Project i Data Scientist

Kunskapskontroll 2



Anna Strbac

EC Utbildning

2024-11

**Abstract**

This study focuses on developing a user-friendly mobile application designed to predict next day Bitcoin price fluctuations and provide actionable buy and sell signals. To achieve accurate and reliable price forecasting, the app leverages a Long Short-Term Memory (LSTM) neural network model, known for its effectiveness in time series prediction tasks. Various configurations of the LSTM model will be explored, adjusting hyperparameters, network depth, and other structural elements to optimize performance. Additionally, multiple input features, such as historical price data, trading volume, technical indicators, and macroeconomic data, will be incorporated and evaluated to identify the most impactful variables for improving prediction accuracy.

To determine the most effective configuration, we will use Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) as the primary metrics for evaluating prediction accuracy, as they offer a clear assessment of the model’s forecasting error. Through back-testing, the study will assess the selected model’s capacity to adapt to various market conditions, aiming to deliver reliable and actionable trading insights for users.

The final goal is to deliver an application that simplifies cryptocurrency trading by merging machine learning techniques with an accessible, user-centric design.

**Keywords: Bitcoin Price Prediction, Long Short-Term Memory (LSTM), Mean Absolute Error, Root Mean Squared (RMSE)**

1 Introduction

The cryptocurrency market, particularly Bitcoin, attracts attention due to its high volatility and the potential for significant profits. However, the rapid price fluctuations make accurate predictions challenging, requiring advanced tools to improve forecasting accuracy. In recent years, neural network-based methods have shown strong results in analyzing time series data and predicting short-term price movements.

This study aims to explore the potential of machine learning techniques, particularly Long Short-Term Memory (LSTM) models, for predicting Bitcoin's short-term price movements. LSTMs are effective at capturing the time-dependent patterns in data, making them well-suited for analyzing the dynamics of the cryptocurrency market. Accurate Bitcoin price forecasts can be highly valuable in automated trading, portfolio management, and financial risk assessment.

Machine learning models that generate clear buy or sell signals based on short-term predictions greatly enhance decision-making, allowing for more flexible trading strategies. In practical applications, the accuracy and reliability of these forecasting models are crucial for making informed decisions. Even small errors in predictions can lead to significant financial losses or missed opportunities. For example, in an automated trading system, a wrong prediction might cause an exit from a trade or an entry at the wrong time, resulting in losses. Therefore, achieving high accuracy in forecasting is essential to provide users with reliable recommendations and support successful investment decisions.

* 1. Objectives

Predict Bitcoin prices and provide buy/sell signals.

Key Goals:

* Develop a model to predict short-term Bitcoin price movements.
* Generate buy/sell signals based on predicted trends and technical indicators.
* Create a user-friendly app to display predictions and signals.

1.2 Research questions

To achieve this objective, the following inquiries will be addressed:

* What are the most relevant features and indicators for predicting short-term Bitcoin price movements?
* How reliable of a model can we build using these features for Bitcoin price prediction?
* What factors might hinder the model from making accurate predictions?

2 Theory

This section introduces theoretical concepts essential for contextual comprehension of this study.

2.1 Train–validation-test split methodology

Training data is used to train and make the model learn the hidden patterns in the data. It's the portion of the dataset that the model has access to during the training process. Validation data is used to adjust the model's hyperparameters and to evaluate its performance during training. Test data represents new, unseen data that the model hasn't been exposed to during training or validation. Test data is used to evaluate the model's performance after it has been trained and fine-tuned.

2.2 Data preprocessing

2.2.1 Feature Engineering

Feature engineering involves creating new features or modifying existing ones to enhance the performance of a machine learning model. This process focuses on extracting and transforming relevant information from raw data, making it more interpretable and useful for the model. By incorporating features that better represent the underlying patterns and relationships in the data, the goal is to improve the model's accuracy and predictive capability [1].

