Comparative Analysis of Deep Learning Models for Time Series Forecasting of Bitcoin Prices

Minh Duc Hoang, Thuy Le, Tuan Nguyen

Abstract

In this study, we investigate the performance of various time series models for predicting the future values of Bitcoin. Specifically, we explore the effectiveness of Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Temporal Fusion Transformers (TFT), and Prophet models in predicting Bitcoin prices. The goal is to compare the accuracy of these models and identify the best-performing one for Bitcoin time series prediction. To achieve this, we collect historical Bitcoin price data and split it into training and testing sets. We then train each model on the training set and evaluate their performance on the testing set. The mean squared error (MSE) is recorded to compare the models. The results show that the Prophet outperforms the other models in terms of accuracy. This study contributes to the literature on Bitcoin price prediction and provides insights into the best-performing time series models for this task.

1 Introduction

1.1 Investigating the problem

Bitcoin, a decentralized digital currency, has attracted significant attention from investors and researchers due to its high volatility and potential for significant price fluctuations. Accurately predicting the price of Bitcoin is a challenging task due to its complex nature and sensitivity to various economic and social factors. Time series models have been widely used for Bitcoin price prediction due to their ability to capture the temporal dependencies in the data. However, there is still no consensus on the best-performing model for Bitcoin price prediction. Therefore, in this project, we explore the performance of various time series models, including RNN, GRU, LSTM, TFT, and Prophet, for Bitcoin price prediction. By comparing the accuracy of these models, we aim to identify the best-performing one for this task. The results of this study will provide valuable insights into the effectiveness of different time series models for Bitcoin price prediction and contribute to the literature on cryptocurrency forecasting.

1.2 Dataset

For this project, we utilized a dataset obtained from Yahoo Finance containing Bitcoin price data from July 2010 to April 2019. The dataset consists of various columns, including Date, Open, High, Low, Close*, Adj Close**, and Volume.

The Date column represents the specific date on which the price data was recorded. The Open, High, Low, and Close* columns represent the opening, highest, lowest, and closing prices of Bitcoin for the given date, respectively. The Adj Close** column represents the adjusted closing price of Bitcoin for the given date, which takes into account any corporate actions, such as dividends or stock splits. Finally, the Volume column represents the number of Bitcoin units traded on the given date.

The dataset provides a comprehensive and extensive record of Bitcoin price movements over almost a decade, making it a valuable resource for investigating the effectiveness of various time series models for Bitcoin price prediction.

In this project, we split the dataset into training and test sets to evaluate the performance of various time series models for Bitcoin price prediction. We used the last 300 days of the dataset for the test set, and the remaining data was used for training.

Preprint. Under review.

2 Related works

Bitcoin price prediction has been a widely researched topic in recent years, and various approaches have been proposed to address this problem. In this section, we discuss some of the related works in this field.

Sequential deep learning models have been widely used for Bitcoin price prediction due to their ability to capture temporal dependencies in the data. Many different studies have focused on using GRU, RNN, and LSTM for cryptocurrency price prediction due to its ability to capture long-term dependencies in the time series data. For instance, [9], [11], and [5] experimented with these deep learning models to predict the price of Bitcoin and achieved improved performance compared to other traditional methods.

On the other hand, ARIMA is a mathematical model that has been widely used in finance and economics, including cryptocurrency prediction. For example, [4] used ARIMA to predict the prices of Bitcoin in short-term with notable performance. We look to use it both independently and in combination with our sequential deep learning model for a hybrid approach

Temporal Fusion Transformers (TFTs) by Lim et al. (2021) [6] is a recently proposed deep learning architecture for time series forecasting. TFTs have been shown to outperform traditional time series models such as ARIMA and LSTM in various domains. In the context of Bitcoin time series prediction, TFTs can be a promising approach for accurate and interpretable forecasting of Bitcoin prices.

Prophet is a time series forecasting model developed by Facebook [10] that is designed to handle time series data with multiple seasonalities. Prophet has been applied to various time series forecasting tasks and has shown promising results. In the context of Bitcoin prediction, Prophet has been used to forecast the cryptocurrency's price movements with a reasonable degree of accuracy.

Overall, these studies demonstrate the effectiveness of deep learning-based models for Bitcoin price prediction, with TFT and LSTM being the most widely used models in recent research.

3 Methods and Algorithms

3.1 RNN

Recurrent Neural Network (RNN)[7] is a type of neural network designed to process sequential data by incorporating feedback connections that allow the network to remember previous inputs. The RNN algorithm allows previous outputs to be used as inputs while having hidden states, in which for each timestep t, the activation $a^{<t>}$ and the output y are expressed as follows:

$$a^{<t>} = tanh(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$y = tanh(W_{ya}a^{<n>} + b_y)$$

where W_{ax}, W_{aa}, W_{ya}, b_y, b_a are coefficient.

