ENTREPRENEURSHIP AND SUSTAINABILITY ISSUES
ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

Publisher
http://jssidoi.org/esc/home

CRYPTOCURRENCY PRICE FORECASTING: A COMPARATIVE ANALYSIS OF AUTOREGRESSIVE AND RECURRENT NEURAL NETWORK MODELS

Joana Katina [1,3], Igor Katin [2,3], Vera Komarova [4]

[1] Institute of Computer Science, Vilnius University, Didlaukio st. 47, 08303 Vilnius, Lithuania
[2] Institute of Data Science and Digital Technologies, Vilnius University, Akademijos st. 4, 08412 Vilnius, Lithuania
[3] Faculty of Electronics and Informatics, Higher Education Institution, J. Jasinskio st. 15, 01111 Vilnius, Lithuania
[4] Daugavpils University, Vienibas st. 13, Daugavpils, Latvia

E-mails: [1] joana.katina@mii.vu.lt; [2] igor.katin@mii.vu.lt; [3] vera.komarova@du.lv

Received 12 February 2024; accepted 8 May 2024; published 30 June 2024

Abstract. This article presents a novel approach to cryptocurrency price forecasting, leveraging advanced machine-learning techniques. By comparing traditional autoregressive models with recurrent neural network approaches, the study aims to evaluate the forecasting accuracy of Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models across various cryptocurrencies, including Bitcoin, Ethereum, Dogecoin, Polygon, and Toncoin. The data for this empirical study was sourced from historical prices of these specific cryptocurrencies, as recorded on the CoinMarketCap platform, covering January 2022 to April 2024. The methodology employed involves rigorous statistical and neural network modelling where each model's parameters were meticulously optimized for the specific characteristics of each cryptocurrency's price data. Performance metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) were used to assess the precision of each model. The main results indicate that LSTM and GRU models, leveraging deep learning techniques, generally outperformed the traditional ARIMA and SARIMA models regarding error metrics. This demonstrates a higher efficacy of neural networks in handling the non-linear complexities and volatile nature of cryptocurrency price movements. This study contributes to the ongoing discourse in financial technology by elucidating the practical implications of using advanced machine-learning techniques for economic forecasting. Importantly, it provides valuable insights that can directly inform and enhance the decision-making processes of investors and traders in digital assets.

Keywords: forecasting; prediction; cryptocurrencies; time series; ARIMA; SARIMA; RNN; LSTM; GRU.

Reference to this paper should be made as follows: Katina, J., Katin, I., Komarova, V. (2024). Cryptocurrency price forecasting: a comparative analysis of autoregressive and recurrent neural network models. *Entrepreneurship and Sustainability Issues*, 11(4), 425-436. http://doi.org/10.9770/jesi.2024.11.4(26)

JEL Classifications: C22, C32, C45, C53, G17

Additional disciplines: Informatics Engineering, Computer Science, Mathematics, Data Science

1. Introduction

The need to predict cryptocurrency prices is an important area of academic interest and active research by numerous researchers and practitioners worldwide (Au et al., 2024; Apostolakis, 2024; Bâra et al., 2024; Singh et al., 2024; Kapur et al., 2024, Demirel, 2024).

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

Indeed, as the market for cryptocurrencies advances with more complexity and scales of use, the necessity for even more precise forecasting models arises. This burgeoning field is dynamic, with established methods getting fine-tuned and polished while new, innovative techniques are constantly explored.

Traditional time series methods such as Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models are mainstream in financial forecasting due to their effectiveness in modelling and forecasting time-varying data. These models, with a focus on understanding the historical volatility and trends, are constantly refined, improved, and enhanced to capture better the sophisticated behaviours of cryptocurrency markets, which are known for massive volatility and unpredictability (Ng et al., 2023; Satheesh & Sundararagan, 2022; Vo & Ślepaczuk, 2022; Corrêa et al., 2016).

In parallel, there has been a significant shift toward applying the deep learning approach in forecasting cryptocurrency prices. Neural networks, in particular Long Short-Term Memory (LSTM) models and Gated Recurrent Units (GRU), have proven to be highly capable of processing sequences with long-range dependencies (Respaty et al., 2023; Nosouhian et al. 2021; Zarzycki & Ławryńczuk, 2021; Zhang et al., 2021). These deep learning models are experts in processing and learning the vast amount of unstructured data generated in the cryptocurrency markets, hence representing value much larger than that projected by traditional statistical models.

A current and relevant problem in this area is the comparative analysis of traditional time series methods with these new deep-learning approaches. Scientists continue researching which methodologies yield the most accurate, reliable, and appropriate forecasts (Bielskis & Belovas, 2022; Sehrawat & Vishwakarma, 2022; Gopu et al., 2023). This includes rigorous testing, validation, and performance evaluation of historical data models with different cryptocurrencies and market conditions. It is these comparisons, therefore, that becomes very critical in the sense that they help to gauge the strengths and weaknesses of each method as it guides traders and investors in making decisions.

