FACULTY OF ENGINEERING AND TECHNOLOGY

NAME	NUR HIDAYAH BINTI ABD RAHMAN

Introduction

Stock market forecasting is a challenging task due to volatility, non-linear price movements, and the influence of external factors. Machine learning offers an opportunity to model these patterns more effectively compared to traditional statistical methods, particularly for time-series data.

NVDA was selected due to its prominence in the technology and semiconductor industry, where stock movements often reflect market momentum. Weekly data was used to reduce market noise and capture medium-term price trends. A Long Short-Term Memory (LSTM) model was chosen for its ability to learn temporal dependencies. Technical indicators such as MACD, RSI, and moving averages were incorporated to enhance the model's ability to detect momentum and trend patterns. This paper follows a data-mining process, covering data understanding, preparation, modelling, and evaluation of model performance.

1. Problem Formulation

1.1 Problem Description

This paper applies a data-driven approach to predict the directional movement of NVIDIA Corporation's (NVDA) stock price for the following week. The dataset includes technical indicators that capture price momentum and trend strength which are key elements in technical analysis that often reflect market psychology. Predicting price direction rather than exact values aligns with how traders commonly approach risk management and timing decisions.

The data was collected from the **Alpaca Markets API**, which provides access to historical stock prices and volume data. Indicators such as **MACD**, **RSI**, and various **Moving Averages (SMA)** were calculated to reflect trend continuation or reversal tendencies. These variables form the basis for supervised learning where the **target variable** represents whether the next week's closing price increases (1) or decreases (0) and prediction is framed as a binary classification problem.

1.2 Justification for Using a Data Mining Approach

Traditional statistical models often assume linear relationships or independence between variables, assumptions that are rarely valid in financial contexts. In contrast, data mining methods, especially those involving neural networks allow for the discovery of hidden structures and temporal correlations.

The **LSTM network** is appropriate because it accounts for sequential dependencies and long-term patterns, which are critical in understanding how past market movements influence future prices. By combining this approach with engineered features derived from technical analysis, the model integrates financial domain knowledge with machine learning's predictive capabilities. The aim is not to eliminate uncertainty but to quantify it, transforming historical fluctuations into structured insights.

2. Problem and Data Understanding

2.1 Problem Understanding

Financial markets are influenced by shifting economic conditions, sentiment, and behavioural factors, making short-term forecasting difficult. This paper aims to explore whether historical pricing patterns and technical indicators contain enough information for an LSTM model to anticipate NVDA's weekly price direction. The focus is on evaluating the feasibility of using a machine-learning approach for directional forecasting rather than exact price prediction. This problem is relevant for data science as it tests the limits of modelling highly volatile, non-stationary time-series data, where patterns may not consistently repeat. It also provides insight into whether technical indicators alone are sufficient for predictive modelling in a dynamic market.

2.2 Data Understanding

The dataset was collected using the **yfinance library** and consists of weekly historical OHLCV data for NVDA. Key considerations include:

Dataset Characteristics

- Weekly OHLCV values spanning multiple years up to Oct 2025.
- Weekly frequency reduces noise and aligns with medium-term forecasting.
- Auto-adjusted data avoids manual split/dividend corrections.

Target Variable

- Binary label:

  1 = next week close > current week close

  0 = next week close < current week close

Features Used

- **Price-based:** OHLCV values.
- **Technical indicators:**
  - SMA20, SMA50, SMA200 (short-, medium-, long-term trends)
  - MACD, Signal Line, Histogram (momentum strength & direction)
  - RSI (overbought/oversold signals)

Data Quality Considerations

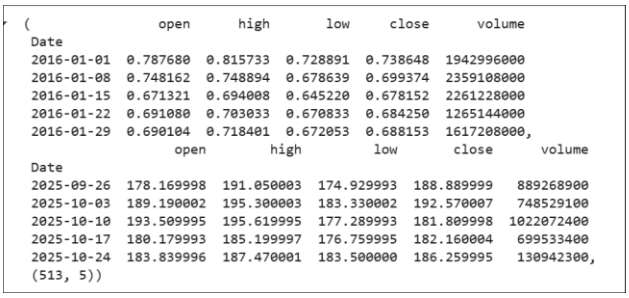
- Missing rows removed to maintain continuity.
- Outliers retained to preserve realistic price behaviour.

3. Data Preparation

This stage focuses on collecting the data required for the model, preparing it for analysis, and engineering relevant features that can help the LSTM learn meaningful patterns from historical stock movements.