For this experiment, we are using the vanilla RNN model, which is the multi-layer Elman RNN.

3.2 GRU

Gated Recurrent Unit (GRU)[3] is a type of recurrent neural network (RNN) with a gating mechanism that allows the network to selectively update its hidden state and control the flow of information through the network. The GRU algorithm uses two gates, a reset gate, and an update gate, to control the information flow through the network. The cell state $c^{<t>}$ and hidden state $c^{<t>}$ for each time step and final output are updated as follows:

$$c^{<t>} = (1 - z^{<t>}) \odot c^{<t-1>} + z^{<t>} \odot \tanh(W_c x^{<t>} + r^{<t>} \odot (U_c h^{<t-1>} + b_c))$$

$$h^{<t>} = (1 - z^{<t>}) \odot h^{<t-1>} + z^{<t>} \odot \tanh(W_h x^{<t>} + r^{<t>} \odot (U_h h^{<t-1>} + b_h))$$

$$y = tanh(W_y h_n + b_y)$$

where W_y, W_h, U_h, and b_h, W_c, U_c, and b_c are learnable parameters.

3.3 LSTM

Long Short-Term Memory (LSTM) [8] is a type of recurrent neural network (RNN) that has gained significant popularity in time series forecasting. The LSTM architecture is composed of multiple memory cells that can selectively store or discard information using gating mechanisms, including the forget gate, input gate, and output gate. The cell state $c^{<t>}$ and hidden state $c^{<t>}$ for each time step and final output are updated as follows:

$$c^{<t>} = f^{<t>} \odot c^{<t-1>} + i^{<t>} \odot \tanh(W_c x^{<t>} + U_c h^{<t-1>} + b_c)$$

$$h^{<t>} = sigmoid(W_o x^{<t>} + U_o h^{<t-1>} + b_o) \odot \tanh(c^{<t>})$$

$$y = tanh(W_y h_n + b_y)$$

where W_y, W_o, U_o, ,b_o, W_c, U_c, and b_c are learnable parameters.

3.4 ARIMA

Autoregressive Integrated Moving Average (ARIMA) [2] is a widely used time series forecasting method that incorporates both past values and forecast errors to generate predictions. ARIMA models are composed of three parts: autoregression (AR), differencing (I), and moving average (MA). The AR component uses past observations to predict future values, while the MA component models the error term as a linear combination of past errors. The differencing component is used to transform non-stationary time series data into stationary data by subtracting the previous observation from the current one. In this project, ARIMA is used both independently and in hybrid with LSTM. In the Hybrid model, each prediction weighted average of the forecasts from the two models based on the training performance with the default value being the simple average.

3.5 Prophet

Prophet [10] is a time series forecasting model that utilizes an additive regression model with a Fourier series to capture periodicity in the data. The model consists of four main components: trend, seasonality, holidays, and error.

The trend component models the underlying growth rate of the time series, which can be linear or logistic. The seasonality component captures the periodic fluctuations in the data and is modeled using a Fourier series. The holidays component accounts for any significant events that may affect the time series, such as public holidays or major news events. Finally, the error component captures the noise and any irregularities in the data that are not accounted for by the other components.

Prophet uses a Bayesian approach to model fitting, which allows for uncertainty estimation in the forecasted values.

The equation for prophet can be written as: $y(t) = g(t) + s(t) + h(t) + e(t)$

where:

$y(t)$: The observed value of the time series at time t.
$g(t)$: The trend component at time t (this is modelled as a linear regression function).
$s(t)$: The seasonality component at time t.
$h(t)$: The holidays component at time t.
$e(t)$: The error term at time t, which represents the difference between the observed value and the predicted value.

3.6 Temporal Fusion Transformer

Temporal Fusion Transformers (TFT)[6] is a state-of-the-art deep learning model that was developed for multi-horizon time series forecasting. Unlike traditional time series models, TFT is designed to capture both the temporal and cross-sectional dependencies in the data, which makes it particularly well-suited for predicting complex time series with multiple variables.

TFT consists of a stack of several layers of self-attention and feed-forward neural networks. The self-attention mechanism allows TFT to capture the dependencies between different time steps and variables, while the feed-forward neural networks enable it to learn complex non-linear relationships between the input variables and the target variable. The TFT architecture is shown in details in Fig 1.

In this project, we used the open-source implementation of TFT provided by the package *pytorch-forecasting*[1] and fine-tuned the model using the training set of our Bitcoin dataset.

