A combined recurrent neural network model for cryptocurrency time series forecasting

Wiem Ben Romdhane[a], Heni Boubaker[a]

[a]LaREMFiQ, IHEC Sousse, Tunisia.

ARTICLE HISTORY
Compiled March 9, 2025

ABSTRACT
Forecasting cryptocurrency time series presents a difficult challenge, primarily attributable to their inherent non-linearity and chaotic behavior. While traditional statistical methodologies have yielded notable efficacy in specific domains, such as directional market prediction and individual equity price forecasting, the advent of neural networks and recurrent neural networks has catalyzed the exploration of novel paradigms in financial time series prediction. Furthermore, contemporary research posits that the synergistic integration of statistical and machine learning techniques can yield enhanced predictive accuracy relative to their isolated application. This study, therefore, proposes a combined framework that integrates statistical features derived from financial time series with a recurrent neural networks architecture for temporal forecasting. The efficacy of this methodology was evaluated through the prediction of Bitcoin (BTC) closing prices, utilizing a suite of performance metrics. Empirical results demonstrate the superiority of our model compared to univariate statistical and machine learning models.

KEYWORDS
Forecasting; BTC; Time series; RNN; LSTM; GRU; Combined

1. Introduction

The rapid growth of cryptocurrencies over the past decade has transformed them into a significant component of the global financial system. However, the volatile and non-linear nature of cryptocurrency prices presents unique challenges for accurate forecasting. Traditional statistical methods often fall short in capturing the intricate temporal dependencies and high volatility of cryptocurrency time series. As a result, researchers have increasingly turned to deep learning models, particularly Recurrent Neural Networks (RNNs), to address these challenges (Nasirtafreshi 2022; Kumar T et al. 2023; Seabe, Moutsinga, and Pindza 2023) .

Recurrent Neural Networks (RNNs) are well-suited for time series forecasting due to their ability to model sequential data and capture temporal dependencies. Variants of RNNs, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have been widely adopted for cryptocurrency forecasting. These models are particularly effective in handling long-term dependencies and mitigating the vanishing gradient problem, which is common in traditional RNNs. For instance, Nasirtafreshi (2022) proposed an RNN-LSTM model to predict cryptocurrency prices, demonstrating its ability to outperform traditional methods in terms of accuracy.

To further enhance the predictive performance of RNNs, researchers have explored hybrid models that combine RNNs with other techniques. For example, Guo et al. (2021) developed a hybrid method that integrates a multiscale residual block with an LSTM network to forecast Bitcoin prices. This approach leverages the strengths of both components: the residual block captures multi-scale features, while the LSTM network models temporal dependencies.

Similarly, other studies have proposed combining RNNs with convolutional layers or graph-based methods to improve the model's ability to capture spatial and temporal patterns in cryptocurrency data. The combined RNN models offer several advantages over standalone RNNs or traditional methods. By integrating complementary techniques, these models can better handle the non-stationarity and noise inherent in cryptocurrency time series. For instance, hybrid models that incorporate wavelet transforms or attention mechanisms have been shown to improve feature extraction and focus on relevant patterns in the data. Additionally, ensemble approaches that aggregate predictions from multiple RNN-based models have demonstrated enhanced robustness and accuracy in cryptocurrency forecasting.

2. Related works

Machine Learning is an artificial intelligence device that uses beyond information to predict the future. In simple words, we can expect the future price movements of cryptocurrencies to some extent by training a machine learning model using their past price data. Some recent studies have shown that machine learning based methods have many advantages of using traditional forecasting models, such as the ability to produce results that are approximately equal or identical to the actual outcome, while also improving the precision of the outcome (Hitam et al. 2021). Decision trees, support vector machine, and neural networks are some of the different machine learning methods that can be used for this purpose. As evidenced by the authors in (Andrianto and Diputra 2017), the inclusion of cryptocurrencies in multi-asset portfolios significantly improves portfolios in several ways. To start, it will enhance the portfolio's minimal variance and furthermore transfers the green frontier to a higher location.