3.1 Data Collection

Weekly NVDA stock price data was downloaded using the yfinance library in Google Colab. The dataset includes the weekly open, high, low, close and volume (OHLCV) values, spanning multiple years and extending up to October 2025. Working with weekly data helps reduce short-term noise and aligns with the objective of predicting the next week's price direction. Figure 1 shows the first and last few rows of the weekly dataset after loading and cleaning, as well as the total number of observations available for modelling.



```
(                open      high       low     close      volume
 Date
 2016-01-01  0.787680  0.815733  0.728891  0.738648  1942996000
 2016-01-08  0.748162  0.748894  0.678639  0.699374  2359108000
 2016-01-15  0.671321  0.694008  0.645220  0.678152  2261228000
 2016-01-22  0.691080  0.703033  0.670833  0.684250  1265144000
 2016-01-29  0.690104  0.718401  0.672053  0.688153  1617208000,
                   open        high         low       close      volume
 Date
 2025-09-26  178.169998  191.050003  174.929993  188.889999   889268900
 2025-10-03  189.190002  195.300003  183.330002  192.570007   748529100
 2025-10-10  193.509995  195.619995  177.289993  181.809998  1022072400
 2025-10-17  180.179993  185.199997  176.759995  182.160004   699533400
 2025-10-24  183.839996  187.470001  183.500000  186.259995   130942300,
 (513, 5))
```

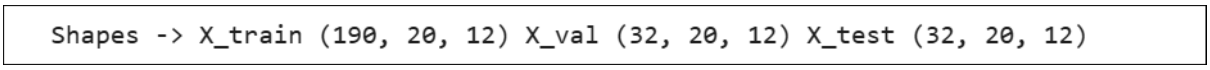
Figure 1: Weekly NVDA OHLCV data after loading and initial cleaning

3.2 Data Cleaning and Pre-processing

Before modelling, the dataset was checked for missing values and inconsistent rows, and any incomplete observations were removed to maintain a continuous weekly sequence. Since the data already comes auto adjusted from yfinance, stock splits and dividends did not require manual adjustment.

The data was then split chronologically into training, validation, and testing sets to reflect a realistic forward-looking prediction scenario. The most recent 52 weeks were reserved as the test set, while the 52 weeks prior to that were used for validation. The remaining earlier observations formed the training set. This approach ensures that the model is trained on past data and evaluated on future unseen data, which mimics real-world stock forecasting.

To prepare the data for the LSTM model, MinMax scaling was applied only to the training set, and the same scaler was used to transform the validation and test sets. This prevents information leakage from future data into the model. Finally, the data was converted into sequences, where each input sample contains 20 weeks of historical data used to predict the following week's direction. Figure 2 displays the resulting shapes of the training, validation and test sequences.



```
Shapes -> X_train (190, 20, 12) X_val (32, 20, 12) X_test (32, 20, 12)
```

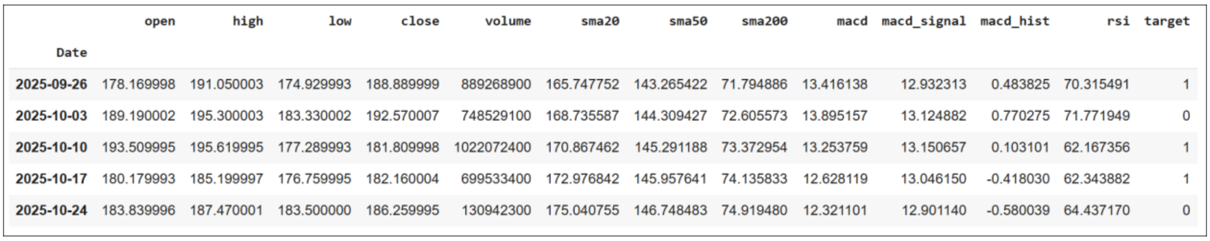
Figure 2: Shapes of X_train, X_val and X_test after scaling and sequence creation

3.3 Feature Engineering

Technical indicators were added to provide the model with additional trend and momentum context beyond raw price data. These indicators are commonly used in trading to interpret market direction and potential reversals:

- **Simple Moving Averages (SMA20, SMA50, SMA200):** Capture short-, medium-, and long-term trend behaviour to help the model understand different market horizons.
- **MACD, Signal Line and Histogram:** Indicate momentum shifts and crossover signals, which may reflect changes in buying or selling pressure.
- **Relative Strength Index (RSI):** Highlights possible overbought or oversold conditions that could lead to trend reversals.

