Project Report

Stock Price Prediction

Nupur

Student Number: 1340285

Course Title: TensorFlow and Keras with Python

Fanshawe College

Abstract

This project is for stock price prediction using both regression and classification approaches.The process commences with data acquisition and feature engineering, including the derivation of various technical indicators. Deep learning architectures, specifically RNN, LSTM, BiLSTM and GRU, are employed for regression tasks to forecast future closing prices, while classification models are applied to predict the direction of price movement. Model performance and robustness are assessed using cross-validation techniques and relevant evaluation metrics. The findings demonstrate that the use of both regression and classification approaches, reinforced by visualizations and error analysis, improves the accuracy and reliability of predictive models in financial time series forecasting.

1 Introduction

Stock price prediction is a challenging and critical task in financial markets, with significant implications for investors and traders. Accurate forecasts of stock prices and their movements can lead to better decision-making, risk management, and improved portfolio performance. Traditional methods often rely on statistical techniques or technical indicators; however, advances in deep learning have opened new avenues for modeling complex temporal patterns in financial data.

In this project, both regression and classification approaches were used for stock price prediction and stock price classification. The regression component focuses on forecasting the actual closing prices for the next day, while the classification part predicts the direction of price movement—whether the stock price will go up or down. By integrating these two perspectives, the project seeks to leverage the strengths of both approaches.

To achieve this, the project involves data acquisition and preprocessing, including the calculation of relevant technical indicators. Deep learning architectures such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Bidirectional LSTM and Gated Recurrent Units (GRU) are applied to capture sequential dependencies in the data. The models are evaluated using cross-validation and various performance metrics to ensure their effectiveness. The combined strategy aims to provide a comprehensive predictive framework applicable to real-world financial time series forecasting.

2 Dataset and Preprocessing

2.1 Data Acquisition

The dataset used for this project consists of historical stock price data for Apple Inc. (AAPL), retrieved using the `yfinance` Python library. The data spans from **January 1, 2010 to June 30, 2025**, providing a comprehensive time range that captures various market phases, including bull and bear markets, economic events, and recent AI-driven growth. The dataset includes standard financial metrics such as Open, High, Low, Close, Adjusted Close, and Volume.

2.2 Feature Engineering

To enhance the predictive power of the dataset, extensive feature engineering was performed by computing a range of technical indicators. These features capture multiple dimensions of stock market behavior, including trends, momentum, volatility, and volume-based patterns. The following indicators were added:

- **Trend Indicators:**
  20-period Moving Average (MA20), 12-period Exponential Moving Average (EMA12), Moving Average Convergence Divergence (MACD)

- **Momentum Indicators:**
  Relative Strength Index (RSI), Stochastic Oscillator (%K), Commodity Channel Index (CCI), Williams %R

- **Volatility Indicators:**
  20-period Rolling Standard Deviation (Volatility), Average True Range (ATR), Bollinger Bands (Upper, Lower)

- **Volume-Based Indicators:**
  Volume Ratio, On-Balance Volume (OBV)

These indicators were computed using rolling window techniques and formulas from technical analysis, enriching the dataset with useful signals for learning market dynamics.

2.3 Normalization

To prepare the features for deep learning models, standard scaling was applied, transforming the data to have zero mean and unit variance. This normalization helps ensure faster convergence and improved model performance, particularly in neural networks sensitive to feature scaling.

2.4 Sequence Preparation

The data was structured into sequences using a sliding window approach. Each sequence consists of a fixed number of time steps, 20 days for this project forming the input for RNN-based models. The corresponding target values include:

- **Regression:** To predict the next day's closing price.

- **Classification:** To predict the direction of the stock's price movement. The model attempts to determine whether the mean closing price over the next few days (referred to as the *future horizon*) will increase or decrease compared to the mean closing price over the preceding few days (referred to as the *past horizon*).

