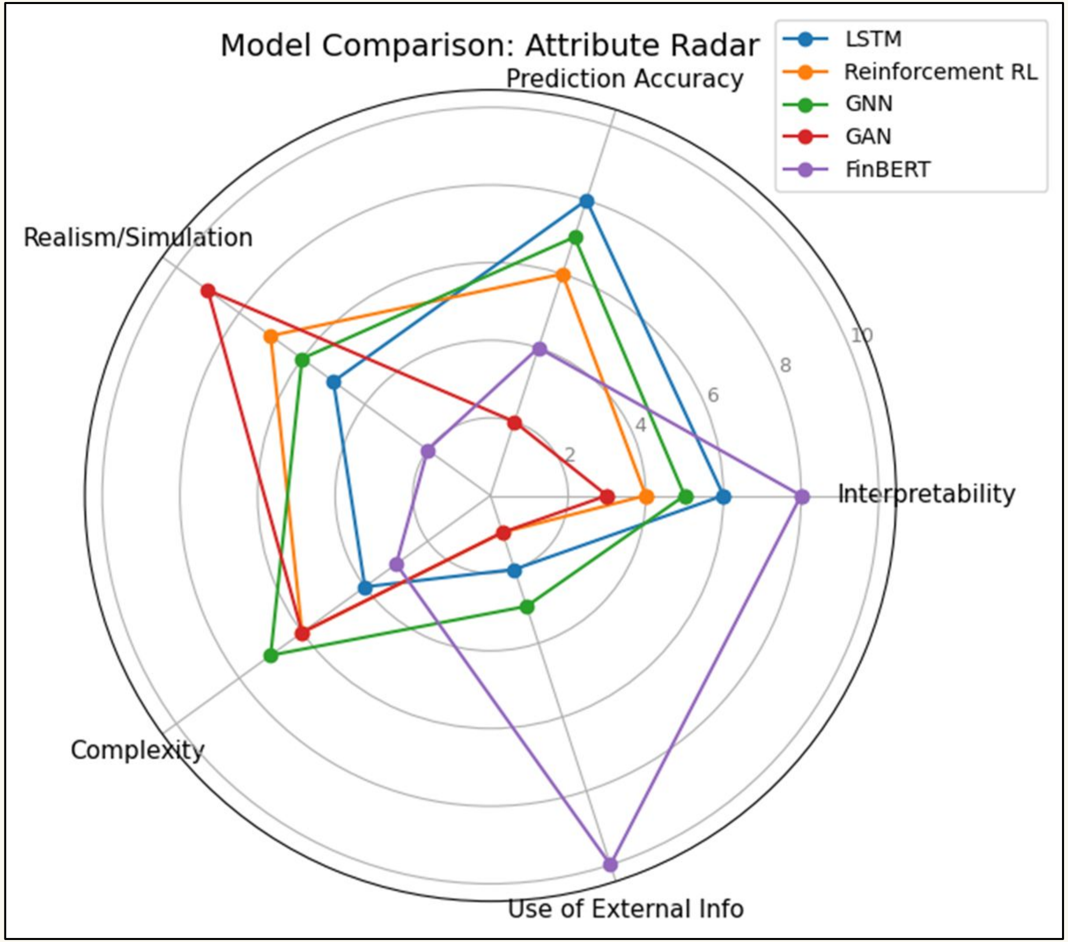


Sentiment Breakdown by Stock
Number of Headlines
0
2
4
6
8
10
AAPL
GOOGL
MSFT
AMZN
TSLA
Positive
Neutral
Negative