After feature creation, initial rows that lacked sufficient values for longer indicators were removed. A binary target variable was then generated to label whether the following week's closing price increased or decreased.



	open	high	low	close	volume	sma20	sma50	sma200	macd	macd_signal	macd_hist	rsi	target
Date													
2025-09-26	178.169998	191.050003	174.929993	188.889999	889268900	165.747752	143.265422	71.794886	13.416138	12.932313	0.483825	70.315491	1
2025-10-03	189.190002	195.300003	183.330002	192.570007	748529100	168.735587	144.309427	72.605573	13.895157	13.124882	0.770275	71.771949	0
2025-10-10	193.509995	195.619995	177.289993	181.809998	1022072400	170.867462	145.291188	73.372954	13.253759	13.150657	0.103101	62.167356	1
2025-10-17	180.179993	185.199997	176.759995	182.160004	699533400	172.976842	145.957641	74.135833	12.628119	13.046150	-0.418030	62.343882	1
2025-10-24	183.839996	187.470001	183.500000	186.259995	130942300	175.040755	146.748483	74.919480	12.321101	12.901140	-0.580039	64.437170	0

Figure 3: Engineered dataset showing added MACD, RSI, SMA and target variable

4. Data Modelling

4.1 Model Selection

Among various supervised learning algorithms, the **LSTM neural network** was chosen because of its proven capacity to model time series data. Traditional models like Decision Trees or Logistic Regression cannot adequately capture temporal dependencies, whereas LSTM architectures are designed to retain relevant past information over long sequences. In financial data, this is particularly valuable, market patterns often depend not only on recent behavior but also on longer historical trends.

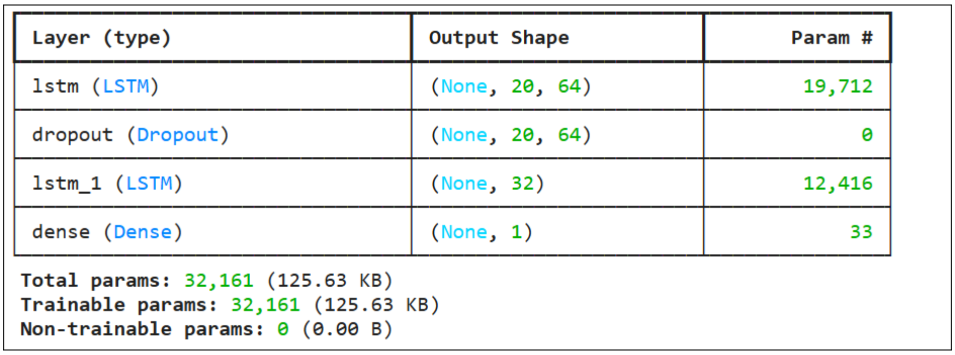
The model architecture comprised two LSTM layers followed by a dense layer with a sigmoid activation function to output probabilities for binary classification. Dropout regularization was used to prevent overfitting. The network was optimized using the Adam optimizer and trained over 50 epochs with a batch size of 16.

4.2 Implementation Overview

A LSTM neural network was implemented to model the sequential nature of stock price movements. The model was built in Google Colab using TensorFlow and Keras. The LSTM architecture was selected because it is designed to learn temporal patterns and retain information across time steps, which aligns with the objective of predicting weekly directional movement.

The model consisted of two stacked LSTM layers with 64 and 32 units. The first LSTM layer used **return_sequences=True** to allow the second LSTM layer to process the full hidden state sequence. A dropout layer with a rate of 0.2 was applied to reduce overfitting by preventing the model from relying too heavily on specific neurons during training. A final Dense layer with a sigmoid activation function was used to output a probability between 0 and 1 for binary classification (up or down next week). The model was compiled using the Adam optimizer and binary cross-entropy loss, which is suitable for two-class classification tasks.

Figure 4 shows the model summary including the layer configuration and total trainable parameters.



```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ lstm (LSTM)                     │ (None, 20, 64)         │        19,712 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout (Dropout)               │ (None, 20, 64)         │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ lstm_1 (LSTM)                   │ (None, 32)             │        12,416 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense (Dense)                   │ (None, 1)              │            33 │
└─────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 32,161 (125.63 KB)
 Trainable params: 32,161 (125.63 KB)
 Non-trainable params: 0 (0.00 B)
```

Figure 4: LSTM model architecture and parameter summary

4.3 Model Training

The model was trained using the prepared training sequences, with the validation set used to monitor performance during training. Early stopping was applied with a patience of 10 epochs to prevent overfitting by stopping training once the validation loss stopped improving. A model checkpoint callback was also used to save the best-performing model weights during training.

