VIRTUAL ECONOMICS

ISSN 2657-4047 *(online)*

Research article

INVESTIGATING THE ROLE OF ACTIVATION FUNCTIONS IN PREDICTING THE PRICE OF CRYPTOCURRENCIES DURING CRITICAL ECONOMIC PERIODS

Laszlo Vancsura, Tibor Tatay, and Tibor Bareith

Abstract. Accurate cryptocurrency price forecasting is crucial due to the significant financial implications of prediction errors. The volatile and non-linear nature of cryptocurrencies challenges traditional statistical methods, revealing a gap in effective predictive modelling. This study addresses this gap by examining the impact of activation functions on neural network models during critical economic periods, specifically aiming to determine how optimising activation functions enhances accuracy in neural network models, including RNN, GRU, LSTM, and hybrid architectures. Using data from January 2016 to June 2022—encompassing stable periods, the COVID-19 pandemic, and the onset of the 2022 Ukraine conflict—we analysed price trends under various market conditions. Our methodology involved testing three activation functions (ReLU, sigmoid, and Tanh) across these models. Both univariate and multivariate analyses were conducted, with the latter incorporating additional metrics such as opening, highest, and lowest prices. The results indicate that optimising activation functions enhances prediction accuracy. Among the models, GRU demonstrated the highest accuracy, whereas RNN was the least efficient. Multivariate models outperformed univariate ones, highlighting the benefits of incorporating comprehensive data. Notably, the Tanh activation function led to the greatest improvements, particularly in underperforming models such as RNN. These findings underscore the critical role of activation function selection in enhancing the predictive power of neural networks for cryptocurrency markets. Optimising activation functions can lead to more reliable forecasts, facilitating better trading decisions and risk management. This study highlights activation functions as key parameters in neural network modelling, encouraging further exploration. Future research could investigate different economic periods and cryptocurrency behaviours to assess model robustness. Additionally, examining a broader range of cryptocurrencies may reveal whether the benefits of activation function optimisation are consistent across various assets. Incorporating external factors such as macroeconomic indicators or social media sentiment could further enhance models and improve forecasting accuracy.

Keywords*:* cryptocurrencies; deep learning; forecasting; activation functions; COVID-19; Russian-Ukrainian conflict

Author(s):

Laszlo Vancsura
Department of Agricultural Logistics, Trade and Marketing, Institute of Agricultural and Food Economics, Hungarian University of Agriculture and Life Sciences, 7400 Kaposvár, Hungary
E-mail: vancsura.laszlo@uni-mate.hu
https://orcid.org/0000-0002-6345-8209

Tibor Tatay
Department of Statistics, Finances and Controlling, Szechenyi Istvan University, 9026 Győr, Hungary
E-mail: tatay@sze.hu
https://orcid.org/0000-0001-5436-4545

Tibor Bareith
HUN-REN, Centre for Economic and Regional Studies, Institute of Economics, 1097 Budapest, Hungary
E-mail: bareith.tibor@krtk.hun-ren.hu
https://orcid.org/0000-0002-9971-9597

Corresponding author: Tibor Tatay, tatay@sze.hu

Citation: Vancsura, L., Tatay, T., & Bareith, T. (2024). Investigating the Role of Activation Functions in Predicting the Price of Cryptocurrencies during Critical Economic Periods. *Virtual Economics, 7*(4), 64-91. https://doi.org/10.34021/ve.2024.07.04(4)

Received: September 3, 2024. Revised: November 12, 2024. Accepted: December 9, 2024.

1. Introduction

Financial markets have undergone significant transformation, driven by globalisation and digitalisation, with new products and services emerging over the past several decades. Cryptocurrencies, one of these innovations, have arisen as a new asset class and are increasingly popular with investors [1,2]. Bitcoin, introduced in 2009, is the most prominent cryptocurrency, followed by Ethereum and Litecoin. Since that time, three major advances in cryptocurrency development have occurred: altcoins (essentially copies of Bitcoin), stablecoins (which are pegged to national currencies), and cryptocurrency platforms that enable application development [3]. Cryptocurrencies are classified as either coins or tokens, depending on whether they are built on their own blockchain or an existing one [4]. While the coin/token distinction may not directly affect trading mechanics, it can influence market behaviour and credibility, as coins may be perceived as more independent and technologically robust than tokens.

In this study, we aim to improve price forecasts for three leading cryptocurrencies: Bitcoin, Ethereum, and Litecoin. The period analysed, from 1 January 2016 to 30 June 2022, encompasses significant economic epochs, such as the 2018 period of calm, the COVID-19 pandemic in 2020, and the first months of the conflict in Ukraine (2022). These timeframes were chosen to represent both volatile and stable economic conditions, providing a comprehensive basis for our analysis. Given the relatively recent development of cryptocurrencies, their price data spans a shorter timeframe than traditional assets, hence the limited duration under investigation.