However, notwithstanding advancement, cryptocurrency prediction is still a very problematic issue. In this case, the volatility of cryptocurrencies, which is influenced by news of regulation, market sentiment, or changes in technology, is volatile and, therefore, difficult to predict. This has led to the development of hybrid models that combine the soundness of time series analysis and the adaptiveness of machine learning and deep learning models (Li & Zhao, 2022; Jisha et al., 2023; Si, 2023). In addition, the inclusion of sentiment analysis and blockchain analysis in forecasting models is an interesting area, as it can provide more nuanced insights into future price developments.

The dynamic nature of the cryptocurrency market makes sure the area of price forecasting will continue to evolve. Continued innovations in AI and machine learning, along with improved understanding and incorporation of market-specific dynamics, promise improvements in the accuracy and effectiveness of the forecasting models. Such constant development benefits not only individual investors or traders but also allows the market of cryptocurrencies to become more stable and mature.

2. Research objective and methodology

In the volatile domain of cryptocurrency markets, forecasting price movements presents a formidable challenge due to rapid shifts in economic conditions and the inherently noisy nature of financial data. This research aims to conduct a comprehensive comparative analysis of traditional time series forecasting models, such as ARIMA and SARIMA, against advanced deep learning approaches, including LSTM and GRU networks, for predicting cryptocurrency prices. By examining the performance of these models on various cryptocurrencies, we aim to identify the most accurate, reliable, and appropriate forecasting methodologies for highly volatile and complex cryptocurrency markets.

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

2.1. Autoregressive models

ARIMA is a statistical analysis model that can handle non-stationary data and fits well within the mostly ambiguous market conditions experienced in cryptocurrencies. Meanwhile, the SARIMA model considers the effect of seasonality present in data, which is common in financial time series but often ignored by ARIMA models.

2.1.1. Auto-Regressive Integrated Moving Average (ARIMA)

The Auto-Regressive Integrated Moving Average (ARIMA) model was introduced by Box & Jenkins (1970) to analyze and forecast time series data. It is particularly efficient for non-stationary data and is prevalent in many real-world applications, such as stock prices, economic indicators, and cryptocurrency prices.

The model contains Auto-Regressive (AR), Integrated (I), and Moving Average (MA) components and, therefore, captures most structures and patterns present in the time series data. Auto-Regressive (AR) component assumes that the regressing variable of interest is regressed with some of its previous values. This component makes predictions because it specifies the future values as a function of past values. The integrated (I) component differentiates the observational data in the series to make the series stationary, meaning that the mean and variance of the series do not change over time. Stationarity is one of the statistical conditions. The Moving Average (MA) component models the error term as a linear combination of error terms happening simultaneously and different past lags. This combination of three components makes ARIMA capabilities a powerful tool in time series forecasting and, therefore, a reliable choice in scenarios where data has evidence for non-stationarity.

The ARIMA model is described by the notation ARIMA(p, d, q), where:

- p is the number of lag observations in the model (lag order).
- d is the degree of differencing (the number of past times values have been subtracted from the data).
- q is the size of the moving average window (order of moving average).

2.1.2. Seasonal Autoregressive Integrated Moving Average (SARIMA)

The Seasonal Autoregressive Integrated Moving Average (SARIMA) was introduced by Box & Jenkins (1976) in later revisions of their book. This model extends the standard ARIMA model to account for seasonal variations in time series data. The SARIMA model is highly recommended for datasets that show regular changes in the pattern at specified, definite intervals throughout the year and are ideal for forecasting seasonal economic, weather, and consumption data.

The SARIMA model is described by the notation SARIMA(p, d, q)(P, D, Q)s, where:

- p, d, and q are the non-seasonal parameters that describe the autoregressive, integrated, and moving average components of the model, as in a standard ARIMA model.
- P, D, and Q are the seasonal parameters that describe the autoregressive, integrated, and moving average components at the seasonal level.
- s denotes the length of the seasonal cycle, which could be quarterly, monthly, or weekly, depending on the context.

There is a need to explain what the P, D, and Q parameters are in more detail below:

- Seasonal Autoregressive Order (P): It reflects the use of the values of the previous seasonal periods to predict the future value. It is equivalent to the autoregressive component in an ARIMA model but applied on a seasonal scale.
- Seasonal Differencing Order (D): This would mean that the series is subjected to differencing at the seasonal period to ensure stationarity on the seasonal level, much like the integrated component in the ARIMA model.