2.5 Train/Test Split and Cross-Validation

The dataset was split into training and testing sets(with test size being 20 percent of total dataset), maintaining the temporal order to respect the time series structure of stock prices. During model development, K-fold cross-validation was applied to the training data to ensure robustness and reduce overfitting. In conjunction with this, hyperparameter tuning was performed to identify the best-performing model by selecting the optimal combination of parameters such as learning rate, batch size, number of epochs, and sequence length. This combined approach allows for a reliable evaluation of model accuracy while preserving the sequential nature of financial data.

These preprocessing steps ensured the dataset was clean, information-rich, and well-suited for training advanced deep learning models aimed at forecasting stock prices and predicting price movement directions.

3 Methodology

This project addresses two tasks: regression (predicting next-day closing price) and classification (predicting price direction).

3.1 Regression Methodology

3.1.1 Model Selection

A baseline RNN model was first implemented without tuning. A model selection pipeline was then applied to compare RNN-based architectures — LSTM, BiLSTM, GRU — using grid search and K-fold cross-validation. GRU was selected as the best performer based on predictive accuracy.

3.1.2 Hyperparameter Tuning & Cross-Validation

Grid search was used to tune architecture type, hidden units, dropout rate, learning rate, batch size, batch normalization, regularization, and loss type (MSE, Huber). K-fold cross-validation ensured robust evaluation, and average MSE/Huber loss across folds were used to select the best configuration.

Grid search with 3-fold cross-validation was used to optimize hyperparameters and find the best model;the best regression model came out to be GRU with the following parameters:

Parameter	Value
sequence_length	20
test_size	0.2
epochs	100
batch_size	32
learning_rate	0.001
hidden_units	[32, 16]
dropout_rate	0.2
model_type	GRU
loss_type	huber
use_regularization	True

Table 1: Best hyperparameters for the regression GRU model.

3.1.3 Training Strategy

The GRU model was trained on sequential data using the Adam optimizer with a batch size of 32 and Early stopping and validation monitoring were applied to prevent overfitting, with training and validation losses tracked to evaluate convergence.

3.1.4 Challenges Faced During Training

Stock price regression posed challenges due to the noisy and non-stationary nature of financial time series. To address this, extensive preprocessing was required, including normalization, sequencing into fixed time steps, and detrending key features to improve stationarity. These steps helped the GRU model learn meaningful short-term patterns while mitigating the effects of long-term trends and volatility.

3.1.5 Loss Function

The MSE loss has been used for base RNN model while, Huber loss function has been used for regression tasks to balance sensitivity to outliers and smooth training.

3.1.6 Evaluation Metrics

Performance was evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE).

This comprehensive methodology ensures that the regression model is carefully selected, effectively trained, and rigorously validated, enabling accurate and interpretable stock price predictions.

3.2 Classification Methodology

3.2.1 Model Selection

A baseline RNN model was initially implemented. Subsequently, architectures including RNN, LSTM, BiLSTM, and GRU were compared via grid search and K-fold cross-validation. The BiLSTM model was selected as the best-performing architecture based on classification metrics.

3.2.2 Hyperparameter Tuning & Cross-Validation

Grid search was used to tune architecture type, hidden units, dropout rate, learning rate, batch size, past horizon(previous days average value used for predicton),furture horizon (future days average value to be predicted) etc. using 3-fold cross-validation ensuring robust evaluation.

Parameter	**Value**
Architecture	BiLSTM
Hidden Units	[64, 32]
Dropout Rate	0.3
Batch Size	32
Learning Rate	0.001
Sequence Length	20
Past Horizon	30
Future Horizon	30

Table 2: Best hyperparameters for the classification BiLSTM model.

3.2.3 Training Strategy

The BiLSTM model was trained on sequential data using the Adam optimizer with a batch size of 32 and 100 epochs with Early stopping and validation monitoring were applied to prevent overfitting, with training and validation losses tracked to evaluate convergence.

3.2.4 Challenges Faced During Training

Training the classification model was challenging due to the noisy, volatile, and non-stationary nature of stock prices. To address this, the data underwent normalization, detrending(making data stationary by using differencing), and sequencing into fixed time steps. Additionally, defining meaningful past and future horizons was critical for generating reliable target labels and ensuring that the model learned accurate directional patterns rather than reacting to random fluctuations.