Traditional statistical methods struggle with the non-linear nature of cryptocurrency markets, making machine learning models more appropriate for forecasting despite the limited insight they offer into the relationship between variables [5]. In this study, we utilise several neural network models—Simple Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM)—as well as hybrid models (LSTM-GRU, RNN-LSTM) to evaluate their performance in predicting cryptocurrency prices. The mean absolute percentage error (MAPE) metric is employed to assess the accuracy of the models.

Activation functions are crucial to improving the predictive power of neural networks by enabling the modelling of complex, non-linear relationships [6–9]. In this paper, we explore the impact of three commonly used activation functions—Rectified Linear Units (ReLU), sigmoid, and hyperbolic tangent (Tanh)—on the performance of the models above. By comparing the outcomes of using these activation functions, we aim to assess how optimisation of these functions can lead to better predictions.

The remainder of the paper is organised as follows: In Section 2, we provide a comprehensive literature review of previous studies on cryptocurrency price prediction using machine learning models. Section 3 details the data and methods used in our study, including descriptions of the neural network models and activation functions. Section 4 presents the results of our analyses and discusses the findings in the context of existing literature. Finally, Section 5 concludes the paper with a summary of our key insights and suggests directions for future research.

2. Literature Review

Several studies have explored the efficacy of machine learning models for predicting cryptocurrency prices, focusing on different methodologies and datasets. Sun and colleagues [10] studied the trends of 42 different cryptocurrencies between January and June 2018 using the Light Gradient Boosting Machine (LightGBM), Support Vector Machine, and Random Forest algorithms. The model specifications also used 40 different feature variables. It was found that LightGBM outperformed the other two models in terms of robustness. Wang et al. [11] investigated the predictability of 12 different cryptocurrency returns between August 2017 and March 2021 using Random Forest, Logistic Regression, Support Vector Machine, LSTM, and ANN models. They concluded that the Long Short-Term Memory algorithm achieved the best estimation accuracy. It was also found that only the performance of LSTM could be significantly improved by including trading-related feature variables. Oyedele and co-researchers [12] modelled the prediction of the closing prices of six different cryptocurrencies using Adaptive Boosting (ADA), Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGB), Deep Feedforward Neural Networks (DFNN), Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN). Based on their results, Convolutional Neural Networks performed the best in terms of both estimation accuracy and consistency.

Akyildirim et al. [13] analysed the predictability of 12 cryptocurrency prices using Support Vector Regression, Logistic Regression, Random Forest, and ANN models for the period April 2013 to June 2018. Based on their results, SVR was found to be the most useful method. Borges and Neves [14] came to a similar conclusion regarding the best model performance. The authors analysed the price movements of several cryptocurrencies using technical indicators and machine learning algorithms (Logistic Regression, Random Forest, Support Vector Regression, and Gradient Tree Boosting). In their case, SVR was found to be the most effective, and they also found that using ML methodologies in trading strategies yields higher returns than the buy-and-hold approach. In their study, Zhang et al. [15] compared ARIMA, Random Forest, XGBoost, MLP, LSTM, GRU, CNN, and hybrid models (LSTM+CNN, GRU+CNN, WAMC - Weighted & Attentive Memory Channels) for predicting the prices of six different cryptocurrencies. They concluded that well-constructed hybrid methods can further improve predictive performance, and their proposed WAMC algorithm proved to be the best. Alonso-Monsalve and co-authors [16] addressed the short-term trend analysis of six popular cryptocurrencies. The modelling was based on Convolutional Neural Networks, CNN-LSTM hybrids, Multilayer Perceptron, and Radial Basis Function Neural Network methodologies. The results of the hybrid architecture were always significantly better than the others, and this network was the only one that could predict Dash and Ripple trends with the smallest error (around 4%).

Bitcoin is also popular in the field of scientific research, as it appears in most publications on cryptocurrency. In their study, Cavalli and Amoretti [17] attempted to predict the trend of Bitcoin using price data from April 2013 to February 2020. They used the LSTM and CNN (Convolutional Neural Network) algorithms for modelling. It was found that CNN achieved better estimation accuracy than LSTM.

All of the following studies emphasise the virtues of the LSTM model in predicting Bitcoin prices. Chen et al. [18] investigated Bitcoin price prediction using logistic regression, linear discriminant analysis, Random Forest (RF), XGBoost, quadratic discriminant analysis, Support Vector Machine (SVM), and LSTM models. It was concluded that for 5-minute price data, LSTM provided the best estimation performance, while for modelling with daily price data, logistic regression performed best.

Jaquart et al. [19] used feedforward neural networks, LSTM (Long Short-Term Memory), GRU, Random Forest, and Gradient Boosting models to investigate the direction of Bitcoin price movements in the short term. They found that LSTM achieved the highest estimation accuracy. A comparison of technical, blockchain-based, sentiment/interest-based, and asset-based feature sets showed that, for most methods, technical features remained overwhelmingly important. For longer prediction horizons, the relative importance seemed to be distributed among several features, such as transactions per second and weighted sentiment.