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

- Seasonal Moving Average Order (*Q*): This indicates the moving averages of the lagged forecast errors used to predict the current value, applied in the same manner as a moving average component of the ARIMA model but considering seasonal data points.

By using a seasonal component, the SARIMA model can provide more accurate and relevant forecasts for seasonal data compared to non-seasonal models, which might fail to capture the cyclical nature of the data. It is a handy tool in the arsenal of any analyst dealing with time series data affected by seasonality.

2.2. Recurrent Neural Network Models

Recurrent neural networks (RNNs) are exceptional in their ability to process sequential data because they can remember past occurrences and correlate them with current data. This makes them entirely up to the task in many applications where context is crucial, like text generation and forecasting of time series, but still not that easy to work with. This leads to the adjustment of weights during training. In other words, it results in problems of gradient vanishing since the alterations done in the model weights are too minimal, which makes the training procedure ineffective. On the contrary, if massive updates take place, there could be a gradient explosion that triggers the problem of learning instability. Both these are caused by the incredible complexity of how training RNNs must be done. Fortunately, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are much more advanced variations of the RNN that have been architected to solve these very issues.

2.2.1. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) network, a pivotal innovation in the realm of neural networks, was first proposed by Hochreiter & Schmidhuber (1997). LSTM was designed to store information for long periods and retrieve it efficiently when needed. The idea was to subsume, within the architecture of the neural network itself, these so-called "gates" that would control the flow of information by giving explicit instructions on what to remember and what to forget.

LSTM consists of a series of memory cells that can store and update information over long time steps. Each memory cell has three types of gates: input gates, forget gates, and output gates. Input gates define which pieces of information will be considered worthy of storage in long-term firing. It thereby ensures that only the most relevant information is being treated as necessary. A forgetting gate in a cell decides when to forget information and when to keep information. Therefore, the output gates establish what the information in the current state of the cell is to be transmitted to the output of the network; the factors might be code of pertinent characteristics or characteristics of the feature that are significant at that time. In this way, the network can distribute information more accurately and efficiently. This would give the LSTM the power to handle much longer data sequences than the standard RNN.

2.2.2. Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) is another advancement in the field of RNNs, proposed by Cho et al. (2014). GRU was introduced as a simplified version of LSTM, designed to make the learning process more efficient while retaining the ability to capture dependencies in sequential data. GRU combines the hidden state and the cell state into a single state and uses gating units to control the information flow, similar to LSTM, but with fewer parameters. Therefore, GRU is less computationally intensive. This makes GRU faster to learn without a significant loss in performance.

GRU does not have a separate memory location like LSTM and it has only two gates: a reset gate and an update gate. The update gate helps the model decide how much of the previous hidden state should be retained. This is done by combining the past state and the new information. The reset gate determines which part of the past state is important for the current input, and the update gate uses this information to form the new hidden state. This allows the GRU to adjust the flow of information more flexibly than the LSTM. Also, GRU is easier to implement and requires fewer parameters.

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

2.3. Data

For this research, five cryptocurrencies were selected: Bitcoin (BTC), Ethereum (ETH), Dogecoin (DOGE), Polygon (MATIC), and Toncoin (TON). This selection was made to explore different areas of cryptocurrencies. The first and second of them—Bitcoin and Ethereum—are fundamental cryptocurrencies. Bitcoin is the first one, and it is known as a classic of cryptocurrencies or digital gold. Bitcoin was introduced by Nakamoto (2008) as a decentralized digital currency that allows for peer-to-peer transactions without the need for a trusted third party. It operates on a blockchain, a distributed ledger technology that ensures transparency and security through cryptographic proof, enabling users to send and receive bitcoins securely over the internet.

Ethereum was introduced by Buterin (2013) as a decentralized platform that enables developers to build and deploy smart contracts and decentralized applications (dApps). Unlike Bitcoin, which primarily focuses on peer-to-peer transactions, Ethereum extends blockchain technology to handle more complex programmable transactions. It is a versatile foundation for various decentralized applications and financial instruments.

Dogecoin, introduced in 2013 by software engineers B. Markus and J. Palmer, started as a humorous take on the cryptocurrency phenomenon, featuring the Shiba Inu dog from the "Doge" meme as its logo. Despite its origins as a joke, Dogecoin quickly developed a dedicated community and became known for its use in tipping and charitable donations. Due to its ease of use and widespread appeal, Dogecoin maintains a significant presence in the cryptocurrency market (Brichta, 2023).

Polygon is a layer 2 solution used as a protocol and a framework for building and connecting Ethereum-compatible blockchain networks. It was established in 2017, and since then, its token MATIC has been traded on top cryptocurrency exchanges. The Polygon network is widely used for crypto asset transfers and is supported by most crypto exchanges, dApps, and crypto services.

