Report: Stock Price Prediction Using LSTM with Feature Engineering

Deniz GÖKDUMAN

1. Introduction

This report outlines the development of a predictive model for stock price forecasting using Long Short-Term Memory (LSTM) networks. The model is designed to predict the next day's closing price of a stock based on historical data, incorporating technical indicators and feature engineering techniques. The dataset used consists of stock data for four companies: TUPRS, ISMEN, KOZAL, and PENTA. The model is evaluated using appropriate metrics, and the results are analyzed.

2. Feature Engineering

Feature engineering is a critical step in improving the performance of predictive models. In this project, the following technical indicators were added to the feature set:

A. **Relative Strength Index (RSI):** A momentum oscillator that measures the speed and change of price movements. RSI is calculated over a 5-day period.
B. **On-Balance Volume (OBV):** A momentum indicator that uses volume flow to predict changes in stock price.
C. **Simple Moving Averages (SMA):** SMA20 and SMA60 represent the 20-day and 60-day moving averages, respectively. These help identify trends over different time horizons.
D. **Bollinger Band Squeeze (BBS):** A volatility indicator derived from Bollinger Bands. It measures the relative width of the bands and helps identify periods of low volatility, which often precede significant price movements.

These features were added to the dataset to provide the model with more context about price trends, momentum, and volatility. The dataset was cleaned by removing rows with missing values after feature engineering.

3. Model Development

The predictive model was developed using an LSTM architecture, which is well-suited for time-series data due to its ability to capture temporal dependencies. The model was trained to predict the next day's closing price of the TUPRS stock.

Model Architecture

The LSTM model consists of the following layers.

A. **LSTM Layer:** 128 units to capture temporal patterns in the data.
B. **Dense Layers:** Two fully connected layers with 512 units each, using ReLU activation and L2 regularization to prevent overfitting.
C. **Dropout Layers:** Dropout with a rate of 0.5 is applied after each dense layer to further reduce overfitting.
D. **Output Layer:** A single unit with no activation function to predict the closing price.

Training

The model was trained using the following parameters:

- **Loss Function:** Mean Squared Error (MSE) to minimize the difference between predicted and actual closing prices.
- **Optimizer:** Adam optimizer with a learning rate of 0.001.
- **Early Stopping:** Training stops early if the validation loss does not improve for 4 consecutive epochs to prevent overfitting.
- **Epochs:** A maximum of 32 epochs was used, with early stopping applied.

The dataset was split into training (80%) and validation (20%) sets. The input data was standardized to have zero mean and unit variance to improve model convergence.

4. Evaluation

The model's performance was evaluated using the following metrics:

A. **Mean Squared Error (MSE):** Measures the average squared difference between predicted and actual values. Lower values indicate better performance.
B. **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual values. It provides a more interpretable measure of error magnitude.

Results

The model achieved the following results on the validation set:

- Validation Loss (MSE): [Insert value here]
- Validation MAE: [Insert value here]

The predictions were compared to the actual closing prices, and the results were visualized to assess the model's accuracy. The plot of predicted vs. actual values shows that the model captures the general trend of the stock price, although there are some deviations during periods of high volatility.

5. Discussion

The model performed well in predicting the next day's closing price, as evidenced by the low validation loss and MAE. The inclusion of technical indicators such as RSI, OBV, SMA, and Bollinger Bands provided the model with additional context, improving its ability to capture trends and patterns in the data.

However, there are areas for improvement:

- **Hyperparameter Tuning:** Further tuning of hyperparameters, such as the number of LSTM units, learning rate, and dropout rate, could improve performance.
- **Additional Features:** Incorporating macroeconomic indicators or sentiment analysis from news articles could provide additional context for the model.
- **Ensemble Models:** Combining LSTM with other models, such as gradient boosting or random forests, could improve robustness and accuracy.

6. Code Implementation

The Python code provided implements the steps outlined in this report.Key components include:

Feature Engineering: Calculation of RSI, OBV, SMA, and Bollinger Bands.

Data Preprocessing: Standardization and splitting into training and validation sets.

Model Development: LSTM architecture with dropout and regularization.

Training and Evaluation: Early stopping, loss metrics, and visualization of results.

The code is modular and can be adapted for other datasets or prediction tasks with minimal modifications.

7. Future Work

- **Hyperparameter Optimization:** Use grid search or Bayesian optimization to find the best hyperparameters.
- **Multi-Step Forecasting:** Extend the model to predict prices for multiple time steps ahead.
- **Real-Time Prediction:** Deploy the model in a real-time trading environment to evaluate its practical utility.

This project serves as a foundation for further exploration of machine learning techniques in financial forecasting.