Alkhodhairi et al. [20] sought to estimate the opening, highest, lowest, and closing prices of Bitcoin using deep learning tools (LSTM, GRU). They modelled 4-hour, 12-hour, and 24-hour forecasts, concluding that LSTM produced the best performance. Similarly, Chen et al. [21] studied Bitcoin price prediction using Random Forest, Artificial Neural Network (ANN), LSTM, ARIMA, SVM, Adaptive Network Fuzzy Inference System (ANFIS), and Genetic Algorithm (GA) models. Their results confirmed that LSTM was the model with the highest estimation accuracy. They also found that various predictive algorithms perform better when feature variables are included in the model development process.

Mudassir and colleagues [22] worked on predicting Bitcoin prices, employing four machine learning methods: Artificial Neural Network, Stacked Artificial Neural Network, Support Vector Machine (SVM), and Long Short-Term Memory (LSTM). Their results showed that the actual BTC price could be predicted with a very low error rate, although its rise and fall were much more difficult to anticipate. The classification model using LSTM achieved the best performance in the literature.

Mallqui and Fernandes [23] investigated the predictability of Bitcoin's maximum, minimum, and closing prices using the Multilayer Perceptron and Support Vector Regression models. Their results showed that the SVR algorithm provided the most accurate predictions. They also found that the choice of attributes and the best machine learning model achieved an improvement of more than 10% in price forecast accuracy compared to methodologies presented in previous studies.

Jang and Lee [24] analysed the evolution of Bitcoin prices and their predictability using linear regression, Bayesian Neural Networks, and Support Vector Machine methodologies. Based on predictive performance, they concluded that the Bayesian model was the most appropriate of the three. Similarly, Al-Nefaie and Aldhyani [25] modelled Bitcoin price predictability using Gated Recurrent Unit (GRU) and Multilayer Perceptron (MLP) algorithms for the period January 2021 to June 2022. Their findings indicated that MLP performed marginally better than the GRU model. These results could have a significant impact on asset pricing strategies, considering the uncertainties associated with digital currencies.

Cocco et al. [26] modelled Bitcoin price predictability using Bayesian Neural Network, Feedforward Neural Network, Long Short-Term Memory Neural Network, Support Vector

Machine, and hybrid algorithms. They found that combined models improved predictive performance. Tapia and Kristjanpoller [27] used econometric, machine learning, and hybrid models to predict Bitcoin volatility, concluding that combining AMEM (Asymmetric Multiplicative Error Model) and LSTM significantly enhanced prediction accuracy.

In their study, Dutta and co-authors [28] compared GRU and LSTM models for Bitcoin price prediction. The hybrid model they proposed, enhanced by GRU recurrent selection, proved to be more accurate than other algorithms.

Lahmiri and Bekiros [29] used the LSTM and GRNN (Generalized Regression Neural Networks) models to investigate the price prediction of Bitcoin, Digital Cash, and Ripple. Their findings indicate that the predictive ability of LSTM neural network topologies is significantly higher than that of the GRNN architecture, which was used as a reference. Although the computational burden of the LSTM model is higher, deep learning ultimately proved to be highly effective in capturing the chaotic dynamics inherent in cryptocurrency markets.

Serrano [30] employed Random Neural Network (RNN), LSTM, and linear regression models to predict the prices of Bitcoin, Ethereum, and Ripple. The results demonstrated that NN algorithms exhibited very similar performance, with all outperforming linear regression. A similar conclusion was reached by Uras et al. [31], who examined the price variability and predictability of Bitcoin, Ethereum, and Litecoin. Their analysis incorporated both univariate (closing prices only) and multivariate (including volume, highest, and lowest prices) versions of linear regression and LSTM models. The study found that Ethereum and Litecoin were easier to predict than Bitcoin, with the univariate LSTM model achieving the highest accuracy. However, the authors also noted that regression models had significantly shorter run times compared to LSTM.

Sebastião and Godinho [32] analysed the prices of Bitcoin, Ethereum, and Litecoin between August 2015 and March 2019. Their methodology was based on linear regression, Random Forest, and Support Vector Machine, which were further adapted into trading strategies. Contrary to expectations, linear regression yielded the best predictive results. Furthermore, the study found that forecasts were most accurate for Ethereum and Litecoin.

In their study, Poongodi et al. [33] applied linear regression (LR) and support vector machine (SVM) models to predict Ethereum prices. Their findings indicate that SVM achieved significantly higher estimation accuracy than LR. Similarly, Zoumpekas et al. [34] investigated Ethereum price prediction using various neural network models, including CNN, LSTM, SLSTM, BiLSTM, and GRU. Their results demonstrated that the traditional LSTM algorithm was the most suitable for forecasting Ethereum price data.