Toncoin is a newer cryptocurrency for the TON blockchain used in Telegram messenger. The Toncoin community has its blockchain, a newer solution that has increased in the past year. Technically, the TON blockchain is also a layer 1 solution, and by the last tests, it is the fastest blockchain. Toncoin is a newer cryptocurrency with great prospects and a quickly growing ecosystem.

Price data was taken from CoinMarketCap (https://coinmarketcap.com) – the biggest platform that provides information about cryptocurrencies, their prices, capitalization, rank, and others. For this research historical data of mentioned cryptocurrencies was taken from 2022-01-04 to 2024-04-17. It is a little bit more than two years, ending on a research date. Historical data for the CoinMarketCap platform has these fields: *timeOpen*, *timeClose*, *timeHigh*, *timeLow*, *name*, *open*, *high*, *low*, *close*, *volume*, *marketCap*, and *timestamp*. In research *timeOpen* (time, when trading starts) and *open* (price on trading starts) fields were used.

Before the experiment, all data was prepared using best practices and splitting it into two sets: train (80%) and test (20%). For LSTM and GRU, test and train data were also split into parameters and feature data at a time step of 100. So, the input data for LSTM and GRU was reshaped into sequences of length 100, where each sequence contains 100-time steps and 1 feature per time step. Each sequence is fed into the model during training, with the model learning to predict the next value in the sequence.

2.4. Parameters and architectural settings

Table 1 shows the parameters *p*, *d*, and *q* used in the ARIMA model for each cryptocurrency. They are different for each cryptocurrency because they were chosen with the *auto_arima()* method from the *pmdarima* statistical library in Python, which automatically discovers the optimal order of *p*, *d*, and *q* for the ARIMA model.

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

Table 1. ARIMA model parameters for each cryptocurrency

	p	*d*	*q*
Bitcoin	2	2	1
Ethereum	2	2	2
Dogecoin	4	2	1
Polygon	2	2	3
Toncoin	2	2	1

Source: own processing

Table 2 shows the parameters *p*, *d*, *q*, *P*, *D*, *Q*, and *s* used in the SARIMAX model for each cryptocurrency. These parameters were also selected using the *auto_arima()* method.

Table 2. SARIMAX model parameters for each cryptocurrency

	p	*d*	*q*	*P*	*D*	*Q*	*s*
Bitcoin	1	2	4	2	0	2	12
Ethereum	4	2	4	2	0	0	12
Dogecoin	2	2	3	2	0	1	12
Polygon	2	2	3	1	0	0	12
Toncoin	4	2	1	3	0	0	12

Source: own processing

Table 3 shows the architecture and parameters used for the LSTM model. Specifically, it details the layer types, output shapes, and the number of parameters for each layer in the model:

- Layer type indicating the type of neural network layer (LSTM or Dense).
- Output shape showing the dimensions of the output tensor at each layer.
- Parameters indicating the number of trainable parameters associated with each layer.

Table 3. LSTM model architecture and parameters

Layer type	Output shape	Parameters
LSTM	(none, 100, 50)	10,400
LSTM	(none, 100, 50)	20,200
LSTM	(none, 50)	20,200
Dense	(none, 1)	51

Source: own processing

Table 4 shows the architecture and parameters used for the GRU model.

Table 4. GRU model architecture and parameters

Layer type	Output shape	Parameters
GRU	(none, 100, 50)	7,950
GRU	(none, 100, 50)	15,300
GRU	(none, 50)	15,300
Dense	(none, 1)	51

Source: own processing

LSTM and GRU model architectures were selected using best practices and doing some initial experiments with different architectures. Each prediction was made using its parameters and calculations. For the ARIMA and SARIMA models the own model parameters were used for each cryptocurrency, and for the LSTM and GRU models, the same architectures were used for all cryptocurrencies. Models were prepared and programmed using Python and its libraries (*numpy*, *pandas*, *tensorflow*, *statsmodels*). After the experiment was finished, all results were collected and provided.

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

2.5. Estimating the accuracy of predictions

This research uses the Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) to provide a holistic assessment of the performance of each model:

- MSE measures the average of the squares of the errors between predicted and actual values, providing a sense of the magnitude of prediction errors. It is susceptible to significant errors due to the squaring process, which can highlight considerable discrepancies between predicted and actual values.
- MAE calculates the average absolute differences between predicted and actual values, offering a straightforward interpretation of prediction accuracy. Unlike MSE, it does not square the errors, making it less sensitive to outliers and significant errors.
- RMSE is the square root of the average of squared differences between predicted and actual values, combining the benefits of MSE and providing error magnitude in the same units as the original data. It helps understand the model's prediction error more interpretably.
- MAPE measures the average absolute percent error between predicted and actual values, offering a normalized measure of prediction accuracy. It is beneficial for comparing the predictive accuracy across different datasets and scales since it expresses error as a percentage.

