Predicting AMD Stock Prices Using Historical Data and Moving Averages

(COMP3125 Individual Project)

Maxim Leonov
Wentworth Institute of Technology

***Abstract*—This project investigates using historical data and moving averages to predict AMD stock prices. I examined how well moving averages predict price trends, whether historical data can predict future prices, and how different timeframes affect prediction accuracy. Moving averages showed moderate success as predictors. After encountering ARIMA model limitations, an LSTM neural network was implemented, demonstrating good prediction accuracy. The model predicts an upward trend in AMD's stock price.**

Keywords—Stock Prediction, moving averages, LSTM, technical analysis, trend forecasting

I. Introduction

Stock price prediction is an important area in finance and investment management. This project focuses on predicting AMD stock prices using technical analysis tools and time series prediction. Moving averages help identify trends by creating constantly updated average prices over specific time periods. At first, this project attempted to use an ARIMA model, but due to some challenges, a Long Short-Term Memory (LSTM) neural network was used instead. LSTM networks can capture long-term dependencies in sequential data, making them suitable for stock price prediction. Three key research questions were addressed: (1) how well moving averages predict AMD's price trends, (2) whether historical data can predict future prices using time series analysis, and (3) how different moving average timeframes influence predictive accuracy. This project helps the understanding of quantitative approaches for stock prediction.

II. Datasets

A. Source of dataset

The main dataset used in this analysis is historical AMD stock price data obtained from a Kaggle dataset (https://www.kaggle.com/datasets/muhammaadmuzammil008/amd-stock-data). The dataset contains daily price information from 1983 to 2024. The data was originally took from Yahoo Finance, a verified provider of financial market data.

B. Character of the datasets

The dataset is in CSV format and contains over 10,000 rows of daily stock price data spanning more than 40 years. Each row represents a trading day with the following attributes:

Column	***Description***	***Data Type***
Date	Trading Date	Date
Open	Opening price of the day	Float
High	Highest price of the day	Float
Low	Lowest price of the day	Float
Close	Closing price of the day	Float
Volume	Number of shares traded	Integer

For this analysis, I focused on the most recent three years of data (757 trading days) for relevance to current market conditions. The data was well-structured and required minimal cleaning. I also calculated several derived features to support the analysis:

1. Moving Averages: 5-day, 10-day, 20-day, 50-day, and 200-day simple moving averages
2. Trading Signals: Indicators for buy and sell signals based on moving average crossovers
3. Daily Returns: Percentage change in closing price from the previous day

The recent 3-year dataset provides a good mix of market conditions, it includes both bullish and bearish periods, making it good for evaluating the effectiveness of the predictive methods.

III. Methodology

A. Moving Average Analysis

A simple moving average (SMA) is calculated by taking the arithmetic mean of a stock's closing prices over a specified period. For this project, I calculated several moving averages:

- 5-day SMA (very short-term)
- 10-day SMA (short-term)
- 20-day SMA (medium short-term)
- 50-day SMA (medium-term)
- 200-day SMA (long-term)

The formula for calculating a simple moving average is:

$$SMA(n) = (P_1+P_2+\ldots+P_n)/n \tag{1}$$

Where:

- SMA(n) is the simple moving average over n days
- $P_1, P_2, \ldots, P_n$ are the closing prices for the most recent n days

To generate trading signals, I used moving average crossovers:

- Golden Cross (buy signal): When a shorter-term MA crosses above a longer-term MA
- Death Cross (sell signal): When a shorter-term MA crosses below a longer-term MA

I tested four crossover combinations:

979-8-3315-0993-4/25/$31.00 ©2025 IEEE

1. MA5/MA20: Very short-term vs. short-term
2. MA10/MA50: Short-term vs. medium-term
3. MA20/MA50: Medium short-term vs. medium-term
4. MA50/MA200: Medium-term vs. long-term

The advantage of moving averages is their simplicity. However, they are lagging indicators, meaning they confirm trends after they have begun rather than predicting them in advance.