3.2.5 Loss Function

In this project, focal loss with weighted loss in binary cross entropy was used to improve classification performance in the presence of class imbalance. This choice was made because the stock prices in the data set exhibited prolonged trends, such as consecutive days of upward or downward movement. As a result, the data showed an imbalanced class distribution, where certain outcomes (e.g., price increases) occurred more frequently. Focal Loss helped the model focus on learning from the minority classes by reducing the contribution of well-classified examples and emphasizing harder, misclassified ones.

3.2.6 Evaluation Metrics

Model performance was assessed using accuracy, precision, recall, F1 score, and confusion matrix to provide a comprehensive evaluation of classification effectiveness.

4 Evaluation Metrics and Results

4.1 Regression

Metrics Comparison for base and selected model(GRU):

Model	MSE	MAE	Std. Dev. of Predictions
Base RNN	18.3124	2.8635	7.2671
Best GRU	15.0929	2.4782	7.9897

Table 3: Comparison of regression performance metrics between the base RNN and best GRU models.

Loss curve Comparison for base and selected model(GRU):

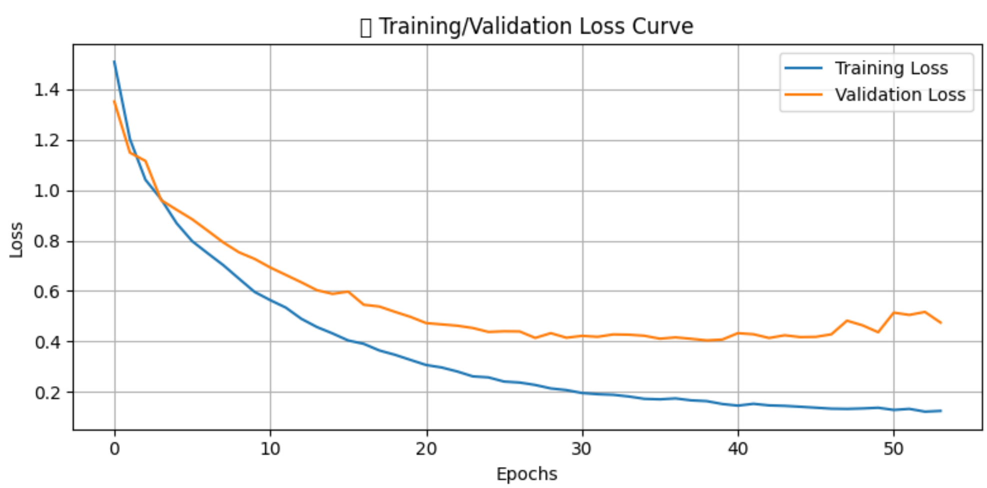

Figure 1: Training and Validation MSE Loss over Epochs base model RNN

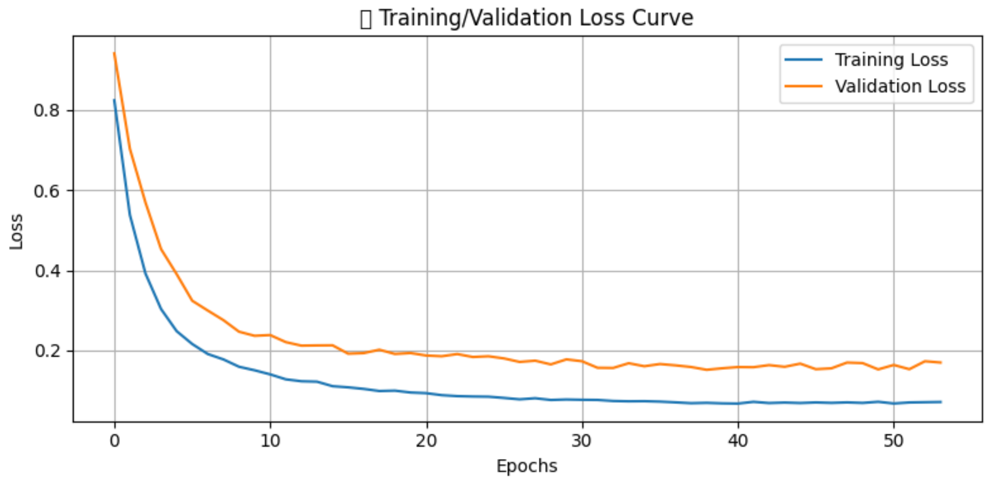

Figure 2: Training and Validation Huber Loss over Epochs best model GRU

4.2 Classification

Metrics Comparison for base and selected model(BiLSTM):

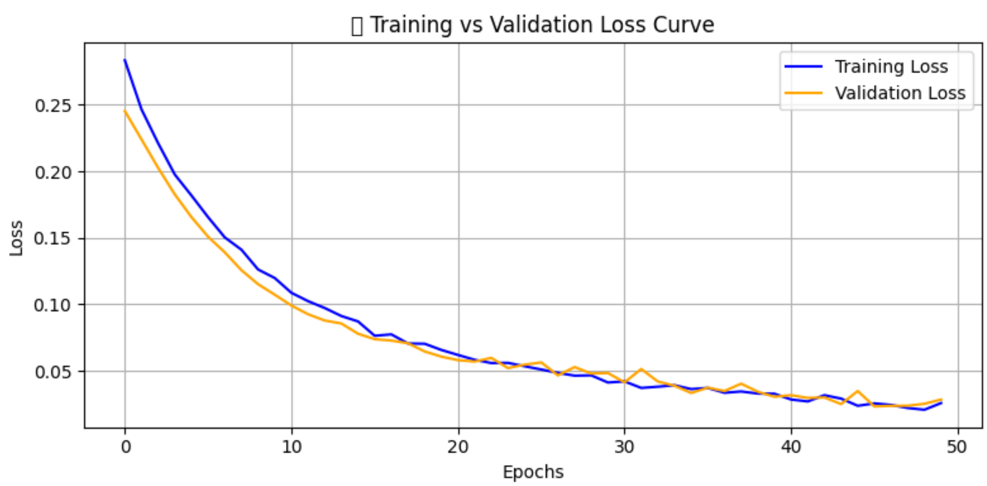

Figure 3: Training and Validation Loss over Epochs

Model	Accuracy	Class	Precision	Recall	F1-score
Base RNN	76.18%	0	0.749	0.654	0.698
		1	0.769	0.840	0.803
Best BiLSTM	86.11%	0	0.743	0.927	0.825
		1	0.954	0.825	0.885

Table 4: Comparison of classification performance metrics between the base RNN and BiLSTM model.