Several research studies in the literature that using machine learning algorithms in BTC price forecasting achieve encouraging results. According to a study (Hitam et al. 2021), machine learning algorithms were applied to perform the price prediction of many currencies including BTC, ETH, LTC, XRP, and Stellar. According to the researchers, the SVM model was able to beat other machine learning models in terms of predicted values as well as accuracy. (Saad et al. 2019) used a variety of variables, carefully choosing the most accurate predictors using correlation analysis. The results showed that linear regression performed better than the other approaches when SVM, linear regression, and random forest (RF) were used to these selected features. The authors also tried predicting the prices of BTC and ETH using LSTM, a specific kind of deep learning, and discovered that LSTM had the lowest prediction error for BTC. To predict the prices of nine different cryptocurrencies, (Chowdhury et al. 2020) studied the application of machine learning-based ensemble methods, namely ANN, KNN, Gradient Boosted Trees and an ensemble model made up of multiple methods. The ensemble learning model had the the lowest error in the predictions. (Derbentsev et al. 2021) studies the difficulties faced when forecasting short-term cryptocurrency time series using supervised machine learning (ML). The ensemble methods were then applied to the daily closing prices of Bitcoin (BTC), Ethereum (ETH), and Ripple (XRP) using historical prices and technical indicators such as moving averages as features: Random Forest (RF) and Stochastic Gradient Boosting Machine (SGBM). The results showed that ML ensemble methods held promise, with SGBM and RF yielding good accuracy on short-term predictions. (Chen et al. 2021) introduces a two-stage approach to investigate whether economic and technology determinants can accurately predict the Bitcoin exchange rate. In the first stage, artificial neural networks and random forests are employed as nonlinear feature selection methods to identify and prioritize key predictors from economic and technology factors. In the second stage, these selected predictors are integrated into a long short-term memory (LSTM) model to forecast the Bitcoin exchange rate without relying on historical exchange rate data. The results demonstrate that LSTM, utilizing economic and technology determinants, outperforms traditional models such as autoregressive integrated moving average, support vector regression, adaptive network fuzzy inference system, and LSTM models that depend on past exchange rates. This highlights that economic and technology factors are more significant for predicting Bitcoin exchange rates than historical exchange rate information. (Patel et al. 2020) introduce a hybrid prediction model for cryptocurrencies, combining LSTM and GRU networks, specifically applied to Litecoin and Monero. The results show that our approach achieves highly accurate price predictions, demonstrating its potential for broader use in forecasting the prices of various cryptocurrencies. This suggests that the model could be a valuable tool for understanding and predicting cryptocurrency market trends. (Zhang et al. 2021) propose a model that is built around three key modules designed to enhance cryptocurrency price prediction. First, the Attentive Memory module integrates a Gated Recurrent Unit (GRU) with a self-attention mechanism, allowing the model to focus on the most relevant parts of each input sequence. Second, the Channel-wise Weighting module analyzes the prices of major cryptocurrencies, learning their relationships by dynamically adjusting the importance of each sequence. Finally, the Convolution & Pooling module captures local patterns in the data, improving the model's ability to generalize. To test its effectiveness, the authors ran a series of experiments, which demonstrated that their model outperforms existing baseline methods in terms of prediction error, accuracy, and profitability, achieving state-of-the-art results. (Nouira, Bouchakwa, and Amara 2024) explore recent advancements in machine learning and deep learning techniques used for predicting cryptocurrency prices, as highlighted in highly regarded publications. By focusing on the latest research, the authors analyze how these models leverage social network data to forecast cryptocurrency trends. The findings reveal a notable shift in the field, with deep learning methods increasingly surpassing traditional machine learning approaches in cryptocurrency price prediction. The study of (Viéitez, Santos, and Naranjo 2024) develops prediction models for Ethereum (ETH) prices using Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) networks, alongside a Support Vector Machine (SVM) for trend classification. Sentiment analysis is incorporated to assess its impact, though it shows minimal influence on results. Two innovative knowledge-based investment strategies are designed and tested across different time periods using real market data. The findings reveal that these models can achieve a profit factor of up to 5.16 with limited trading activity, demonstrating their reliability and potential for generating returns in the current cryptocurrency market. (Lapitskaya, Eratalay, and Sharma 2024) combine technical analysis tools—such as Moving Average Convergence/Divergence, Commodity Channel Index, and Relative Strength Index—with the eXtreme Gradient Boosting (XGBoost) model to predict cryptocurrency prices and returns. Using historical daily data for Bitcoin, Ether, Golem, and FUNToken, the model achieves strong accuracy and reliable performance across both training and test datasets. This approach highlights the potential of integrating traditional indicators with machine learning for effective cryptocurrency price forecasting. Recent work of (Shirwaikar et al. 2025) focuses on predicting the values of three cryptocurrencies—Litecoin, Ethereum, and Monero—using deep learning techniques, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). To enhance accuracy, the research incorporates the Direction Algorithm (DA), which analyzes Bitcoin's price movements as a benchmark for other cryptocurrencies, and Change Point Detection (CPD) using Pruned Exact Linear Time (PELT) to identify sudden price shifts. Additionally, sentiment analysis of news data is performed using the Vader algorithm to further improve forecasting precision. The study of (Hafidi et al. 2025) presents a novel method that automates hyperparameter tuning, boosting accuracy by analyzing intricate relationships between cryptocurrencies. Using advanced deep learning tools like RNNs and LSTMs, combined with Genetic Algorithms (GA), the approach optimizes model performance. Two architectures, LAO and LOEE, are introduced and compared to tackle the volatility and complexity of cryptocurrency markets. By streamlining hyperparameter selection and exploring cryptocurrency interconnections, it delivers a reliable solution for predicting prices in a fast-changing market.