By incorporating all these metrics, the research identifies which model is more precise and how each model performs in varying magnitudes of data. This approach to error estimation makes the analysis much more detailed, providing deeper insights into which models are best suited for specific types of data and forecasting needs.

3. Results

After evaluating the experiment's results, it became evident that traditional time series forecasting models are significantly inferior to RNNs. In all cases, RNN models provided much better forecasts than traditional time series models, as illustrated in Table 5.

Table 5. Comparative analysis of cryptocurrencies forecasting models

Cryptocurrency	Forecasting model	MSE	MAE	RMSE	MAPE
Bitcoin	ARIMA	287296754.8864	12712.0440	16949.8305	0.2180
Bitcoin	SARIMA	247012237.4136	11418.5642	15716.6230	0.1928
Bitcoin	LSTM	17658797.9000	3311.6718	4017.0880	0.0527
Bitcoin	**GRU**	**12537535.3988**	**2896.3915**	**540.8382**	**0.0450**
Etherium	ARIMA	1032049.4497	815.4652	1015.8983	0.2708
Etherium	SARIMA	1672989.5835	1061.8292	1293.4410	0.3562
Etherium	LSTM	50373.9706	190.8614	224.4415	0.0574
Etherium	**GRU**	**30878.1696**	**136.5649**	**175.7219**	**0.0410**
Dogecoin	ARIMA	0.0040	0.0453	0.0633	0.3375
Dogecoin	SARIMA	0.0014	0.0256	0.0379	0.1859
Dogecoin	**LSTM**	**0.0003**	**0.0116**	**0.0165**	**0.0746**
Dogecoin	GRU	0.0003	0.0127	0.0166	0.0808
Polygon	ARIMA	0.1193	0.3088	0.3454	0.3308
Polygon	SARIMA	0.0737	0.2385	0.2715	0.2533
Polygon	LSTM	0.0059	0.0611	0.0771	0.0639
Polygon	**GRU**	**0.0808**	**0.0482**	**0.0634**	**0.0500**
Toncoin	ARIMA	1.9613	0.7338	1.4005	0.1673
Toncoin	SARIMA	1.7605	0.7163	1.3269	0.1704
Toncoin	LSTM	0.4158	0.4798	0.6448	0.1139
Toncoin	**GRU**	**0.2178**	**0.3259**	**0.4667**	**0.0731**

Source: own processing

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

When comparing ARIMA and SARIMA models, Ethereum and Ton were better predicted with the ARIMA model, while Bitcoin, Dogecoin, and Polygon were better predicted with the SARIMA model. However, the differences between ARIMA and SARIMA are not substantial, and we can say that they generally produce similar forecasts. The most noticeable difference is for Dogecoin, where the MAPE for ARIMA is 0.34 compared to 0.19 for SARIMA. For Ethereum, ARIMA provided a better prediction than SARIMA, and for Ton, their performance is practically the same.

The most important observation, seen in Table 5, is that RNN models did a better job of predicting the data. The best predictions are mostly from the GRU model, except for Dogecoin, where the LSTM model performed better. Although, if we look at MAPE, the differences between LSTM and GRU models for Dogecoin are very small.