4 Experiments

It is important to note that all the results presented were obtained by fitting the models using the Open price and Closed price (of the previous 30 days) as input to predict the Close price of the next 30 days. We measure the average mean squared error of the predicted close prices of the 240 30-day windows in the test set

With autoregressive models such as GRU, LSTM, RNN, ARIMA, the model output the next day per prediction then extrapolate the next 30 days. For the TFT models, the input is a 30-day window of open prices and the output is the predicted close price for the 31st day. On the other hand, for the Prophet model, the opening price of a particular day is given as input to predict its closing price. These approaches is commonly used in financial time series prediction, as these features are typically highly correlated with the target variable and can provide valuable information for predicting future prices.

5 Results

Results are shown in Table 1.

Table 1: Bitcoin price prediction results

Model	MSE
Prophet	9311.6991
TFT (4 heads, 160 hidden size)	424,713.2500
TFT (4 heads, 256 hidden size)	557,603.5625
LSTM (1 layer, 32 hidden size)	231,560.39
LSTM (3 layers, 32 hidden size)	242,978.2345
ARIMA (2,1,1)	527,619.5819
Hybrid LSTM (1 layer, 32 hidden size) + ARIMA (2,1,1)	353,134.702
RNN (1 layer, 16 hidden size)	159,507.22
RNN (8 layers, 64 hidden size)	201,153.27
GRU (1 layer, 32 hidden size)	202,163.2
GRU (8 layers, 64 hidden size)	212,634.02

6 Conclusions and Future Work

In this project, we compared the performance of several time series models, including Prophet, LSTM, ARIMA, and Temporal Fusion Transformers (TFT), for predicting the closing price of Bitcoin using data from Yahoo Finance. We found that Prophet had the best performance, with the lowest mean squared error (MSE).

Overall, our results suggest that machine learning models can be effective in predicting Bitcoin prices, with Prophet being the best-performing model in this study. However, we note that there is still room for improvement and further research in this area, especially for the remaining models. For example, future work could explore the use of more complex models or incorporate additional features beyond the ones used in this study, such as social media sentiment or news articles related to Bitcoin.Additionally, it should be noted that due to computational limitations and time constraints, hyperparameter tuning for each model was not thoroughly conducted. Therefore, there is potential for further optimization and improvement of the models with more extensive hyperparameter tuning.

It is also worth noting that the cryptocurrency market is highly volatile and subject to external factors such as regulatory changes, technological advances, and global events, which can have a significant impact on price movements. Therefore, it is important to consider the limitations and potential risks associated with cryptocurrency investment and trading before making any decisions based on price predictions.

References

[1] Jan Beitner. *PyTorch Forecasting*. https://github.com/jdb78/pytorch-forecasting. Accessed: April 17, 2023. 2021.

[2] G.E.P. Box and G.M Jenkins. "Time series analysis: forecasting and control". In: (1970).

[3] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://aclanthology.org/D14-1179.

[4] T. Widiyaningtyas I. M. Wirawan and M. M. Hasan. "Short Term Prediction on Bitcoin Price Using ARIMA Method". In: *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*. IEEE. 2019, pp. 260–265.

[5] Jordan Jones and Doga Demirel. "Long Short-Term Memory for Bitcoin Price Prediction". In: *2022 ACM International Conference on Information System and Data Mining (ICISDM)*. ACM. 2022, pp. 25–30.

[6] Bryan Lim et al. "Temporal fusion transformers for interpretable multi-horizon time series forecasting". In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764.

[7] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[8] Sepp Hochreiter & Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural Computation* (1997), pp. 1735–1780.

[9] Pindza E. Seabe PL Moutsinga CRB. "Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach". In: (2023), p. 2023.

[10] Sean J Taylor and Benjamin Letham. "Forecasting at scale". In: *The American Statistician* 72.1 (2018), pp. 37–45.

[11] Chih-Hung Wu et al. "A new forecasting framework for bitcoin price with LSTM". In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 168–175.

A Appendix

A.1 Contributions of each team member

LSTM,ARIMA, Hybrid model training and results: Minh Duc Hoang
RNN,GRU model training and results: Thuy Le
Prophet, TFT model training and results: Tuan Nguyen

The project report is written with contributions from all team members.