3. Methodology

In this section, we outline the steps taken during the research and modeling phases. We then present the results of our predictive framework for BTC price, followed by a comprehensive evaluation of performance and research insights.

The aim of this study is to leverage advanced deep learning techniques — such as LSTM, GRU, and LSTM-GRU - to forecast the prices of BTC. To achieve this, we follow a structured process: (1) collecting historical price data for BTC; (2) visualizing the data to uncover patterns; (3) splitting the data into training and testing sets; (4) training the three deep learning models; (5) validating the models; and (6) comparing the performance of each method to identify the most effective approach.

3.1. *Dataset*

For this study, the main emphasis is on analyzing the historical closing price data of the BTC close price. This data is obtained from Yahoo Finance. The closing price is a key indicator in financial time series analysis, as it plays a significant role in forecasting and understanding market trends. The closing price is a vital piece of information widely utilized by investors, analysts, and traders to gauge market trends and make well-informed decisions. It plays a central role in many technical analysis methods and is frequently used to compute moving averages, oscillators, and other financial metrics. These tools, in turn, help shape trading strategies and guide decisions related to portfolio management. Essentially, the closing price acts as a essential element for understanding market behavior and planning investment actions.

The data collection process gathers as much historical closing price data as possible for the BTC close price spanning from January, 06, 2016 to February, 20, 2025. The Figure 1 represents the graph of Bitcoin's (BTC) closing price typically illustrates the cryptocurrency's price movements over a specific period, showcasing its volatility and trends. Data summary table for BTC is given in **??**.

