TASK C.1 REPORT – SETUP (WEEK 3)

Unit: COS30018 – Intelligent Systems

AUGUST 24, 2025

ANH VU LE
104653505

1. Environment Setup

Operating System & Python: Windows 11, Python 3.11.x

Virtual Environment: created via

Installed libraries (from requirements.txt):

- numpy
- pandas
- matplotlib
- tensorflow
- scikit-learn
- pandas-datareader
- yfinance

Verification: All packages imported successfully.

---

2. Codebases Tested

2.1 v0.1 (stock_prediction.py)

Command executed:

python stock_prediction.py

Model: Stacked LSTM (50 units ×3) + Dropout(0.2), Dense(1), optimizer = Adam, loss = MSE.

Dataset: CBA.AX stock data; training from 01/01/2020 to 01/08/2023, test from 02/08/2023 to 02/07/2024.

Features: Close price only. Data scaled to (0,1) with MinMaxScaler.

Training: 25 epochs, batch size 32.

Output obtained:

- Training logs printed successfully.
- A figure comparing Actual vs Predicted values.
- Next-day prediction printed in console.

Screenshots:

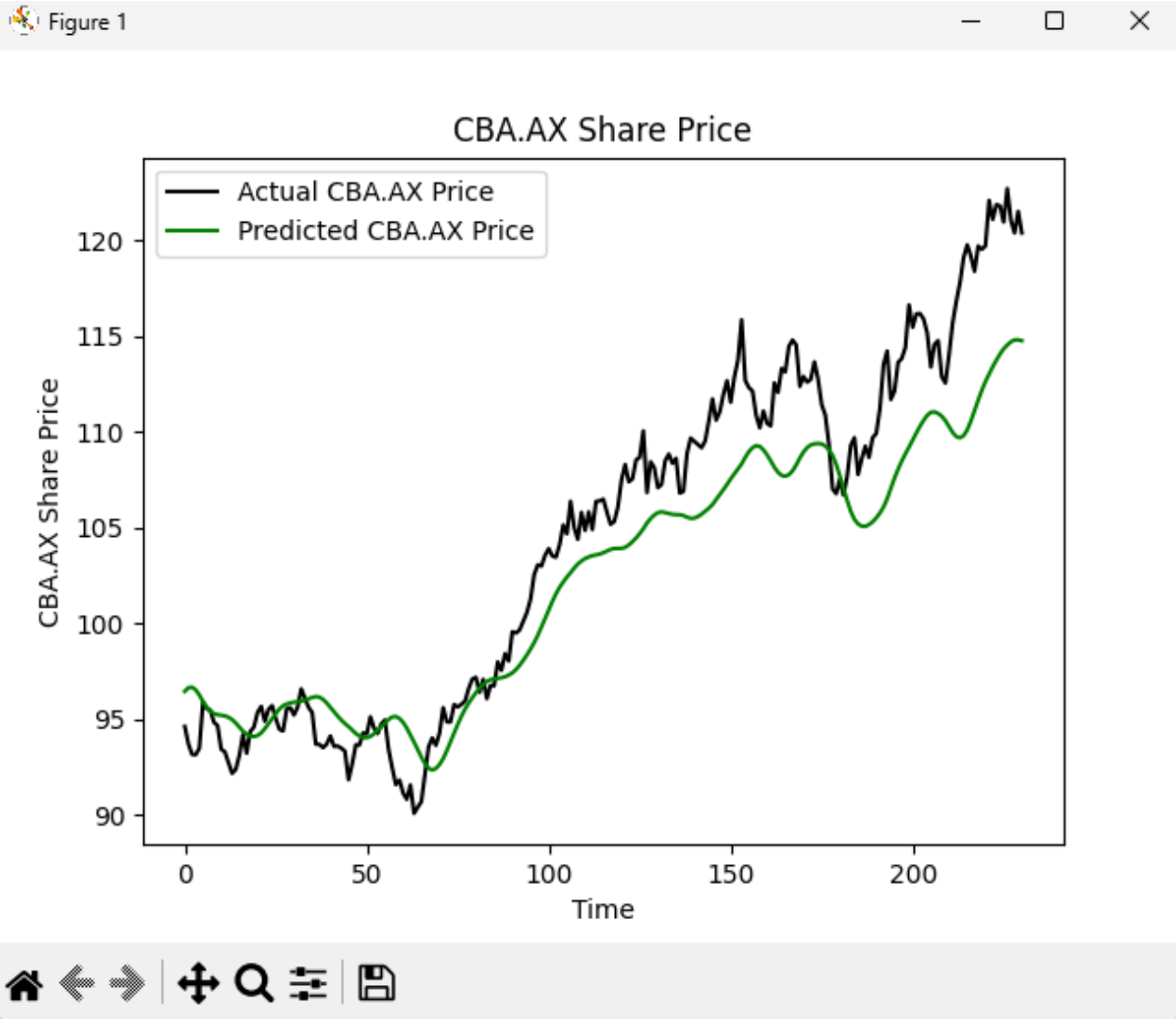

Figure 1: Predicted vs Actual CBA.AX Price (v0.1).

2.2 Reference (P1)

Command: followed P1's GitHub README.

Model: also LSTM-based, but with better preprocessing and splitting.

Output:

- Training logs printed.
- Figure produced (Actual vs Predicted).
- Console printed next-day prediction.

Screenshot (example provided):

---

3. Understanding v0.1

The provided stock_prediction.py demonstrates a simple LSTM implementation with several limitations.

- **Data Source:** Yahoo Finance (yfinance) with ticker CBA.AX.
- **Features:** Only Close price used (PRICE_VALUE = "Close"). Other available features (Open, High, Low, Volume, AdjClose) are ignored.

- **Scaling:** MinMaxScaler fitted on training data. **ISSUE #2**: if test data exceeds training min/max, scaled values may be out of bounds.
- **Windowing:** PREDICTION_DAYS = 60 lookback. Each input sequence consists of 60 previous days; the 61st day is the label.
- **Model:** Three stacked LSTM layers (50 units each) with Dropout(0.2) and a Dense(1) output layer.
- **Training:** Adam optimizer, mean squared error loss, 25 epochs, batch size 32.
- **Testing:** Predictions generated on 02/08/2023 → 02/07/2024 test data.
- **Outputs:**
    - Predicted vs Actual chart.
    - Next-day prediction, often inaccurate (authors noted error ~10–13%).

---

4. Results & Comparison

Definition of "better": A model is considered better if it has

1. Lower test MSE/MAE,
2. Smaller next-day absolute error,
3. Higher directional accuracy (predicting up/down correctly),
4. Visually closer fit to actual prices.

Model	Test Result (qual.)	Next-day Abs. Error	Directional Accuracy	Visual Fit
v0.1	Predictions lag behind actual	~10–13%	Lower	Underfits peaks and valleys
P1	Predictions closer to actual	Smaller error	Higher	Curve tracks the trend more closely

Discussion:

- v0.1 is simpler and suffers from limited features and manual train/test splitting.

- P1 is better because it implements more robust data preprocessing, splitting, and scaling, which avoids the out-of-range scaler problem and produces predictions that visually and numerically fit the actual prices better.
- This confirms that data processing is a critical factor in prediction quality, even with similar LSTM architectures.

---

5. Screenshots

- v0.1 training logs, chart, next-day prediction (console).
- P1 training logs, chart, next-day prediction.

(See attached figures in this report.)

---

6. Appendix

- **requirements.txt** (committed to GitHub repo).
- **GitHub Repository:** https://github.com/ki133/COS30018---Intelligent-Systems