B. ARIMA Model Attempt

Initially, I attempted to implement an ARIMA(1,1,0) model for time series forecasting. ARIMA is a general class of models that captures several aspects of a time series:

- Autoregressive (AR): Uses the dependent relationship between an observation and a number of lagged observations
- Integrated (I): Uses differencing to make the time series stationary
- Moving Average (MA): Uses the dependency between an observation and a residual error from a moving average model

The ARIMA(1,1,0) model can be expressed as:

$$(1-\varphi_1 B)(1-B)Y_t = \varepsilon \qquad (2)$$

Where:

- B is the backshift operator ($BY_t = Y_{t-1}$)
- φ_1 is the autoregressive coefficient
- ε_t is the error term

However, during implementation, I encountered challenges with the ARIMA model. The model fit successfully, but it was unable to generate valid predictions, resulting in NaN (Not a Number) values for the predicted prices. This made it hard to calculate meaningful error metrics or create useful visualizations for the ARIMA predictions.

C. LSTM Model Implementation

Following the suggestion from my professor, I implemented a Long Short-Term Memory (LSTM) neural network as an alternative approach. LSTM networks are a type of recurrent neural network designed to capture long-term dependencies in sequential data, making good for stock price prediction.

The LSTM model architecture used in this project consisted of:

- An input layer accepting sequences of 20 days of price data
- A single LSTM layer with 50 units
- A dense output layer with a single unit to predict the next day's price

Before training the model, the data was preprocessed:

- The closing prices were scaled to a range of [0, 1] using MinMaxScaler
- The data was split into 80% training and 20% testing sets
- The data was transformed into sequences of 20 consecutive days (input) and the following day's price (target)

The model was trained for 5 epochs with a batch size of 32, using the Adam optimizer and mean squared error loss function. After training, the model was used to make predictions on the test set and to predict prices for the next 30 days.

IV. Results

A. Moving Average Prediction Accuracy

The analysis of moving average crossovers showed interesting stuff about their effectiveness for predicting AMD's stock price trends.

All four MA strategies (MA5/MA20, MA10/MA50, MA20/MA50, and MA50/MA200) showed same trend prediction accuracy at 52.94%. This means that approximately 53% of the signals generated by these strategies correctly predicted the future price movement direction over a 20-day horizon.

The number of signals generated varied significantly across strategies:

- MA5/MA20: 24 buy signals, 23 sell signals (most active)
- MA10/MA50: 10 buy signals, 10 sell signals
- MA20/MA50: 9 buy signals, 8 sell signals
- MA50/MA200: 1 buy signal, 1 sell signal (least active)

In terms of the overall returns, the strategies showed varying performance:

- MA5/MA20: 77.99% return
- MA10/MA50: 25.31% return
- MA20/MA50: 29.11% return
- MA50/MA200: 79.99% return

It's interesting to note that while all strategies had the same prediction accuracy, their returns varied significantly, with the MA50/MA200 strategy showing the highest return although generating the fewest signals. This means that while the strategies may be equally good at predicting direction, the magnitude and timing of the movements they capture differ widely.

To compare, a simple buy-and-hold strategy would have had a 90.10% return over the same period, outperforming all the MA crossover strategies.

B. ARIMA Model Results

As mentioned in the methodology section, the ARIMA (1,1,0) model was successfully fit to the data, but had problems when generating predictions. The model summary showed the following parameters:

	coef	*std err*	*z*	*P>|z|*	*[0.025*	*0.975]*
ar.L1	-0.02	0.035	-0.704	0.482	-0.093	0.044
sigma2	10.66	0.438	24.329	0.000	9.802	11.520

The negative AR(1) coefficient (-0.0246) means a slight mean-reverting behavior in the differenced series, but the p-value of 0.482 means that this coefficient is not statistically significant. Despite the model fitting, when attempting to generate predictions, the model produced NaN values, making it impossible to evaluate its predictive performance or to use it for future price predictions.

C. LSTM Model Results

The LSTM model demonstrated promising results for both short-term predictions and future forecasting.