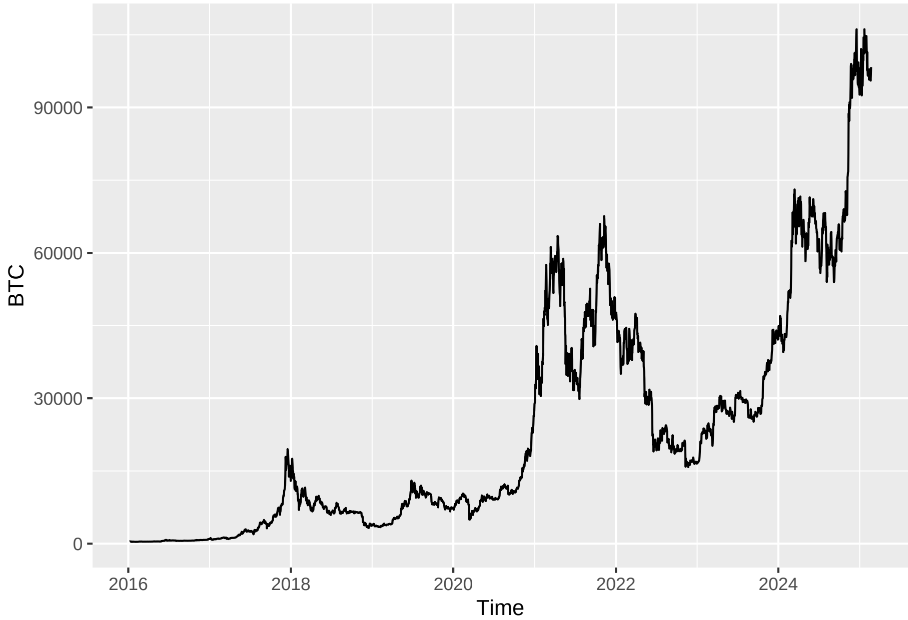

Figure 1. BTC close price series

3.2. *Pre-processing techniques*

To prepare the BTC close price data for deep learning analysis, we applied several pre-processing steps. First, we reshaped the data to make it compatible with advanced neural network models like LSTM, GRU, and LSTM-GRU.

Then, we split the data into training and testing sets using an 90:10 ratio. This approach maintained the continuity of features for the BTC price, ensuring that the models could learn from a coherent dataset while being evaluated on a separate, unseen portion. These steps collectively ensured the data was well-prepared for robust and reliable deep learning analysis.

Finally, normalization was a crucial step to ensure the model's accuracy and prevent bias. To handle the issue of variables with different scales, we applied feature-wise normalization using MinMax Scaling, which recent studies have shown to significantly improve model performance (Ahsan et al. 2021). This technique helped standardize the data, making it easier for the models to learn patterns effectively.

Table 1.: Descriptive statistics of BTC close price

	All data	Training data	Test data
nbr.val	3334.000	3000.000	334.000
min	364.331	364.331	53948.752
max	106146.263	73083.501	106146.263
range	105781.932	72719.170	52197.511
sum	78528495.741	53657679.885	24870815.857
median	11596.294	10064.596	67383.843
mean	23553.838	17885.893	74463.521
SE.mean	416.957	313.522	876.722

	All data	Training data	Test data
std.dev	24075.429	17172.292	16022.686
coef.var	1.022	0.960	0.215
skewness	1.242	1.005	0.722
kurtosis	0.945	-0.027	-1.151

3.3. *Recurrent neural networks*

Recurrent Neural Networks (RNNs) have the unique ability to understand complex, short-term relationships in time series data. There are two main types of RNNs: fully connected and partially connected. The first RNN was created by (Williams and Zipser 1989) in the late 1980s, during a period when interest in neural network architectures was on the rise, leading to many significant advancements in this field. The second type of RNN is introduced in 2014 by (Chung et al. 2014).

3.3.1. LSTM model

Long Short Term Memory networks, often referred to as LSTMs, are a specific type of Recurrent Neural Network (RNN) designed to learning long-term dependencies in data. They were first introduced by (Hochreiter and Schmidhuber 1997) and have since been further developed and embraced by many researchers in the field. LSTMs have proven to be highly effective across a wide range of applications, which has contributed to their popularity today. LSTM are an evolved form of traditional RNNs. They are specifically engineered to manage the challenges that arise with long-term dependencies, effectively addressing the vanishing gradient issue by incorporating a mechanism that helps retain information over extended periods (Hochreiter and Schmidhuber 1997). Essentially, the LSTM architecture consists of several interconnected memory blocks that function as recurrent sub-networks. These memory blocks play a crucial role in both preserving the network's state over time and controlling the flow of information between the cells. In Figure 2[1], you can see the structure of an LSTM block, which includes the input signal x_t, the output h_t, and the activation function. The input gate is key in deciding which pieces of information should be stored in the cell state, while the output gate determines what information should be released from the cell state. The forward training process of an LSTM network can be outlined with the following equations:

$$i_t = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right) \tag{1}$$

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_t\right) \tag{2}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(W_c \odot [h_{t-1}, x_t] + b_c\right) \tag{3}$$