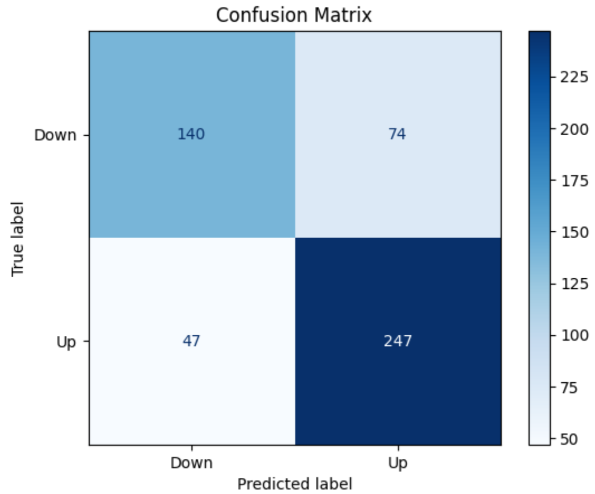

Figure 4: Confusion matrix for base RNN model

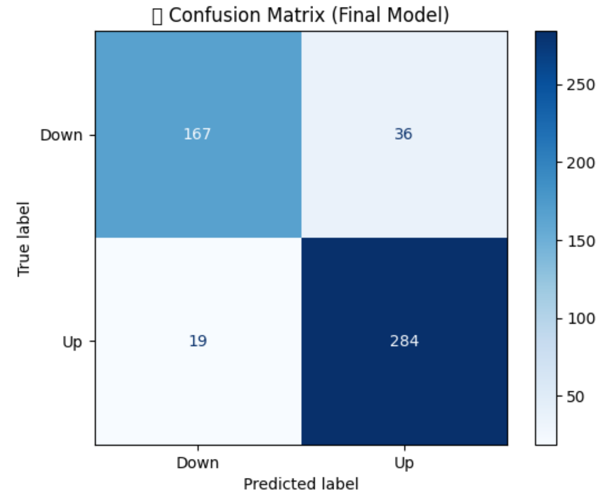

Figure 5: Confusion matrix for BiLSTM model

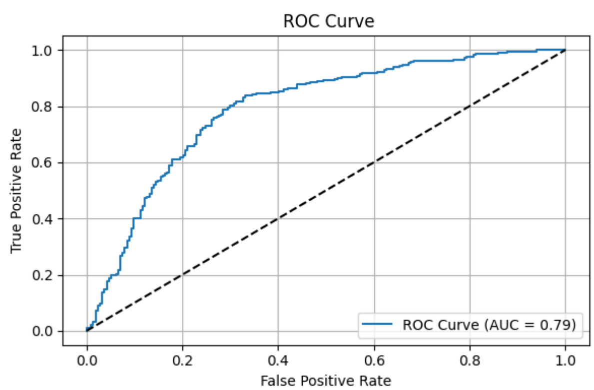

Figure 6: ROC AUC curve for base RNN model

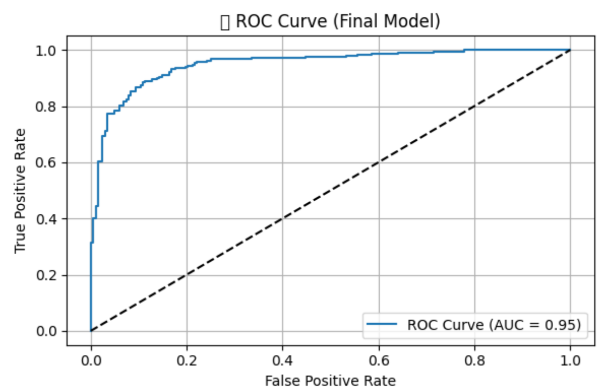

Figure 7: ROC AUC curve for BiLSTM model

5 Conclusion

This project demonstrated the effectiveness of deep learning architectures—GRU-based models—for stock price prediction through regression and BiLSTM model for stock price classification tasks. Compared to baseline RNN models, the tuned GRU and BiLSTM models consistently achieved better performance, reflected in lower MSE and MAE for regression, and higher accuracy, precision, and recall for classification.

The systematic approach to hyperparameter tuning and the use of cross-validation played a crucial role in enhancing model generalization and preventing overfitting. The classification model accurately predicted the direction of stock price movement, while the regression model provided reliable estimates of future closing prices.

These results underscore the potential of deep learning in financial forecasting when combined with thorough data preprocessing and engineering of technical indicators.

Future work may involve incorporating more recent architectures such as Transformer-based models, applying ensemble learning techniques to combine predictions, or including external data sources like financial news or sentiment analysis to further enhance predictive performance.

References

[1] Achelis, S. B. (2001). *Technical Analysis from A to Z* (2nd ed.). McGraw-Hill.

[2] Ran, R. (2019). *yfinance: Download market data from Yahoo! Finance API*. Retrieved from =`https://github.com/ranaroussi/yfinance`

[3] TensorFlow Tutorials. (2024). Retrieved from https://www.tensorflow.org/tutorials

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[5] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. https://doi.org/10.1109/78.650093

[6] TensorFlow Team. *tf.keras.layers.GRU - TensorFlow*. Available at: https://www.tensorflow.org/api$_{d}ocs/python/tf/keras/layers/GRU$ (Accessed: 2025-08-02).