A.2 Link to the code base of the project

https://github.com/minhduchoang301/CSC413_Project

A.3 Graph and Figures

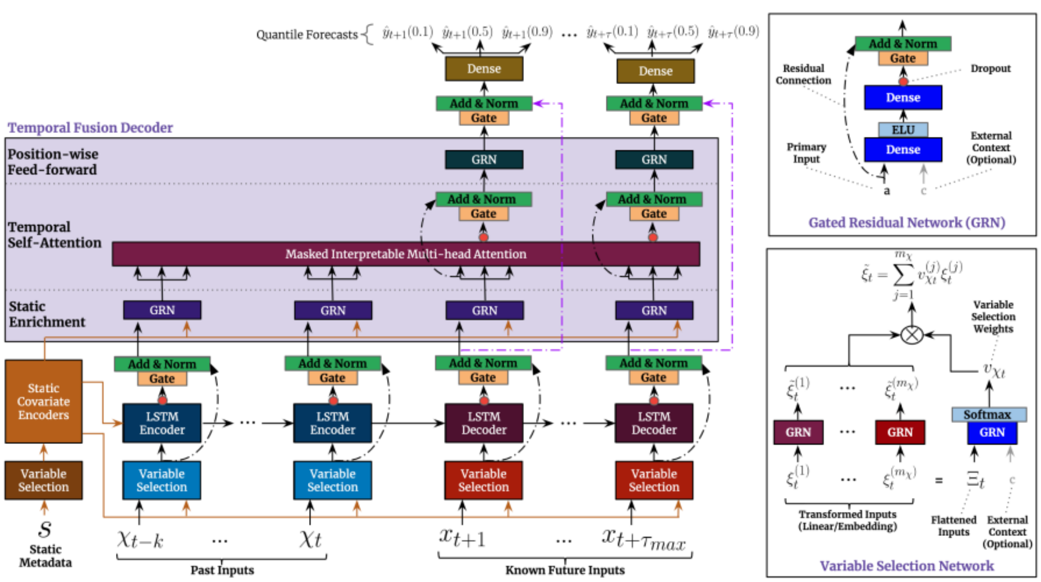

Figure 1: TFT architecture. TFT inputs static metadata, time-varying past inputs and timevarying a priori known future inputs. Variable Selection is used for judicious selection of the most salient features based on the input. Gated Residual Network blocks enable efficient information flow with skip connections and gating layers. Time-dependent processing is based on LSTMs for local processing, and multi-head attention for integrating information from any time step.[6]

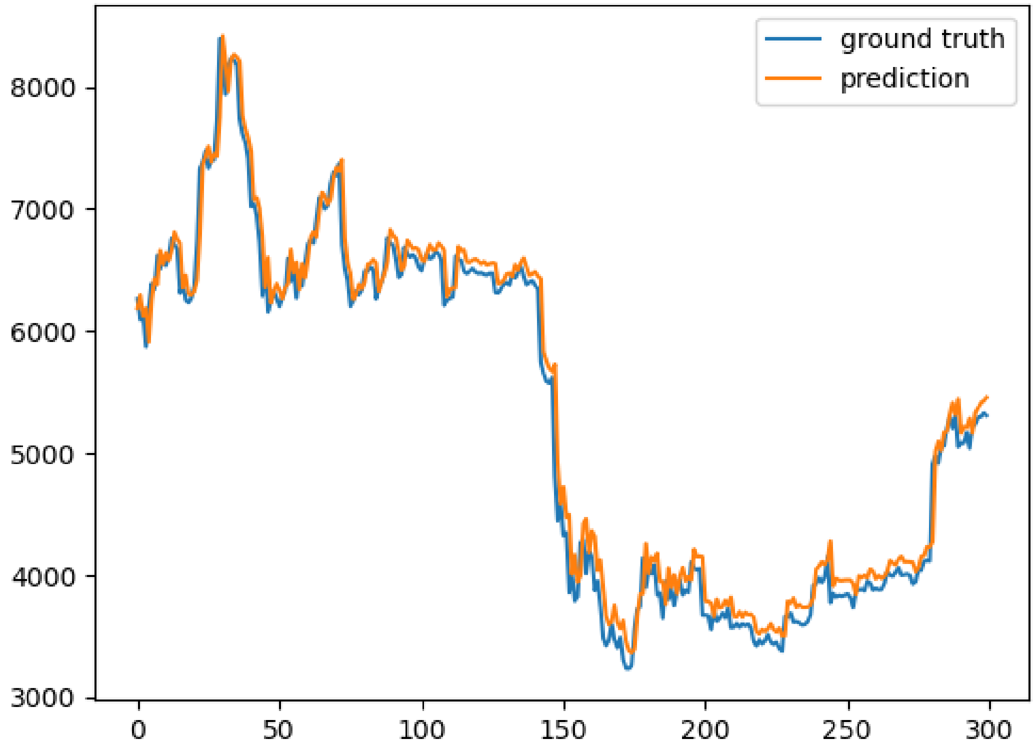

Figure 2: Prophet predictions on the test set of 300 last days.