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right) \tag{4}$$

$$h_t = o_t \odot \tanh(c_t) \tag{5}$$

[1]Source: (Kılıç and Ömür Uğur 2023)

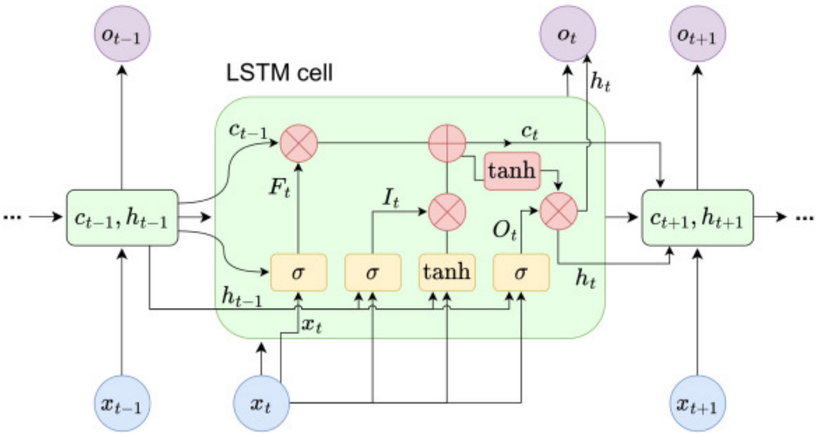

Figure 2. Structure of LSTM cell

3.4. *Gated Recurrent Units (GRU)*

Gated Recurrent Units (GRU) are a type of Recurrent Neural Network (RNN) introduced by (Chung et al. 2014) as a more efficient alternative to traditional LSTM networks. Similar to LSTM, GRU are capable of handling input sequences of varying lengths and maintaining a state that captures past information. However, GRU simplify the architecture by using only two gates; an update gate to determine what information to keep and a reset gate to decide what information to forget. This streamlined design makes GRU less complex and faster to train compared to LSTM, which rely on multiple gates and an internal memory cell. Despite their simplicity, GRU often deliver comparable performance to LSTM across many tasks (Yang, Yu, and Zhou 2020), making them a popular choice for sequence modeling.

Figure 3[2] represent the diagram of GRU cell. The hidden state at time t, denoted as h_t, is computed using the input at time t, x_t, and the previous hidden state, h_{t-1}, through the following equations (Dey and Salem 2017):

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{6}$$

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{7}$$

$$\widehat{h}_t = \phi\left(W_h x_t + U_h (r_t \odot h_{t-1}) + b_r\right) \tag{8}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widehat{h}_t \tag{9}$$

3.5. *Hyperparameter tuning*

Hyperparameter tuning using random search optimization for LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) models in time series prediction involves systematically exploring a predefined hyperparameter space to identify the optimal configuration that maximizes model performance. Key hyperparameters include the number of layers, hidden units, learning rate, batch size, dropout rate, and sequence length. Unlike grid search, random search randomly samples combinations of these hyperparameters from the specified ranges, which is more computationally efficient and often yields competitive results. For LSTM and GRU models, the process begins by defining the search space for each hyperparameter, followed by training

[2]Source: (Vasilev 2019)