The model was trained for up to 50 epochs with a batch size of 16. The training and validation accuracy and loss over the epochs are presented in Figure 5. From the plots, training accuracy increased steadily, while validation accuracy fluctuated, indicating that the model learned patterns from the training data but had limited generalisation to unseen data. Figure 5 compares the training and validation accuracy and loss over the epochs.

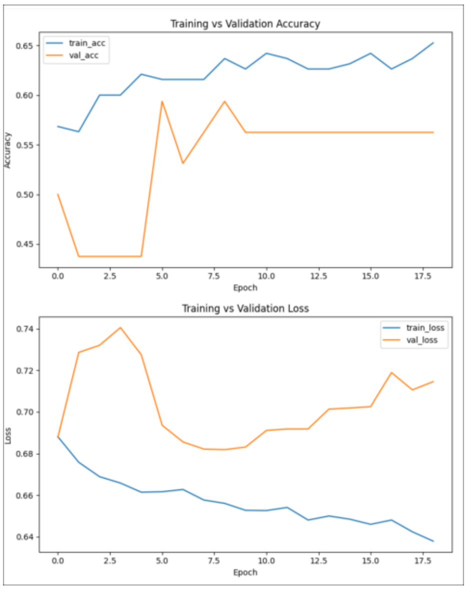


Figure 5: Training and Validation Accuracy and Loss

After training, the model was evaluated on the test set to assess generalisation to unseen data. The confusion matrix and classification report are shown in Figure 6, while the Receiver Operating Characteristic (ROC) curve is presented in Figure 7.

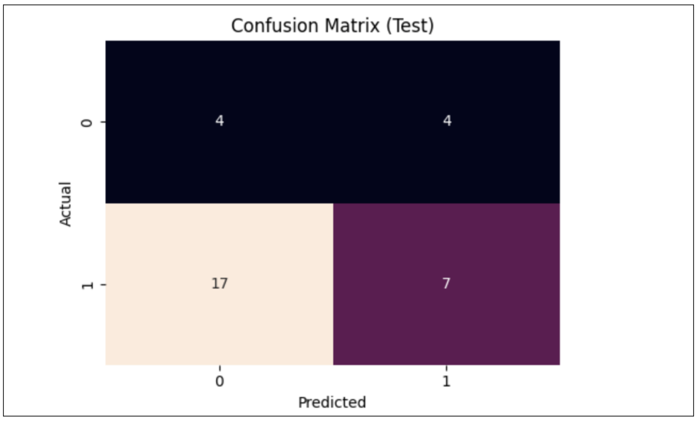


Figure 6: Confusion Matrix and Classification Report

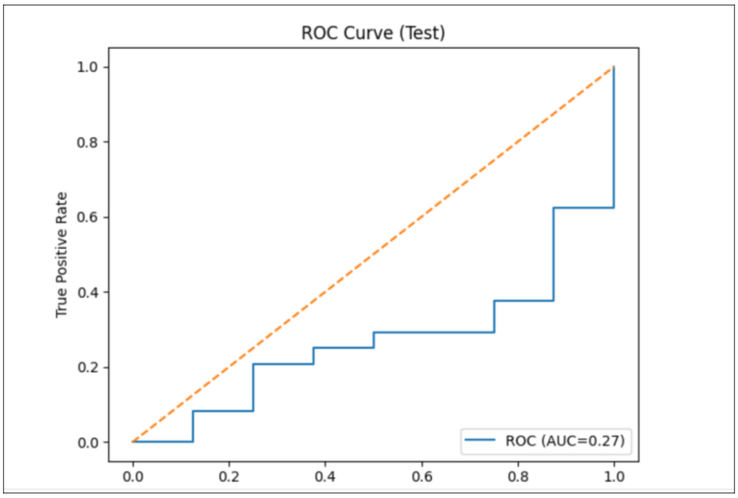
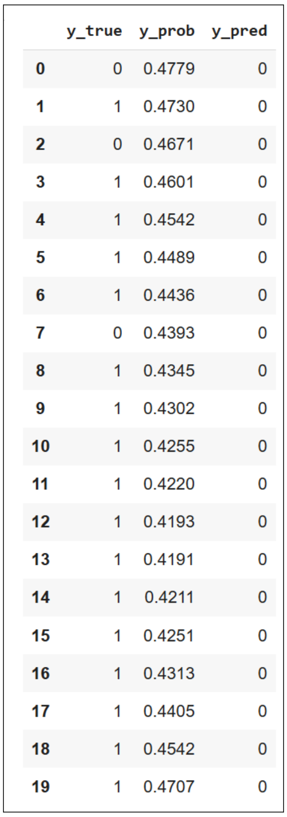