Patel et al. [35] examined the predictability of Litecoin and Monero prices for 1-day, 3-day, and 7-day estimation periods. Their findings suggest that a GRU-LSTM hybrid model achieved superior accuracy compared to the traditional LSTM algorithm. Similarly, Peng et al. [36] analysed the predictability of Bitcoin, Ethereum, and Dash volatility using GARCH, SVR, and hybrid models. Their results indicate that the SVR-GARCH hybrid approach outperformed all other models in terms of predictive efficiency.

The literature on activation functions is relatively extensive, but studies specifically addressing cryptocurrency markets remain limited. The following section presents existing research, beginning with general applications and subsequently narrowing the focus to cryptocurrency-specific studies.

Fabozzi et al. [37] employed deep learning neural network regression models to analyse economic and financial data, incorporating 500 different variables. Their research primarily explored the effectiveness of rectified linear unit (ReLU) and sigmoid activation functions, concluding that the ReLU activation function resulted in higher forecasting accuracy than sigmoid.

Konak et al. [38] examined the predictability of dividend payout ratios for companies from 2011 to 2021 using a hybrid method combining genetic algorithms and artificial neural networks (ANNs). A distinguishing feature of their study was the experimentation with six different activation functions to optimise model performance. Their results indicate that a model with three hidden layers yielded the best predictive outcomes, where the activation functions used per layer were, in order: tanh, sigmoid, and radial basis transfer.

Fraszka-Sobczyk and Zakrzewska [39] studied volatility forecasts for the WIG20, DAX, FTSE 250, Nikkei 255, Hang Seng, S&P 500, and Nasdaq 100 stock indices between 2016 and 2020. Their analysis, conducted using the Multi-Layer Perceptron method, found that a combination of sigmoid and tanh activation functions resulted in the best forecasting performance. Furthermore, they noted that the tanh function was particularly effective in the middle layers of the network.

Kayim and Yilmaz [40] investigated the volatility of EUR/USD and gold at different time intervals using RNN and LSTM algorithms. They incorporated sigmoid, ReLU, and their proposed volatility activation function into the models. Their findings indicate that models developed with the volatility function demonstrated improved accuracy and lower average learning and validation loss, suggesting that this approach could serve as a viable alternative to commonly used activation functions.

Tripathi and Sharma [41] focused on short-term Bitcoin price forecasts. Their study employed DANN, LSTM, BiLSTM, and CNN-BiLSTM models, with Bayesian optimisation techniques applied to select hyperparameters. They experimented with linear and ReLU activation functions for optimisation, finding that the linear function produced superior model performance. Additionally, their results indicate that the DANN method, which was designed using technical indicators, achieved the highest estimation accuracy.

Sbrana and Lima de Castro [42] examined the construction of optimal cryptocurrency portfolios using various machine learning and deep learning models. Their findings suggest that the use of the Mish activation function led to greater estimation accuracy compared to the commonly used ReLU function.

 ISSN 2657-4047 (online)

3. Data and Methods
3.1. Applied Methods
3.1.1. Simple Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is an artificial neural network composed of three main layers: input, hidden, and output. Unlike traditional networks, RNNs have two key differences. First, the nodes within the same hidden layer are interconnected. Second, the inputs to the hidden layer at the current time step include the outputs of the input layer at present and the outputs of the hidden layer from the previous time step. This unique structure enables RNNs to better capture the temporal dynamics of sequential data. As a result, RNNs can leverage previously learned information to identify current patterns, improving their ability to model data [43]. The following equations can represent the basic structure of an RNN:

$$S_t = (Ux_t + WS_{t-1} + b_h) \tag{1}$$
$$y_t = f(VS_t + b_0) \tag{2}$$

where x_t represents the inputs at time t, S_t the outputs of the hidden layer, y_t the information of the output layer, f() the activation function, and b_h and b_o the bias vectors of the hidden and output layers, respectively. The weights are defined as W, which is the weight of the hidden layer; U, which is the weight of the inputs at the current time; and V, which is the weight of the outputs.

3.1.2. Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) shares the same input and output structure as a basic RNN. However, its internal structure consists of only two gates: the update gate z_t and the reset gate r_t. The update gate z_t determines how much of the previous memory is retained for the current time step, while the reset gate r_t controls how the new input data is combined with the previous memory. Unlike the LSTM model, the update gate z_t in GRU can both forget and select memory contents, leading to improved computational efficiency and reduced runtime. The following equations govern the GRU:

$$z_t = \sigma(W_z h_{t-1} + U_z x_t) \tag{3}$$
$$r_t = \sigma(W_r h_{t-1} + U_r x_t) \tag{4}$$
$$\tilde{h}_t = \tanh(W_0(h_{t-1} \odot r) + U_0 x_t) \tag{5}$$
$$h_t = z_t \odot \tilde{h}_t + (1 - z_t) \odot h_{t-1} \tag{6}$$

In this case, h_{t-1} represents the hidden state of the neuron at the previous time step, and σ() denotes the logistic sigmoid function, defined as $\sigma(x) = 1/1 + e^{-x}$. The weight matrices for the update gate are W_z and U_z, and for the reset gate, they are W_r and U_r. The weight matrices for the intermediate output are represented by W_0 and U_0. The input value at time t is denoted by x_t, while the hidden layer output and the temporary unit state at time t are represented by $\widetilde{h_t}$ and h_t, respectively [44].