2.2.2 Min Max Scaler

The Min Max Scaler is a data scaling technique for normalization that sets the minimum value of a feature to zero and the maximum value to one. By compressing the data within a predefined range, typically between 0 and 1, it transforms the features to fit within this range without altering the shape of the original distribution.

2.3 Performance Measures

2.3.1 MSE / RMSE

Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals. The lower the MSE, the better a model fits a dataset.

[Root Mean Squared Error](https://www.analyticsvidhya.com/blog/2021/05/know-the-best-evaluation-metrics-for-your-regression-model/) is the square root of Mean Squared error. It measures the standard deviation of residuals. The lower the RMSE, the better a model fits a dataset [2].

2.4 Machine Learning Models

2.4.1 LSTM

LSTM (Long Short-Term Memory) is a powerful variant of recurrent neural networks (RNNs) specifically designed for processing sequential data and making predictions based on time-series information [3]. Traditional RNNs often face challenges in capturing long-term dependencies due to the vanishing gradient problem. LSTMs address this limitation with a unique architecture that allows them to retain and utilize information over extended sequences effectively.

The LSTM network achieves this through three primary gates: the input gate, the forget gate, and the output gate. The input gate controls which new information is added to the cell state, the forget gate decides which information to remove, and the output gate determines the portion of the cell state to use for generating the output. This gate-based mechanism ensures efficient management of both short-term and long-term dependencies within the data.

By capturing intricate temporal patterns and relationships, LSTMs excel in various applications, including natural language processing, audio analysis, and financial forecasting, such as predicting stock prices or cryptocurrency trends.

2.4.2 Grid Search

Grid Search is a traditional method used for hyperparameter tuning in machine learning. It exhaustively tries every combination of the provided hyper-parameter values , including the number of LSTM units, learning rates, dropout rates, activation functions, and regularization techniques to find the best model [4].

2.4.3 Data Augmentation

Data augmentation is a technique used to artificially increase the size of a training dataset by applying various transformations to the existing data. This technique is commonly used in machine learning and deep learning tasks, to improve the generalization and robustness of the trained models [5].

2.4.4 SHAP

SHAP (SHapley Additive exPlanations) values are a widely used technique in machine learning to interpret model predictions by quantifying the contribution of each feature to the final output. SHAP values provide a structured approach to understand the importance of individual features in a predictive model [6].

2.5 Indicators for Financial Prediction

2.5.1 Technical Indicators

Technical indicators utilize historical price data to detect market trends and predict potential price movements:

* Bollinger Bands: Measure price volatility and potential overbought/oversold conditions.
* SMA (Simple Moving Averages): Smooth past price data over 10 and 50 periods to identify trends.
* RSI (Relative Strength Index): Evaluates momentum, indicating overbought or oversold market conditions.
* MACD (Moving Average Convergence Divergence): Tracks the relationship between two moving averages to signal trend changes.
* Signal Line: A smoothed MACD line used to confirm buy or sell signals.
* Bullish and Bearish Indicators: Identify upward (bullish) or downward (bearish) price trends.
* Resistance and Support Levels: Key price levels where the market historically struggles to rise above (resistance) or fall below (support).

2.5.2 Macroeconomic Indicators

These indicators reflect broader economic conditions that impact financial markets:

* CPIAUCNS: Consumer Price Index for All Urban Consumers, measuring inflation.
* SAHMREALTIME: A real-time measure of labor market conditions.
* UNRATE: The unemployment rate, reflecting job market strength.
* EFFR: Effective Federal Funds Rate, indicating monetary policy and interest rate trends.

2.5.3 Macro-Financial Indicators

Indicators representing broader market dynamics and external economic influences:

* S&P 500: A benchmark index for U.S. equities, reflecting overall stock market performance.
* XAU/USD: The price of gold in USD, often seen as a hedge against market volatility and economic uncertainty.
* NVDA: Nvidia’s stock price, relevant due to its role in technology and cryptocurrency mining sectors.
* DXY: The U.S. Dollar Index, tracking the dollar’s strength against a basket of other currencies, often inversely related to Bitcoin.