Training and Testing Performance:

- During training, the model showed rapid convergence, with the loss decreasing from 0.0437 in the first epoch to 0.0030 by the fifth epoch.
- When evaluated on the separate test set (using all data), the model achieved:
  - Root Mean Squared Error (RMSE): 6.96
  - Mean Absolute Percentage Error (MAPE): 4.27%

This indicates that, on average, the model's predictions were within 4.27% of the actual prices, which is a good level of accuracy for stock price prediction.

Recent Prediction Performance:

- When tested on the most recent 40 days of data (which is more relevant for evaluating current performance), the model showed:
  - RMSE: 30.02
  - MAPE: 18.04%

The higher error on the most recent data means that there might have been increased volatility or changing patterns in the most recent period that were more challenging for the model to capture.

Future Price Forecast:

Based on the last known closing price of $173.87 on February 16, 2024, the LSTM model predicted the following prices for selected future dates:

Date	*Predicted Price*
Feb 19, 2024	$174.93
Feb 23, 2024	$176.23
Mar 1, 2024	$178.51
Mar 15, 2024	$182.19
Mar 29, 2024	$184.87

The model predicts a gradual upward trend, with a predicted price of $184.87 after 30 trading days, representing a 6.33% increase from the last known price.

V. DISCUSSION

Despite promising results, this study has several limitations. The moving average strategies, while showing some predictive power with a 52.94% accuracy rate, performed only a little better than random chance. All MA strategies did worse than a simple buy-and-hold approach during the study period, meaning limited utility in strong bull markets.

The ARIMA model implementation was not quite good. It failed to generate valid predictions despite successful model fitting. The model produced NaN values when predicting, making it unusable for application. This shows the challenges of applying common statistical models to volatile financial data.

The LSTM model, while mostly effective with a 4.27% MAPE on test data, showed worse performance (18.04% MAPE) on the most recent 40 days. This inconsistency can question the model's reliability during changing market conditions or periods of increased volatility.

Future work could fix these limitations through several approaches. First, extending the analysis period to include different market cycles would better assess strategy power. Applying the same methods over multiple stocks or market sectors would determine if findings generalize beyond AMD. A more complex LSTM architecture having more features beyond price data might improve performance. Also, combining technical indicators with LSTM predictions in a hybrid approach could enhance overall accuracy. Finally, using reinforcement learning algorithms could optimize entry and exit points based on changing market conditions.

VI. CONCLUSION

This project measured the effectiveness of using historical data and moving averages to predict AMD stock prices. The key findings were:

1. Moving averages showed moderate effectiveness in identifying trend changes. All tested MA strategies achieved a trend prediction accuracy of 52.94%.
2. Different moving average timeframes show a clear exchange between signal frequency and performance, with very short-term (MA5/MA20) and very long-term (MA50/MA200) strategies outperforming medium-term approaches for AMD stock during the analyzed period.
3. The ARIMA model encountered challenges in generating valid predictions, showing the limitations of common statistical approaches for complex financial time series.
4. The LSTM model showed good predictive performance with a MAPE of 4.27% on test data, supporting the use of deep learning approaches for stock price prediction.
5. Based on the LSTM model, AMD's stock price is predicted to increase by 6.33% over the next 30 trading days, reaching about $184.87 by late March 2024.

These findings are useful for investors interested in AMD stock or similar technology companies. They show that while technical indicators like moving averages have some predictive power, more complex approaches like LSTM networks can provide more accurate predictions. However, the consistent underperformance of all MA strategies compared to a buy-and-hold approach during this period also shows how important considering the whole market context when developing trading strategies is.

This project shows the use of data science techniques to financial market analysis. It shows both the potential and limitations of different approaches to stock price prediction.

References

[1] Murphy, J. J. (1999). Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance.

[2] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control. John Wiley & Sons.

[3] Edwards, R. D., Magee, J., & Bassetti, W. H. C. (2018). Technical Analysis of Stock Trends. CRC Press.

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.

[5] Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. Neural Computing and Applications, 32(6), 1609-1628.

[6] Yahoo Finance. (2023). Advanced Micro Devices, Inc. (AMD) Historical Data. Retrieved from https://finance.yahoo.com/quote/AMD/history/