3.1.3. Long-Short Term Memory (LSTM)

RNNs, such as LSTM networks, are commonly used for analyzing sequential data. In these models, short-term memory is associated with internal cell states, while long-term memory is tied to learning weights. LSTM was developed to address the issue of vanishing gradients in traditional RNNs. The key difference is that LSTM replaces the RNN's intermediate layer with memory blocks, allowing it to maintain information over longer periods. The primary advantage of LSTM is its ability to learn long-term dependencies, which standard RNNs cannot do effectively. The data from the initial time interval must be preserved to update the network's weight values and forecast future data points. While RNNs can learn short-term patterns, they struggle with long-term time series, a problem that LSTM is designed to overcome. LSTM consists of memory blocks, also known as recurrent subnets, each containing three gates—input, output, and forget—that control the flow of information. These gates manage continuous read, write, and cell operations, along with one or more autoregressive memory cells [45]. The following equations describe the LSTM model:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \tag{7}$$
$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \tag{8}$$
$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c) \tag{9}$$
$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \tag{10}$$
$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \tag{11}$$

The matrices W_{xc} and W_{hc} represent the weight matrices of the gated unit, and b_c is the bias term for this unit. C_t is the current cell state, while C_{t-1} refers to the cell state at the previous time step. Similarly, W_{xo} and W_{ho} are the weight matrices of the output gate, and b_o is the corresponding bias term [46].

Here, X_t denotes the input batch at time t, and H_{t-1} is the hidden state from the previous time step. The weight matrices for the input gate are W_{xi} and W_{hi}, with b_i as the bias term, and the sigmoid function σ is used in this equation to control the activation. The weight matrices of the forget gate are W_{xf} and W_{hf}, and the bias term is b_f. The candidate memory cells are denoted by C_t. As mentioned earlier, the weight matrices for the gated unit are W_{xc} and W_{hc}, with b_c as the bias term. The current cell state C_t differs from the previous cell state C_{t-1} at this time step, while the output gate's weight matrices remain W_{xo} and W_{ho}, and the bias term remains b_o [46].

3.1.4. LSTM-GRU Hybrid

GRU and LSTM networks can selectively retain important information and discard irrelevant data. Using three gates to control the flow of information, LSTM effectively addresses the problem of long-term dependencies. However, due to the large number of parameters in LSTM, each cell consists of four fully connected layers. In practice, when dealing with large time intervals or deep LSTM networks, there is a higher risk of overfitting, which leads to increased computational requirements.

In contrast, GRU simplifies the LSTM architecture by replacing its input, forget, and output gates with two gates: the update gate (z_t) and the reset gate (r_t). This reduction in parameters lowers the risk of overfitting and decreases computational complexity. However, GRU may not perform as well as LSTM when handling large datasets. A hybrid LSTM-GRU model combines the strengths of both networks, reducing overfitting and achieving highly accurate forecasts [47].

In this hybrid model, the first hidden layer is LSTM. Each LSTM neuron processes the input and generates a weighted output. This data is then passed to the second hidden layer, which is a GRU layer, where another weighted output is produced. Similarly, the data is passed to a third hidden layer, a dense layer, where a final weighted output is generated. The dense layer is a standard fully connected neural network layer used to produce the final output. The data from the dense layer is then passed to the output neuron [48].

3.1.5. RNN-LSTM Hybrid

The RNN-LSTM hybrid model combines the strengths of both RNN and LSTM, significantly improving time series predictability while minimizing their weaknesses [49]. The first hidden layer in this model is an RNN, where neurons process the input and generate a weighted output. This output is then passed to the second hidden layer, an LSTM layer, where another weighted output is produced. Additional weighted values are generated as the data moves from the RNN layer to the LSTM layer. Finally, the data is transferred to the third hidden layer, a dense layer, where the final weighted output is produced.

3.1.6. RNN-GRU Hybrid

While numerous hybrid algorithms are discussed in the literature, research on the RNN-GRU combination remains limited. This model is similar to the RNN-LSTM hybrid, with the first hidden layer being an RNN, where neurons process the input and generate a weighted output. This output is then passed to the second hidden layer, a GRU layer, where additional weighted values are generated. Further weighted outputs are produced as the data moves from the RNN to the GRU layer. Finally, the data is transferred to the third hidden layer, a dense layer, where the final weighted output is generated.
Python version 3.9 was used for modelling with this layer, with the Scikit-Learn and TensorFlow libraries providing essential support for machine learning methods.