2.5.4 Blockchain-Related Indicators

Metrics specific to Bitcoin's blockchain network:

* Hash Rate: Represents the total computational power used to secure the Bitcoin network, linked to miner activity and network security.
* Block Reward: Refers to the reward miners receive for validating a block, influencing Bitcoin’s supply dynamics and miner profitability.

3 Metod

In this chapter, we delve into data exploration, model selection, and the techniques used for Bitcoin price prediction.

3.1 Tools

The Bitcoin price prediction model was developed using TensorFlow for building and training the LSTM model. Pandas and NumPy were used for data manipulation and preprocessing. Matplotlib was employed for visualizing model predictions against actual Bitcoin prices. Streamlit was used to create an interactive web interface for displaying predictions.

3.2 Data Collection

The OHLCV (Open, High, Low, Close, Volume) and Market Cap data were sourced from CoinCodex.com in CSV format. Economic data, including CPIAUCNS, SAHMREALTIME, UNRATE, and EFFR, were retrieved from fred.stlouisfed.org. Data for the S&P 500, XAU/USD, NVDA, and DXY were obtained from [www.investing.com](http://www.investing.com/). The Hash Rate data was sourced from [www.blockchain.com](http://www.blockchain.com/). Block Reward (blockchain-related indicators) was sourced from [www.bitcoinblockhalf.com](http://www.bitcoinblockhalf.com/). All other features were created using feature engineering techniques, where new features were derived from the raw data to enhance model performance and capture relevant patterns in the Bitcoin price prediction process. Below are the key features included in the dataset:

* Price-related features: Close-t-1 (lagged), Open-t-1 (lagged), High-t-1 (lagged), Low-t-1 (lagged), VWAP-t-1 (Volume-Weighted Average Price lagged).
* Volatility and trend indicators: Bollinger Bands indicators, SMA (Simple Moving Averages) of 10, 50 periods, RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), Signal Line.
* Market indicators: Lagged Volume and Lagged Market Cap.
* Price action features: Bullish and Bearish indicators, Resistance, Support levels, MACD crossed below and above .
* Time-based features: Day of the week, Month, Year.

These features provide a comprehensive view of the market, capturing various factors that can influence Bitcoin’s price movement.

3.3 Data Exploration

During the data exploration phase, a comprehensive analysis was performed to understand the dataset’s structure, key characteristics, and the relationships between variables. This analysis provided valuable insights into the market, capturing the various factors influencing Bitcoin’s price movements. Exploratory Data Analysis (EDA) was conducted to examine the distribution, interrelationships, and patterns within the features, while also identifying potential data cleaning and preprocessing tasks. These included handling missing values, detecting outliers, and transforming features into appropriate formats for the LSTM model.

Summary statistics, such as mean, standard deviation, minimum, maximum, and percentiles, were calculated for all features to gain a deeper understanding of their distribution. Correlation matrices were also created to identify strong correlations, especially between technical indicators and Bitcoin prices.

Through these exploratory steps, the dataset was refined, key features were selected, and the data was prepared for model training, laying a solid foundation for the subsequent machine learning processes.

3.4 Training and Evaluation

In this study, we utilize the train-validation-test split methodology, dividing the dataset into three essential subsets: the training set, the validation set, and the test set.

The split uses a random process to divide the data. In this methodology, the random state parameter ensures reproducibility by fixing the random seed used for the splitting process.

3.5 Data preprocessing

The Min Max Scaler were applied to preprocess the data when using LSTM model.

Missing values were handled, and features were scaled to ensure that the LSTM model could process them efficiently.

3.6 Model Implementation

The Long Short-Term Memory (LSTM) model was employed for predicting Bitcoin prices due to its effectiveness in handling sequential and time-series data. To optimize the model's performance, hyperparameter tuning was performed using Grid Search.

The model’s performance was evaluated using metrics such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to assess the accuracy of the predictions. The model’s generalization ability was further evaluated by using test loss. The best model was selected based on the lowest test loss, as this indicated the model’s ability to generalize well to unseen data. The model was then retrained using both the training and validation sets to further improve its robustness.