Large Language Model– Sentiment Analysis

1. **Apple: -0.60**
   a. **Analysis**: The majority of headlines are negative (e.g., Apple's stock drop, tariff costs, legal disputes).
   b. **Interpretation**: News sentiment around Apple is quite negative, suggesting potential downward pressure on its stock, driven by financial and legal concerns.
2. **Google: -0.20**
   a. **Analysis**: Although there are neutral and some negative headlines (e.g., Reddit stock slide, traffic concerns), no overwhelming negative sentiment.
   b. **Interpretation**: Google is facing some challenges, but the sentiment is not extremely bad. It's more neutral to slightly negative, meaning investors might view it with some caution.
3. **Microsoft: 0.30**
   a. **Analysis**: Most headlines are positive or neutral, especially the high praise for Microsoft's growth (Azure, AI). Positive movements in its stock price are linked to growth sectors.
   b. **Interpretation**: Strong positive sentiment, which suggests confidence in Microsoft's future and growth trajectory, especially in AI and cloud services.
4. **Amazon: 0.10**
   a. **Analysis**: While most headlines are neutral, there are positive mentions related to its strong earnings and stock performance.
   b. **Interpretation**: Amazon has a neutral-to-slightly positive sentiment, indicating stable outlook with mixed opinions, mostly unaffected by major negative news.
5. **Tesla: -0.10**
   a. **Analysis**: TSLA has more neutral headlines, with a few negative ones (e.g., sales collapse in Europe, rivalry issues).
   b. **Interpretation**: Tesla is facing some negative sentiment, but it isn't overwhelming. The stock sentiment is more balanced, with challenges but no major market shift..

Figure 10



Model Comparison: Attribute Radar
LSTM
Reinforcement RL
GNN
GAN
FinBERT
Prediction Accuracy
Interpretability
Use of External Info
Complexity
Realism/Simulation
2
4
6
8
10