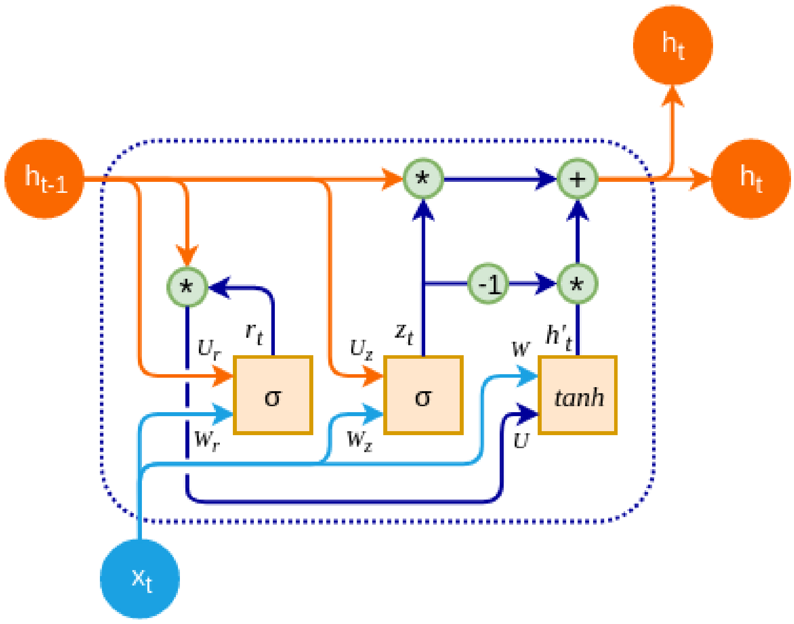

Figure 3. Structure of GRU cell

and evaluating multiple model instances with different configurations using a validation set or cross-validation. Performance metrics such as Mean Squared Error (MSE) or Mean Absolute Error (MAE) are used to assess the models. The best-performing configuration is then selected for final training and testing time series data. This approach balances exploration and computational cost, making it suitable for optimizing complex recurrent neural networks like LSTM and GRU.

Table 2.: Hyperparameter space

Hyperparameter	Range/Values
Number of layers	1 to 3 (integer)
Number of units	32 to 128 (Increments of 32)
Learning rate	0.0001, to 0.01 (Logarithmic scale)
Dropout rate	0.1 to 0.5 (Increments of 0.1)
Dense activation	[‘relu’,‘sigmoid’]

Table 3.: Best Hyperparameters

Hyperparameter	LSTM	GRU	LSTM-GRU
Number of layers	1		
Number of units	96		
Learning rate	0.0003		
Dropout rate	0.1		

Hyperparameter	LSTM	GRU	LSTM-GRU
Dense activation	relu		

3.6. *Performance Metrics*

To assess how well the proposed RNN algorithms performed, we relied on two key metrics: root mean squared error (RMSE) and mean absolute error percentage (MAPE). These metrics help us evaluate the accuracy of the prediction models—the lower the RMSE and MAPE values, the more accurate and reliable the model's predictions are. Essentially, smaller values indicate better performance.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(a_i - p_i)^2}{n}} \tag{10}$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{|a_i - p_i|}{ai} \tag{11}$$

Where p_i represents the predicted value, a_i denotes the actual value, and n stands for the total number of time steps.

4. Results

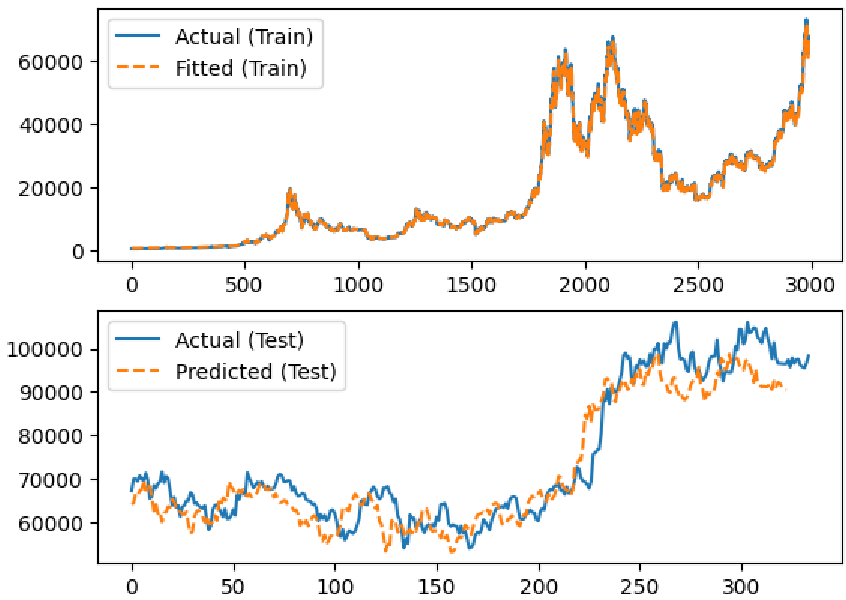

Figure 4. Actual and predicted results using the LSTM model.