SHAP (SHapley Additive exPlanations) was used to analyze feature importance and understand how individual features influenced the model’s predictions. Data augmentation techniques were applied to improve the model by increasing the diversity of the training data, thereby enhancing its generalization capability.

4 Results and Discussion

In this chapter, we present a detailed comparison of the models' performance, analyze the challenges faced in Bitcoin price prediction, and discuss the results obtained from various experiments. This section aims to provide insights into the effectiveness of different approaches, the impact of feature selection, and how model performance evolved with different parameters.

4.1 Model Selection

To develop an effective model for short-term Bitcoin closing price prediction, several LSTM (Long Short-Term Memory) models were evaluated, each tested with a variety of features and hyperparameters. The goal was to identify the optimal combination of parameters and features that could deliver the most accurate predictions.

Initially, a simpler model was built using just one feature: Lagged Close1 (the previous day’s closing price) to predict the next day’s closing price (t+1). Despite its simplicity, this model performed well, achieving a Mean Squared Error (MSE) of 0.00078 and a Root Mean Squared Error (RMSE) of 0.028. These metrics suggest that the model was capable of making relatively accurate predictions. However, it is important to note that the model relies solely on historical data—such as past prices—to forecast future values. This means it cannot anticipate sudden shifts or external market influences in real-time, resulting in what is commonly referred to as a "prediction delay." This delay is an inherent limitation in time series forecasting, particularly when models are trained exclusively on historical data. When a model is trained on lagged features, such as the previous day's closing price, this delay becomes inevitable, as the model lacks the ability to "see" future data. In highly volatile markets like Bitcoin, this delay can limit the model’s usefulness for real-time trading, as the model may struggle to react quickly to sudden market shifts.

4.2 Feature Importance

The feature selection process provided valuable insights into the factors influencing the model's performance. A variety of features were tested, including market-related data such as OHLCV (Open, High, Low, Close, Volume) values. For each day, lagged versions of these features were created using the values from previous days. These past values helped the model recognize historical patterns and price trends without introducing future information.

In addition to OHLCV data, a range of other features was tested, including technical indicators such as the Simple Moving Average (SMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Bollinger Bands. Macroeconomic indicators like the Unemployment Rate (Unrate), Consumer Price Index for All Urban Consumers (CPIAUCNS), and the SAHM Real-Time Economic Indicators were also considered. Other factors, such as the performance of the S&P 500 Index, the stock price of Nvidia (NVDA), the U.S. Dollar Index (DXY), and the price of Gold (XAUUSD), were included in the analysis. Additionally, cryptocurrency-specific metrics such as Bitcoin's hash rate were tested. However, not all of these features contributed significantly to the model's performance.

One feature that slightly improved the model’s performance was the inclusion of the 'Lagged Open-Close Range,' which represents the difference between the opening and closing prices of the previous day. By incorporating this feature, the model experienced a small reduction in both the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). This suggests that accounting for daily price fluctuations (or volatility) plays a role in enhancing the model’s ability to forecast Bitcoin’s short-term price movements.

However, further attempts to add additional features did not yield substantial performance gains. This finding implies that while the model benefits from certain additional context, excessive feature complexity does not necessarily enhance prediction accuracy. Furthermore, this suggests that the LSTM model, as a sequential neural network, can inherently learn important patterns in time series data, including those provided by various technical indicators. Therefore, the simpler model, utilizing only Lagged Close 1 and Lagged Open-Close Range, ultimately demonstrated the best performance.

4.3 Further AdjustmentsTo improve robustness, data augmentation techniques were applied by introducing slight noise and scaling variations. However, this step did not enhance the model's generalization. It did not lead to an improvement in performance and even worsened the results, suggesting that the augmentation may have introduced unnecessary variations that the model struggled to handle effectively.