Figure 7: ROC Curve

To provide further insight into the model's predictions, Figure 8 illustrates a sample of predicted probabilities alongside the actual outcomes. The model's predicted probabilities were mostly concentrated around 0.40–0.48, showing low confidence in directional prediction.



	y_true	y_prob	y_pred
0	0	0.4779	0
1	1	0.4730	0
2	0	0.4671	0
3	1	0.4601	0
4	1	0.4542	0
5	1	0.4489	0
6	1	0.4436	0
7	0	0.4393	0
8	1	0.4345	0
9	1	0.4302	0
10	1	0.4255	0
11	1	0.4220	0
12	1	0.4193	0
13	1	0.4191	0
14	1	0.4211	0
15	1	0.4251	0
16	1	0.4313	0
17	1	0.4405	0
18	1	0.4542	0
19	1	0.4707	0

Figure 8: Sample of Model Predictions vs Actual Outcomes

5. Model Evaluation and Results

The model’s performance was evaluated using the test set to assess its ability to generalise to unseen data. The LSTM achieved a testing accuracy of 34.38%, indicating weak predictive performance for weekly price direction. As shown in Figure 6, the confusion matrix and classification report reveal that the model correctly identified only 7 out of 24 upward weeks and 4 out of 8 downward weeks, showing difficulty in distinguishing between the two classes.

The ROC curve in Figure 7 further highlights this limitation, with an AUC score of 0.27, below the random benchmark of 0.50. This suggests that the patterns learned during training did not generalise and may have captured noise rather than meaningful signals. This outcome aligns with the nature of stock market data, where weekly price movements are highly volatile and influenced by unpredictable external factors that are not reflected in historical or technical indicators alone.

As shown in Figure 5, the training accuracy increased steadily, while validation accuracy remained inconsistent and lower, indicating mild overfitting. The widening gap between training and validation loss suggests that the model learned patterns specific to the training data but did not transfer well to new data. Figure 8 presents a sample of prediction outputs, which mostly fell near the 0.50 probability threshold, indicating low confidence in classification. This reinforces the model’s limited predictive capability for weekly directional forecasting.

6. Interpretation and Discussion

The evaluation results show that LSTM struggled to generalise patterns from historical data to unseen weekly price movements. While the model was able to learn from training data, its lower performance on validation and test sets suggests limited effectiveness in real-world forecasting. This reflects the inherent difficulty of modelling financial time series, as markets are influenced by sentiment, news events, macroeconomic conditions, and behavioural factors that technical indicators alone may not capture.

The model’s weak predictive power and low AUC score indicate that weekly directional changes in NVDA’s price did not follow consistent patterns that the LSTM could learn effectively. Although technical indicators provided trend and momentum context, they may be insufficient for short-term forecasting. This highlights the limitations of using only market-derived features without incorporating external data sources.

Despite the model’s performance, the report provided meaningful insights into the challenges of financial modelling using machine learning. It emphasised the importance of selecting appropriate features, addressing overfitting, and setting realistic expectations when applying deep learning to noisy, non-stationary data. The results also suggest value in exploring additional feature types and more advanced architecture for future research.

7. Conclusion

In conclusion, this paper successfully applied a Long Short-Term Memory (LSTM) model to predict the weekly directional movement of NVIDIA’s stock price using historical data and technical indicators. Although the model did not achieve strong predictive accuracy on unseen data, the implementation process provided meaningful learning experiences in data preparation, feature engineering, sequence modelling and neural network evaluation. The results highlight the limitations of relying solely on technical indicators for weekly stock direction prediction, especially in a highly volatile and dynamic market environment.

Despite the model’s performance, it helped build a deeper understanding of time series forecasting using machine learning, and it demonstrated the importance of evaluating models realistically rather than solely focusing on accuracy. Future improvements could involve experimenting with testing more advanced architectures like GRUs, Transformers or hybrid models. Kanungo (2025) found that deep learning models such as LSTM, GRU, and Transformers outperform traditional models like ARIMA across multiple prediction horizons in stock forecasting. Overall, the paper provided a valuable foundation for further exploration into developing more robust predictive models for financial forecasting.

Reference

Kanungo, P. (2025). Time series forecasting in financial markets using deep learning models. World Journal of Advanced Engineering Technology and Sciences, 15(1), 709–719. https://doi.org/10.30574/wjaets.2025.15.1.0167

Word count (excluding reference, figures & figures caption): 2150