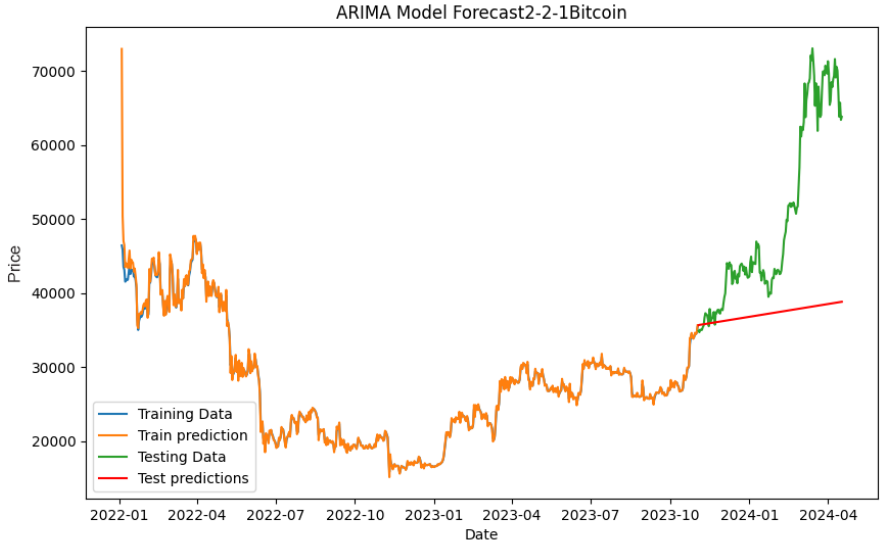

Figure 1. Example of an ARIMA model for forecasting the Bitcoin price

Source: own processing

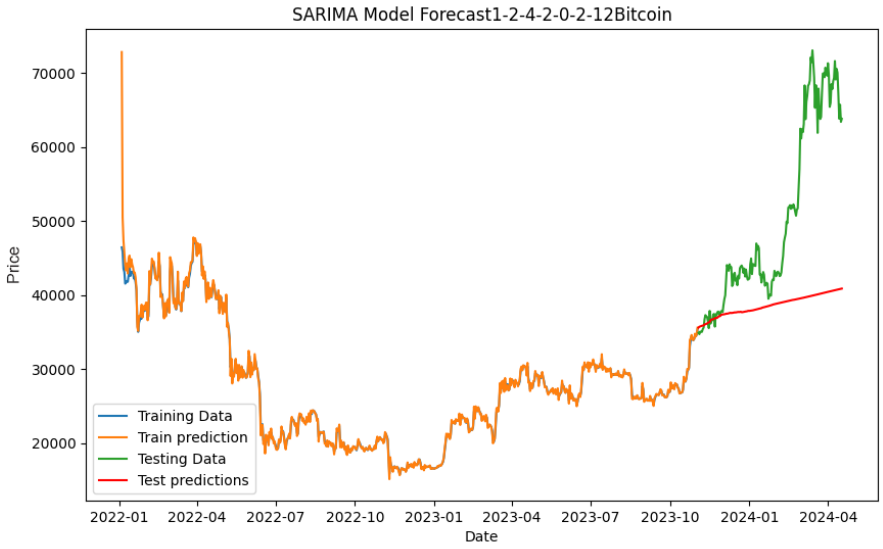

Figure 2. Example of a SARIMA model for forecasting the Bitcoin price

Source: own processing

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

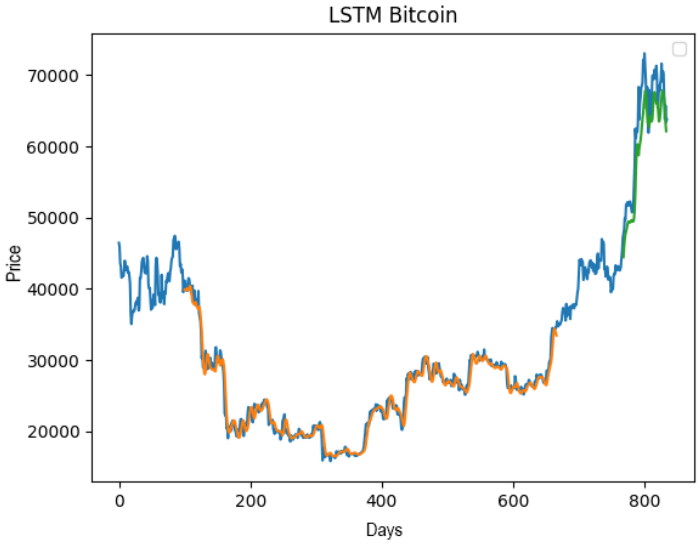

Figure 3. Example of an LSTM model for forecasting the Bitcoin price

Source: own processing

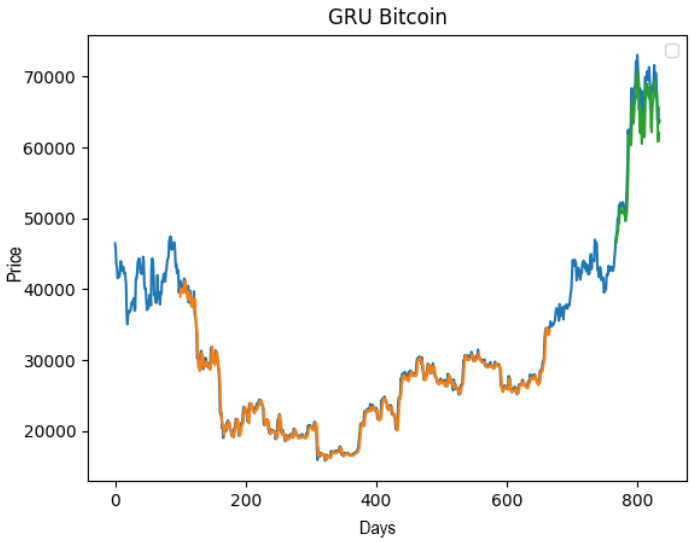

Figure 4. Example of a GRU model for forecasting the Bitcoin price

Source: own processing

The advantage of RNNs is particularly apparent in Figures 1-4. The autoregressive models presented in Figure 1 and Figure 2 predict the future price of Bitcoin almost linearly, regardless of fluctuations. The same tendency has been observed in figures for other cryptocurrencies, so they have not been included in this work to avoid duplication. In contrast, the RNN models in Figure 3 and Figure 4 predict the future price of Bitcoin while taking fluctuations into account, resulting in much more accurate predictions compared to the autoregressive models.