4.4 Results Comparison

Here is a summary of the results from different models and configurations tested:

| **Model Configuration** | **MSE** | **RMSE** | **Comments** |
| --- | --- | --- | --- |
| Lagged Close 1 (t+1) | 0.00064 | 0.025 | Simple model, good performance |
| Lagged Close 1 + Lagged Open Close Range (t+1) | 0.00063 | 0.024 | Improvement from additional feature |
| Lagged Close 1 + Lagged Open-Close Range (t+1) + Data Augmentation | 0.00067 | 0.025 | No improvement with data augmentation |

Tabell 1: MSE / RMSE for the examined models

The results indicate that the Lagged Close 1 model is effective for predicting short-term Bitcoin prices. The addition of the Lagged Open-Close Range feature slightly enhanced the model's predictive power. However, incorporating data augmentation did not result in any significant improvement. More complex models and additional features did not substantially improve performance, confirming that a simpler approach is most effective for this task. Therefore, the model with the additional Lagged Open-Close Range feature was selected for making predictions in the application.

Back testing results for the best model demonstrated that, despite fluctuations in the Bitcoin market, the model's predictions generated a positive return of 363.85% for the year 2024. This highlights the profitability of the trading strategy. However, not all trades were profitable, and some resulted in losses, which reflects the inherently volatile nature of the cryptocurrency market.

4.5 Conclusion

In conclusion, the results demonstrate that a simpler model, using just Lagged Close1 and Lagged Open-Close Range, outperformed more complex feature sets. This highlights the value of a parsimonious approach for this application, where the selected features effectively capture core trends in the data while avoiding unnecessary complexity. Additionally, the study underscores the challenges of using time series models like LSTM, especially in highly volatile markets like Bitcoin, where prediction delays can hinder real-time accuracy. Despite these challenges, the optimized LSTM model serves as a valuable support tool for short-term price forecasting, relying on historical data. However, it remains limited in its ability to predict future trends accurately.

For further model improvement, it might be necessary to better understand the factors that impact Bitcoin's price and incorporate additional relevant data. Leading indicators that could influence future trends should be explored. For example, sentiment analysis of Bitcoin-related news could provide valuable insights into market sentiment, which often drives price movements. Other factors to consider include social media activity (such as tweets, forum discussions, or Reddit posts), regulatory news, or even on-chain data like wallet movements and miner activity. Integrating these additional features could enable the model to capture more complex patterns and enhance its predictive accuracy.

Therefore, in my application, I have incorporated additional technical indicators to help make decisions about buying or selling Bitcoin. However, it is important to note that relying solely on this model is not advisable, as it cannot predict future trends. It uses past information to make decisions, and the future may not follow the same patterns as the past.

References

[1] “Feature Engineering”:

[*https://www.geeksforgeeks.org/what-is-feature-engineering/*](https://www.geeksforgeeks.org/what-is-feature-engineering/)

[2] “MSE /RMSE metrics”:

[*https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e*](https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e)*/*

[3] “Long-Short Memory (LSTM) model”:

[*https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/*](https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/)

[4] “Grid Search”:

[*https://www.dremio.com/wiki/grid-search/#:~:text=Grid%20Search%20is%20a%20traditional,inefficient%20with%20high%2Ddimensional%20data*](https://www.dremio.com/wiki/grid-search/#:~:text=Grid%20Search%20is%20a%20traditional,inefficient%20with%20high%2Ddimensional%20data)*.*

[5] “SHAP (Shapley Additive Explanations) “:

[*https://medium.com/@anshulgoel991/model-exploitability-using-shap-shapley-additive-explanations-and-lime-local-interpretable-cb4f5594fc1a*](https://medium.com/@anshulgoel991/model-exploitability-using-shap-shapley-additive-explanations-and-lime-local-interpretable-cb4f5594fc1a)

[6] “Data Augmentation“:

[*https://medium.com/@abhishekjainindore24/data-augmentation-00c72f5f4c54*](https://medium.com/@abhishekjainindore24/data-augmentation-00c72f5f4c54)