Table 4.: Performance results for the LSTM model

	Train data	Test data
RMSE	968.292	4298.384
MAPE	7.210	4.170

Acknowledgement(s)

Disclosure statement

Funding

Notes on contributor(s)

Nomenclature/Notation

Notes

References

Ahsan, Md Manjurul, MA Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta, and Zahed Siddique. 2021. "Effect of data scaling methods on machine learning algorithms and model performance." *Technologies* 9 (3): 52.

Andrianto, Yanuar, and Yoda Diputra. 2017. "The effect of cryptocurrency on investment portfolio effectiveness." *Journal of finance and accounting* 5 (6): 229–238.

Chen, Wei, Huilin Xu, Lifen Jia, and Ying Gao. 2021. "Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants." *International Journal of Forecasting* 37 (1): 28–43.

Chowdhury, Reaz, M Arifur Rahman, M Sohel Rahman, and MRC Mahdy. 2020. "An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning." *Physica A: Statistical Mechanics and its Applications* 551: 124569.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* .

Derbentsev, V, V Babenko, KIRILL Khrustalev, S Khrustalova, and H Obruch. 2021. "Comparative performance of machine learning ensemble algorithms for forecasting cryptocurrency prices." *International Journal of Engineering, Transactions A: Basics* 34 (1): 140–148.

Dey, Rahul, and Fathi M Salem. 2017. "Gate-variants of gated recurrent unit (GRU) neural networks." In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, 1597–1600. IEEE.

Hafidi, Nasreddine, Zakaria Khoudi, Mourad Nachaoui, and Soufian Lyaqini. 2025. "Cryptocurrency Price Prediction with Genetic Algorithm-based Hyperparameter Optimization." *Statistics, Optimization & Information Computing* .

Hitam, Nor Azizah, Amelia Ritahani Ismail, Ruhaidah Samsudin, and Eman H Alkhammash. 2021. "The effect of Kernel functions on cryptocurrency prediction using support vector machines." In *International Conference of Reliable Information and Communication Technology*, 319–332. Springer.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Kumar T, Vijaya, S. Santhi, K. G. Shanthi, and Gokila M. 2023. "Cryptocurrency Price Prediction using LSTM and Recurrent Neural Networks." In *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 1–5.

Kılıç, Deniz Kenan, and Ömür Uğur. 2023. "Hybrid wavelet-neural network models for time series." *Applied Soft Computing* 144: 110469. https://doi.org/https://doi.org/10.1016/j.asoc.2023.110469.

Lapitskaya, Darya, M Hakan Eratalay, and Rajesh Sharma. 2024. "Prediction of Cryptocurrency Prices with the Momentum Indicators and Machine Learning." *Computational Economics* 1–19.

Nasirtafreshi, I. 2022. "Forecasting cryptocurrency prices using Recurrent Neural Network and Long Short-term Memory." *Data & Knowledge Engineering* 139: 102009. https://doi.org/https://doi.org/10.1016/j.datak.2022.102009.

Nouira, Ameni Youssfi, Mariam Bouchakwa, and Marwa Amara. 2024. "Role of social networks and machine learning techniques in cryptocurrency price prediction: a survey." *Social Network Analysis and Mining* 14 (1): 152.

Patel, Mohil Maheshkumar, Sudeep Tanwar, Rajesh Gupta, and Neeraj Kumar. 2020. "A deep learning-based cryptocurrency price prediction scheme for financial institutions." *Journal of information security and applications* 55: 102583.