3.2. Performance Evaluation

The MAPE was employed in our study to assess the predictive models. For a given set of forecasts, this indicator computes the average magnitude of the error and displays the deviations as a percentage [50].

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{12}$$

 ISSN 2657-4047 (online)

Forecasts are more accurate and reliable when the MAPE value is lower. Since MAPE is not affected by the nominal scale of prices, it is useful for comparing different models and instruments.

3.3. Hyperparameters

The hyperparameters for the models used in this study are shown in Table 2, and the same values were applied to the hybrid models. No hyperparameter optimisation was performed, but the most commonly used values from the literature were applied to the models, except for the activation function, for which three different types were tested.

Table 1. Hyperparameters

Model	Parameters	Value
RNN	Hidden Layers	2
	Hidden layer neuron count	150
	Batch size	16
	Epochs	100
	Activation	ReLU, Tanh, sigmoid
	Learning rate	0,001
	Optimiser	Adam
LSTM	Hidden Layers	2
	Hidden layer neuron count	150
	Batch size	16
	Epochs	100
	Activation	ReLU, Tanh, sigmoid
	Learning rate	0,001
	Optimiser	Adam
GRU	Hidden Layers	2
	Hidden layer neuron count	150
	Batch size	16
	Epochs	100
	Activation	ReLU, Tanh, sigmoid
	Learning rate	0,001
	Optimiser	Adam

Source: Authors' research.

3.4. Activation Functions

Activation functions are crucial for operating neural network models, enabling learning abstract features through non-linear transformations. Some common properties of activation functions are as follows [7]:

- Introduce nonlinearity to the optimisation domain, improving the network's training convergence.
- Avoid significantly increasing the computational complexity of the model.
- Ensure the flow of gradients is not hindered during the learning process.

- Maintain the distribution of data to facilitate more effective network training.

The three most commonly used basic activation functions (Figure 1) are ReLU, sigmoid, and Tanh.

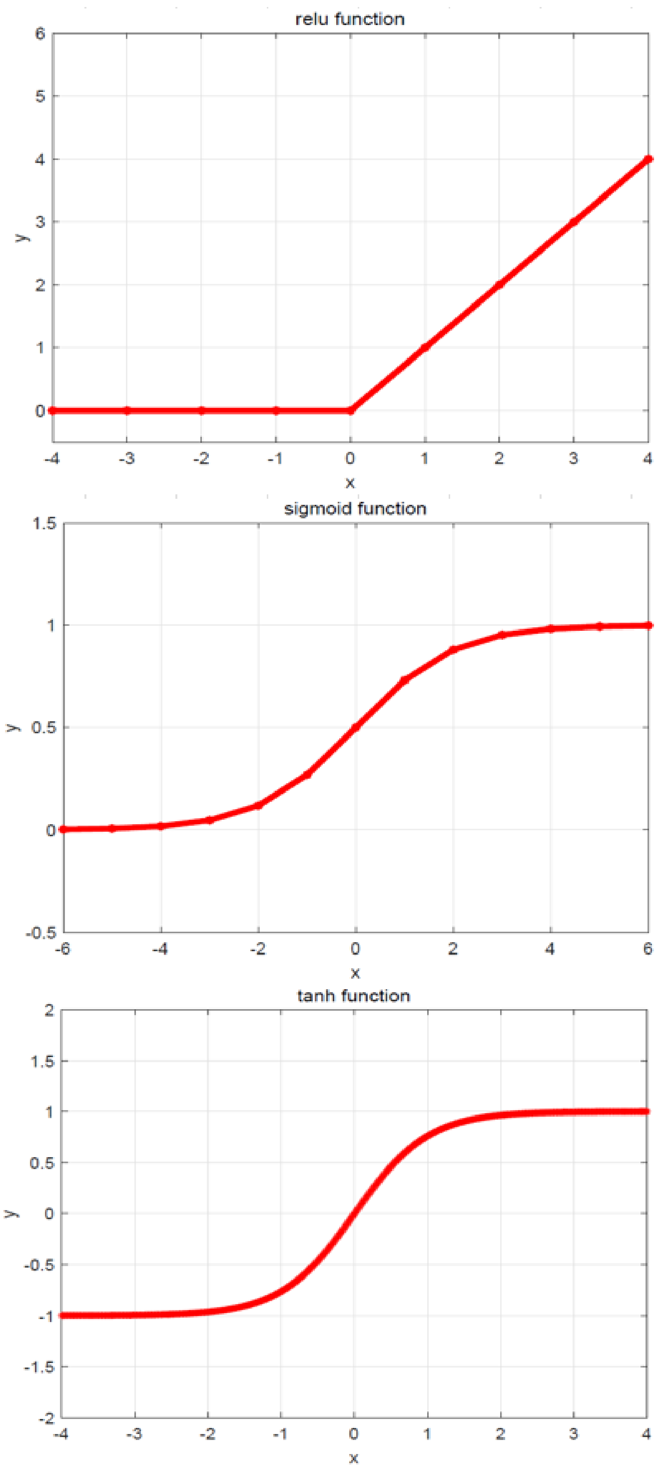

Figure 1. Activation Functions Relu, Sigmoid, and Tanh
Source: Wang et al. [51]