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

Conclusions

The fact that ARIMA and SARIMA models gave similar results suggests that there is no significant seasonality in the data. Since cryptocurrency prices are highly volatile and lack pronounced seasonal patterns, ARIMA and SARIMA models are not suitable for predicting cryptocurrency prices. The accuracy of ARIMA and SARIMA models diminishes, especially over longer periods. While their forecasts may be reasonably accurate over short periods, their accuracy drops significantly over longer horizons due to the high volatility. ARIMA and SARIMA models tend to overfit historical data, resulting in poor performance on new data. This overfitting is particularly problematic in conditions of high volatility, where it leads to a substantial deterioration in forecast accuracy, especially when the new data exhibits new trends.

Meanwhile, RNNs are much better at predicting data, especially in the context of highly volatile and complex time series such as cryptocurrency prices. RNNs can capture long-term dependencies and patterns in data, making them particularly well-suited for handling the non-linear relationships and abrupt changes often seen in cryptocurrency markets. Furthermore, recurrent neural networks are less prone to overfitting compared to ARIMA and SARIMA models. They can generalize unseen data better, providing more reliable forecasts in real-world scenarios.

References

Apostolakis, G.N. (2024). Bitcoin price volatility transmission between spot and futures markets. *International Review of Financial Analysis*, 94, Article Number 103251 http://doi.org/10.1016/j.irfa.2024.103251

Au, C.H., Li, G., Hsu, W.S. Shieh, P.H., Law, K.M.Y. (2024). Characteristics of proliferating cryptocurrencies: a comparative study between stable and non-stable cryptocurrencies. *Enterprise Information Systems* http://doi.org/10.1080/17517575.2024.2356771

Bâra, A., Oprea, S.V., & Panait, M. (2024). Insights into Bitcoin and energy nexus. A Bitcoin price prediction in bull and bear markets using a complex meta model and SQL analytical functions. *Applied Intelligence* http://doi.org/10.1007/s10489-024-05474-2

Bielskis, A., & Belovas, I. (2022). Comparative analysis of stock price ARIMA and LSTM forecasting methods. Lietuvos Matematikos Rinkinys, 63. https://doi.org/10.15388/lmr.2022.29755

Brichta, M. (2023). Fanning Money: The Cultural Economy and Participatory Politics of Dogecoin. *International Journal of Communication*, 17(2023), 6032-6050. Retrieved from https://ijoc.org/index.php/ijoc/article/download/21030/4337

Box, G. E. P., & Jenkins, G. M. 1970. Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco.

Box, G. E. P., & Jenkins, G. M. 1976. Time Series Analysis: Forecasting and Control. Revised Edition, Holden-Day, San Francisco.

Buterin, V. (2013). Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform. Retrieved from https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. https://doi.org/10.3115/v1/w14-4012

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). https://doi.org/10.3115/v1/d14-1179

Corrêa, J. M., Neto, A. C., Teixeira Júnior, L. A., Franco, E. M. C., & Faria, A. E. (2016). Time Series Forecasting with the WARIMAX-GARCH Method. *Neurocomputing*, 216, 805-815. https://doi.org/10.1016/j.neucom.2016.08.046

Demirel, E. (2024). A comparison of arch models: the determinants of bitcoin's price. *Academy Review*, 1, 141-149. http://doi.org/10.32342/2074-5354-2024-1-60-10

Gopu, A., Ramakrishnan, A., Balasubramanian, G., & Srinidhi, K. (2023). A Comparative Study on Forest Fire Prediction using ARIMA, SARIMA, LSTM, and GRU Methods. 2023 IEEE International Conference on Contemporary Computing and Communications (InC4). https://doi.org/10.1109/inc457730.2023.10262965

Hochreiter, S., & Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Computation*, 9, 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jisha, R. C., Nazeer, H., & Kavya, R. (2023). Hybridizing Statistical and Neural Network Models for Enhanced Stock Price Forecasting. 2023 4th IEEE Global Conference for Advancement in Technology (GCAT). https://doi.org/10.1109/gcat59970.2023.10353309