Saad, Muhammad, Jinchun Choi, DaeHun Nyang, Joongheon Kim, and Aziz Mohaisen. 2019. "Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions." *IEEE Systems Journal* 14 (1): 321–332.

Seabe, Phumudzo Lloyd, Claude Rodrigue Bambe Moutsinga, and Edson Pindza. 2023. "Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach." *Fractal and Fractional* 7 (2). https://doi.org/10.3390/fractalfract7020203.

Shirwaikar, Rudresh, Sagar Naik, Abiya Pardeshi, Sailee Manjrekar, Yash Shetye, Siddhesh Dhargalkar, and Ritvik Madkaikar. 2025. "Optimized Deep Learning Framework for Cryptocurrency Price Prediction." *SN Computer Science* 6 (1): 67.

Vasilev, Ivan. 2019. *Advanced Deep Learning with Python: Design and implement advanced next-generation AI solutions using TensorFlow and PyTorch*. Packt Publishing. https://books.google.tn/books?id=TxbEDwAAQBAJ.

Viéitez, Adrián, Matilde Santos, and Rodrigo Naranjo. 2024. "Machine learning Ethereum cryptocurrency prediction and knowledge-based investment strategies." *Knowledge-Based Systems* 299: 112088.

Williams, Ronald J, and David Zipser. 1989. "A learning algorithm for continually running fully recurrent neural networks." *Neural computation* 1 (2): 270–280.

Yang, Shudong, Xueying Yu, and Ying Zhou. 2020. "Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example." In *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, 98–101. IEEE.

Zhang, Zhuorui, Hong-Ning Dai, Junhao Zhou, Subrota Kumar Mondal, Miguel Martínez García, and Hao Wang. 2021. "Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels." *Expert Systems with Applications* 183: 115378.

5. Appendices

Any appendices should be placed after the list of references, beginning with the command `\appendix` followed by the command `\section` for each appendix title, e.g.

```
\appendix
```

```
\section{This is the title of the first appendix}
\section{This is the title of the second appendix}
```

produces:

Appendix A. This is the title of the first appendix

Appendix B. This is the title of the second appendix

Subsections, equations, figures, tables, etc. within appendices will then be automatically numbered as appropriate. Some theorem-like environments may need to have their counters reset manually (e.g. if they are not numbered within sections in the main text). You can achieve this by using `\numberwithin{remark}{section}` (for example) just after the `\appendix` command.

Please note that if the `endfloat` package is used on a document containing appendices, the `\processdelayedfloats` command must be included immediately before the `\appendix` command in order to ensure that the floats in the main body of the text are numbered as such.

Appendix A. Troubleshooting

Authors may occasionally encounter problems with the preparation of a manuscript using LaTeX. The appropriate action to take will depend on the nature of the problem:

(i) If the problem is with LaTeX itself, rather than with the actual macros, please consult an appropriate LaTeX 2ε manual for initial advice. If the solution cannot be found, or if you suspect that the problem does lie with the macros, then please contact Taylor & Francis for assistance (`latex.helpdesk@tandf.co.uk`).

(ii) Problems with page make-up (e.g. occasional overlong lines of text; figures or tables appearing out of order): please do not try to fix these using 'hard' page make-up commands – the typesetter will deal with such problems. (You may, if you wish, draw attention to particular problems when submitting the final version of your manuscript.)

(iii) If a required font is not available on your system, allow TeX to substitute the font and specify which font is required in a covering letter accompanying your files.

Appendix B. Obtaining the template and class file

B.1. *Via the Taylor & Francis website*

This article template and the `interact` class file may be obtained via the 'Instructions for Authors' pages of selected Taylor & Francis journals.

Please note that the class file calls up the open-source LaTeX packages booktabs.sty, epsfig.sty and rotating.sty, which will, for convenience, unpack with the downloaded template and class file. The template calls for natbib.sty and subfigure.sty, which are also supplied for convenience.

B.2. *Via e-mail*

This article template, the `interact` class file and the associated open-source LaTeX packages are also available via e-mail. Requests should be addressed to `latex.helpdesk@tandf.co.uk`, clearly stating for which journal you require the template and class file.