Overall Comparison

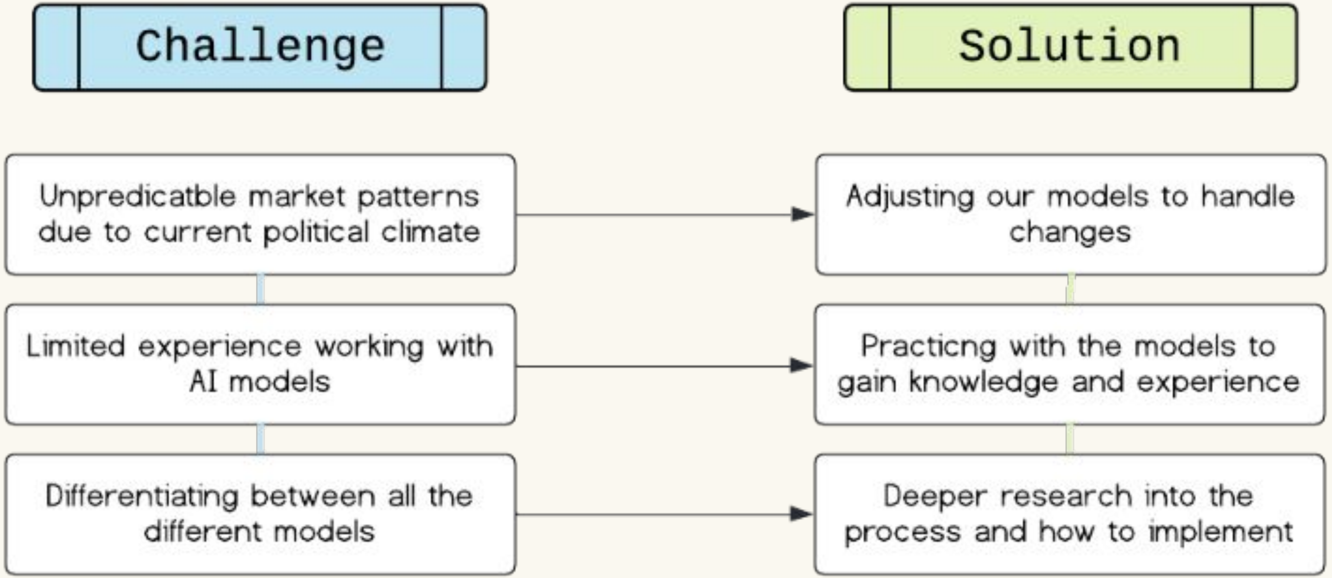
From best -> worst

1. **Interpretability**
   a. LLM -> LSTM -> GNN -> RL -> GAN
2. **Prediction Accuracy**
   a. LSTM -> GNN -> RL -> LLM -> GAN
3. **Realism/Simulation**
   a. GAN -> RL -> GNN -> LSTM-> LLM
4. **Complexity**
   a. GNN-> GAN -> RL -> LSTM -> LLM
5. **Use of External Info**
   a. LLM -> GNN -> LSTM -> RL -> GAN

	Model	Task	Metric	Strengths	Limitations
0	LSTM	Predict future stock price	MAE / MSE / R^2	Captures temporal patterns	Needs clean time series
1	Reinforcement Learning	Optimize portfolio returns	Final Net Worth / P&L	Learns sequential decisions	Sensitive to reward design
2	GNN	Classify next-day direction	Accuracy / F1 Score	Models stock relationships	Hard to tune graph setup
3	GAN	Generate synthetic prices	KL Divergence	Simulates realistic prices	Hard to evaluate quality
4	FinBERT	Analyze news sentiment	Sentiment vs Price Corr.	Handles textual news data	Sentiment ≠ price direction

CHALLENGES

Challenges & Solutions



Citations

Bansal, Devansh. (2025). *Ai in Stock Market: Predicting the Unpredictable in 2025*. damcogroup.com. https://www.damcogroup.com/blogs/ai-in-stock-market-prediction.

Campisi, G., Muzzioli, S., & De Baets, B. (2024). A comparison of machine learning methods for predicting the direction of the US stock market on the basis of volatility indices. International Journal of Forecasting, 40(3), 869–880. https://doi.org/10.1016/j.ijforecast.2023.07.002

Chen, Y., Wei, Z., & Huang, X. (2018). Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18) (pp. 1655–1658). https://doi.org/10.1145/3269206.3269269

Chopra, R., & Sharma, G. D. (2021). Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda. Journal of Risk and Financial Management, 14(11), 526. https://doi.org/10.3390/jrfm14110526

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. https://doi.org/10.1016/j.ejor.2017.11.054

Kirtac, K., & Germano, G. (2024). Sentiment trading with large language models. Finance Research Letters, 62, 105227. https://doi.org/10.1016/j.frl.2024.105227

Citations

Quant Galore. (2023). *The History and Evolution of Quantitative Finance (1980s).* Medium.com. The Financial Journal. https://medium.com/the-financial-journal/the-history-and-evolution-of-quantitative-finance-1980s-2a4eb1f49b52#:~:text=In%20sum%2C%20the%201980s%20were,the%20way%20financial%20markets%20operate

Quantified Trading. (2024). *History of Algorithmic Trading*. quantifiedstrategies.com. https://www.quantifiedstrategies.com/history-of-algorithmic-trading/

Wang, J., & Chen, Z. (2024). Factor-GAN: Enhancing stock price prediction and factor investment with generative adversarial networks. PLoS ONE, 19(6), e0306094. https://doi.org/10.1371/journal.pone.0306094

Yang, H., Liu, X.-Y., Zhong, S., & Walid, A. (2021). Deep reinforcement learning for automated stock trading: An ensemble strategy. In Proceedings of the First ACM International Conference on AI in Finance (pp. 1–8). https://doi.org/10.1145/3383455.3422540

Zhou, L., Zhang, Y., Yu, J., Yongchareon, S., Madanian, S., & Wang, G. (2024). A short survey on graph neural network–based stock market prediction models. In ICIW 2024: The Nineteenth International Conference on Internet and Web Applications and Services (ICIW 2024). (pp. 10–16).

QUESTIONS

Thank You