Li, Z., & Zhao, J. (2022). Stock price prediction based on ARIMA-PCA-BP hybrid model. 2022 2nd International Signal Processing, Communications and Engineering Management Conference (ISPCEM). https://doi.org/10.1109/ispcem57418.2022.00058

Kapur, G., Manohar, S., Mittal, A., Jain, V., & Trivedi, S. (2024). Cryptocurrency price fluctuation and time series analysis through candlestick pattern of bitcoin and ethereum using machine learning. *International Journal of Quality & Reliability Management* http://doi.org/10.1108/IJQRM-12-2022-0363

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

Ng, K. Y., Zainal, Z., & Samsudin, S. (2023). Comparative Performance of ARIMA, SARIMA and GARCH Models in Modelling and Forecasting Unemployment Among ASEAN-5 Countries. *International Journal of Business and Society*, 24(3), 967-994. https://doi.org/10.33736/ijbs.6393.2023

Nosouhian, S., Nosouhian, F., & Kazemi Khoshouei, A. (2021). A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU. https://doi.org/10.20944/preprints202107.0252.v1

Respaty, W. A., Hong, C., Putra, N. K., Kurniadi, F. I., & Riccosan. (2023). Weather Prediction in Jakarta: An Analysis of Climate Data and Regional Influences using LSTM and GRU. 2023 International Conference on Data Science and Its Applications (ICoDSA). https://doi.org/10.1109/icodsa58501.2023.10277097

Satheesh, A., & Sundararagan, S. (2022). An analysis of the feasibility of SARIMAX-GARCH through load forecasting. *Journal of Emerging Investigators*. https://doi.org/10.59720/22-085

Sehrawat, P. K., & Vishwakarma, D. K. (2022). Comparative Analysis of Time Series Models on COVID-19 Predictions. 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS). https://doi.org/10.1109/icscds53736.2022.9760992

Si, Y. (2023). Modeling Opening Price Spread of Shanghai Composite Index Based on ARIMA-GRU/LSTM Hybrid Model. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4493799

Singh, S., Pise, A, & Yoon, B. (2024). Prediction of bitcoin stock price using feature subset optimization. *Heliyon*, 10(7). Article Number e28415 http://doi.org/10.1016/j.heliyon.2024.e28415

Vo, N., & Ślepaczuk, R. (2022). Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index. Entropy, 24(2), 158. https://doi.org/10.3390/e24020158

Zarzycki, K., & Ławryńczuk, M. (2021). LSTM and GRU Neural Networks as Models of Dynamical Processes Used in Predictive Control: A Comparison of Models Developed for Two Chemical Reactors. *Sensors*, 21(16), 5625. https://doi.org/10.3390/s21165625

Zhang, Y., Wang, J., & Zhang, X. (2021). Conciseness is better: Recurrent attention LSTM model for document-level sentiment analysis. *Neurocomputing*, 462, 101-112. https://doi.org/10.1016/j.neucom.2021.07.072

Author Contributions: Conceptualization: *Joana Katina, Igor Katin, Vera Komarova*; methodology: *Joana Katina, Igor Katin*; data analysis: *Igor Katin*; writing—original draft preparation: *Joana Katina, Vera Komarova*; writing, review and editing: *Joana Katina, Igor Katin, Vera Komarova*; visualization: *Joana Katina, Igor Katin*. All authors have read and agreed to the published version of the manuscript.

ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 4 (June)
http://doi.org/10.9770/jesi.2024.11.4(26)

Dr. Joana Katina is an Assistant Professor in the Department of Computational and Data Modelling at the Institute of Computer Science, Faculty of Mathematics and Informatics, Vilnius University. She is also an Associate Professor in the Software Development Department, Faculty of Electronics and Informatics, at Higher Education Institution. Research interests: time series analysis, forecasting methods, neural networks, machine learning, artificial intelligence, financial markets.
ORCID ID: https://orcid.org/0000-0002-0715-1675

Dr. Igor Katin is a Teaching Assistant at the Institute of Data Science and Digital Technologies, Faculty of Mathematics and Informatics, Vilnius University. He is also an Associate Professor in the Software Development Department, Faculty of Electronics and Informatics, at Higher Education Institution. Research interests: time series analysis, forecasting methods, neural networks, machine learning, artificial intelligence, financial markets.
ORCID ID: https://orcid.org/0000-0002-2387-4195

Vera Komarova is Dr. oec., Mg. translat., the Leading researcher at the Social Investigations Centre of the Institute of Humanities and Social Sciences of Daugavpils University (Latvia). She is the expert of the Latvian Council of Science in economics and entrepreneurship, economic and social geography, sociology and social work, and other social sciences, as well as the external expert of the COST Association. Research interests: regional economics, sustainable development, economic research methodology, quantitative methods.
ORCID ID: https://orcid.org/0000-0002-9829-622X