3.4.1. Rectified Linear Unit (ReLU)

Rectified Linear Unit (ReLU) is a widely used activation function that returns the input value for positive and zero for negative inputs. Thus, the range of ReLU is (0, ∞). ReLU addresses the computational complexity issues associated with the sigmoid and Tanh functions. However, a drawback of ReLU is that the gradient vanishes for negative inputs. Despite this limitation, it remains one of the most commonly used activation functions in deep learning models. The ReLU function is calculated as follows [7]:

$$\mathrm{ReLU(x)} = \max(0, x) \tag{13}$$

3.4.2. Sigmoid

Sigmoid is a widely used traditional non-linear activation function that maps input data to a value between 0 and 1. However, for very large or very small inputs, the output of the sigmoid function becomes saturated, leading to the vanishing gradient problem. The vanishing gradient problem occurs when the gradient of the objective function with respect to a parameter becomes very close to zero, preventing meaningful updates to the parameters during training with stochastic gradient descent. Additionally, since the output is not zero-centered, it can result in poor convergence. The sigmoid function is defined as follows [7]:

$$\mathrm{Sigmoid(x)} = \frac{1}{1+\mathrm{e}^{-\mathrm{x}}} \tag{14}$$

3.4.3. Hyperbolic Tangent (Tanh)

The Tanh function is another commonly used activation function in neural network modelling. It is similar to the sigmoid function, but the key difference is that Tanh is zero-centered. The Tanh function compresses the input data to a range between -1 and 1. However, like sigmoid, Tanh also suffers from drawbacks like the vanishing gradient problem and increased computational complexity. The Tanh function is defined as follows [7]:

$$\mathrm{Tanh(x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{15}$$

3.5. Data

In our research, we used the prices of Bitcoin, Ethereum, and Litecoin from 1 January 2016 to 30 June 2022. This period was chosen to capture key economic phases, including the calm period (2018), the COVID-19 pandemic (2020), and the war crisis (2022). We chose these periods because we were interested in comparing the performance of the models during the global economic quiescent period, when we did not observe any ambivalent effects. Then, in order to check the robustness of the models, we used the effects of two consecutive crisis periods to test the consistency of the estimation results of our methodologies in the presence of strong external stress effects. Additionally, because cryptocurrencies are relatively new compared to other financial products, their available price data spans a shorter timeframe. The data were sourced from [52].

After cleaning and scanning the datasets, they were divided into three subsets, and the first period examined a stable economic environment from 1 January 2016 to 30 June 2018. The second period, from 1 January 2018 to 30 June 2020, was chosen due to the impact of the COVID-19 pandemic. The third period relates to the economic instability caused by the Russian-Ukrainian conflict, covering the time interval between 1 January 2020 and 30 June 2022. While both crisis periods were key factors in testing the robustness of the models, it is important to note that cryptocurrencies tend to behave differently from traditional investment products due to their unique cyclicality.

For modelling, the learning and testing datasets were split in an 80%-20% ratio, the most commonly used approach in the literature. For first forecast period (2018) The training dataset covered the period from 1 January 2016 to 31 December 2017 and the testing dataset spanned from 1 January 2018 to 30 June 2018. we have followed a similar approach for the 2020 and 2022 forecasts.

Simulations were conducted using both univariate and multivariate methods. For the univariate models, daily closing prices were used as the basis for predictions, with the current price being estimated based on the data from the previous 50 time units. In the multivariate tests, a similar approach was followed, but in addition to the daily closing prices, the models also incorporated opening, highest, and lowest prices for the same 50-day period.

4. Results and Discussion

4.1. Predictive Performance of Machine Learning Models

Descriptive statistics on the prices of the cryptocurrencies used in the research are presented in Table 2.

Table 2. Descriptive Statistics for the Period 1 January 2016 to 30 June 2022

	N	Average	Median	Std	Min	Max
Bitcoin	2372	14827.59	8040.27	17180.86	364.33	67566.83
Ethereum	2371	807.54	254.81	1149.65	0.92	4800.00
Litecoin	2372	80.21	58.43	69.38	3.00	386.45

Source: Authors' research.

As an initial step, the ReLU activation function was applied in the model runs, and the results are discussed below. The research dataset included Bitcoin, Ethereum, and Litecoin, which together account for approximately 71.68% of the total market capitalization (www.coinmarketcap.com). Figure 2 presents the MAPE values of the predictions for the first half of 2018.

