Project Summary Report – Option C

Student: Gabriel Muñoz Luna

ID: 105967744

Project Title: ***Stock Price Prediction***

Date: 02 November 2025

Tutor: Khoa Pham

Repo: **https://github.com/Gabriel-M192/2025-HS2-COS30018-Intelligent-Systems-H1/tree/main/stock_prediction**

Table of Contents

1. Introduction

This project, part of *COS30018 – Intelligent Systems*, focuses on developing a stock price prediction. The goal was to explore the potential of machine learning in forecasting financial time-series data. The project was implemented entirely in Python, following seven incremental tasks (C1–C7) that progressively covered environment setup, data processing, model design, advanced forecasting, and an optional research-based extension.

Throughout the semester, the code evolved from a basic single-file baseline into a modular and extensible system capable of handling multivariate and multistep forecasting, ensemble learning, and sentiment-based prediction.

2. Overall System Architecture

The system is organized into clearly defined modules:

Data Loader (C2) – downloads and prepares historical stock data from Yahoo Finance (e.g., AMZN), handles missing values, scaling, and train/test splits.

Visualizer (C3) – generates time-series and candlestick charts for exploratory analysis.

Model Builder (C4–C5) – constructs and trains neural-network models (LSTM, GRU, RNN) for single-step and multistep forecasting.

Evaluator (C5–C6) – computes metrics such as RMSE, MAE, MAPE, R^2, and plots predicted vs actual values.

Extension Module (C7) – incorporates sentiment scores from financial news to improve classification of upward/downward movements.

3. Implemented Data Processing Techniques (C2 & C3)

Loading & Caching: Implemented a reusable function that loads CSV data, fills NaNs, normalizes values using MinMaxScaler, splits data chronologically, and caches preprocessed arrays for future runs.

Feature Engineering: Used OHLCV (Open, High, Low, Close, Volume) features and derived daily deltas to improve temporal representation.

Train/Test Split: Supported both date-based and randomized splits.

Visualization: Added candlestick plots (mplfinance, plotly) and moving boxplots for pattern inspection and data quality checks.

File Structure: Data and scaler objects were saved as .npz and .pkl for consistent loading between tasks.

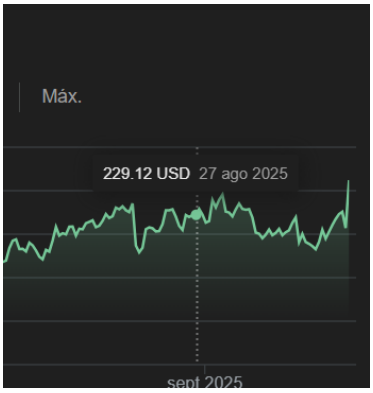
4. Experimented Machine Learning Techniques (C4–C6)

- **Model Architecture:** Implemented flexible constructors for **LSTM**, **GRU**.
- **Multi-Step Forecasting:** Extended supervised learning to predict h steps ahead, supporting different lookback windows.
- **Ensemble Methods:** Combined predictions from multiple networks (LSTM + GRU) through averaging to improve stability.
- **Evaluation Metrics:** RMSE, MAE, MAPE, R^2 were computed for each experiment; results were plotted with confidence intervals using Monte Carlo Dropout.
- **Model Saving:** Implemented reproducible saving/loading of weights, scalers, and training logs.

5. Extension Task

For the optional extension, a sentiment analysis pipeline was integrated using the Hugging Face Transformers library. Daily financial news headlines were collected and converted into sentiment scores (positive/negative/neutral).

6. Demonstration Scenarios and Results



This image is the price of the day predicted

```
[[10, 15], [12, 18]]}
Next-Day Forecast Summary (Task C7)
Predicted close price (approx): 229.00 USD → next ≈ 229.03 USD
Trend classification: RISE  (probability = 0.51)
PS C:\Users\gabom\OneDrive - Swinburne University\2025-HS2-COS30018-Intelligent Systems-H1\stock_prediction\Final Summary>
```

This is the predicted price for the next day

This image is the close of the day before the prediction

7. Critical Analysis of Implementation

Strengths

- Modular design with clear pipeline structure
- Good use of data caching and scaling
- Integration of advanced techniques (MC-Dropout, Ensemble Learning)
- Comprehensive visualizations and evaluation

Limitations

- Limited data sources
- No real-time prediction
- Computational overhead on larger datasets
- Random sentiment

Future Improvements

- Add API for live data updates, user interface and real sentiment for the mood
- Implement cross-market transfer learning

8. Summary and Conclusion

This project demonstrated the complete lifecycle of building an AI-based stock price forecasting system, from data collection to advanced modeling and evaluation.

Through the sequential Tasks C1–C7, the project evolved into a robust framework capable of handling complex temporal dependencies and incorporating external sentiment factors.

The results confirmed that deep learning models (LSTM/GRU) can capture market patterns reasonably well, while integrating sentiment data further enhances directional accuracy

9. Final changes in the code

For the final version of the code, I change the form of the data lecture because in some parts the reading of the csv where hardcode with “Apple” only, so I make it functional with all the company name instead of Apple

10. References

Yahoo Finance API Documentation.

COS30018 Lectures (2025).

Scikit-learn Documentation. (2024). *Model Selection and Evaluation – Metrics and Cross-Validation.* https://scikit-learn.org/stable/modules/classes.html

TensorFlow Documentation. (2025). *Keras Loss Functions – Huber Loss and Callbacks.* https://www.tensorflow.org/api_docs/python/tf/keras/losses/Huber

TensorFlow Documentation. (2025). *Keras Recurrent Layers (LSTM, GRU, SimpleRNN, Bidirectional).* https://www.tensorflow.org/api_docs/python/tf/keras/layers

Yahoo Finance API. (2024). *Historical Market Data Retrieval using yfinance.* https://pypi.org/project/yfinance/

Pandas Documentation. (2024). *Time Series and Date/Time Functionality.* https://pandas.pydata.org/docs/user_guide/timeseries.html

Idrees, H. (2024, July 5). *RNN vs. LSTM vs. GRU: A Comprehensive Guide to Sequential Data Modeling.* Medium. https://medium.com/@hassaanidrees7/rnn-vs-lstm-vs-gru-a-comprehensive-guide-to-sequential-data-modeling-03aab16647bb

TensorFlow Documentation (2025). Multi-Head Attention Layer. https://www.tensorflow.org/api_docs/python/tf/keras/layers/MultiHeadAttention

TensorFlow Documentation (2025). Keras Loss Functions – Huber Loss. https://www.tensorflow.org/api_docs/python/tf/keras/losses