In univariate modelling, GRU provided the most accurate predictive performance for Bitcoin, with a MAPE of 0.0457, while RNN was the least accurate with a MAPE of 0.1771. For Ethereum, the RNN-GRU algorithm yielded the best result (0.0586), whereas RNN-LSTM was the least accurate (0.1015). Simulations for Litecoin proved to be the most challenging, with MAPE values ranging from 0.1130 to 0.3389. The simplest model, RNN, produced the best result (0.1130), while the RNN-GRU hybrid produced the least accurate result (0.3389).

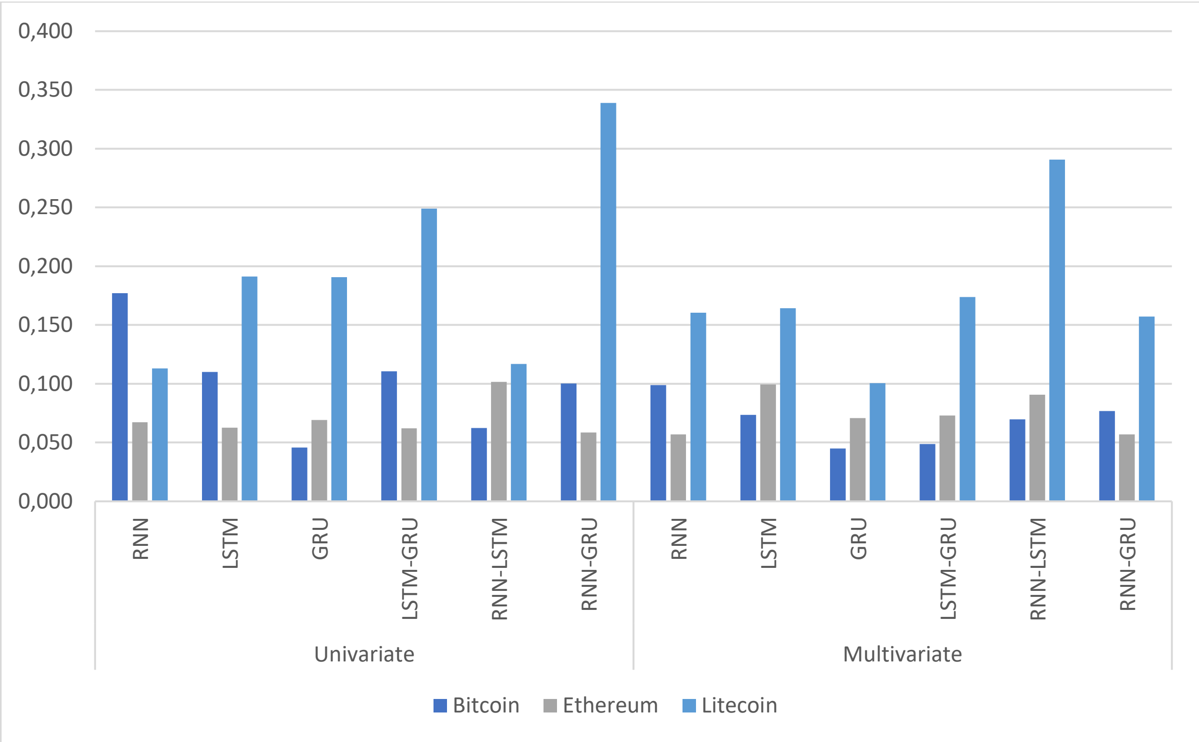

Figure 2. The MAPE values for cryptocurrency forecasts in coefficient form for the first half of 2018 (lower values indicate better accuracy)
Source: Authors' research.

The multivariate modelling enhanced efficiency, as reflected in the improved MAPE values. GRU (0.0448) provided the most accurate prediction for Bitcoin, while RNN had the worst performance (0.0989). Ethereum also saw improved prediction accuracy, with RNN and RNN-GRU achieving the best result (0.0570). The LSTM model placed last with a MAPE of 0.0994, though this still represented an improvement over the univariate estimation. For Litecoin, GRU produced the most accurate estimate (0.1006), while the RNN-LSTM hybrid was the least accurate (0.2907). Overall, using multivariate algorithms improved prediction performance for the first half of 2018 across all cryptocurrencies analyzed.

The cryptocurrency sector was the least affected by the issues related to the spread of COVID-19, as indicated in the test results (Figure 3). In the univariate tests, GRU provided the best predictive performance for Bitcoin, with a MAPE of 0.0301, while RNN had the worst performance with a MAPE of 0.0411. LSTM was the most accurate for Ethereum, achieving an error value of 0.0364, whereas RNN had the lowest predictive accuracy (0.0547). For Litecoin, LSTM again produced the lowest MAPE (0.0342), while GRU had the highest (0.0533).

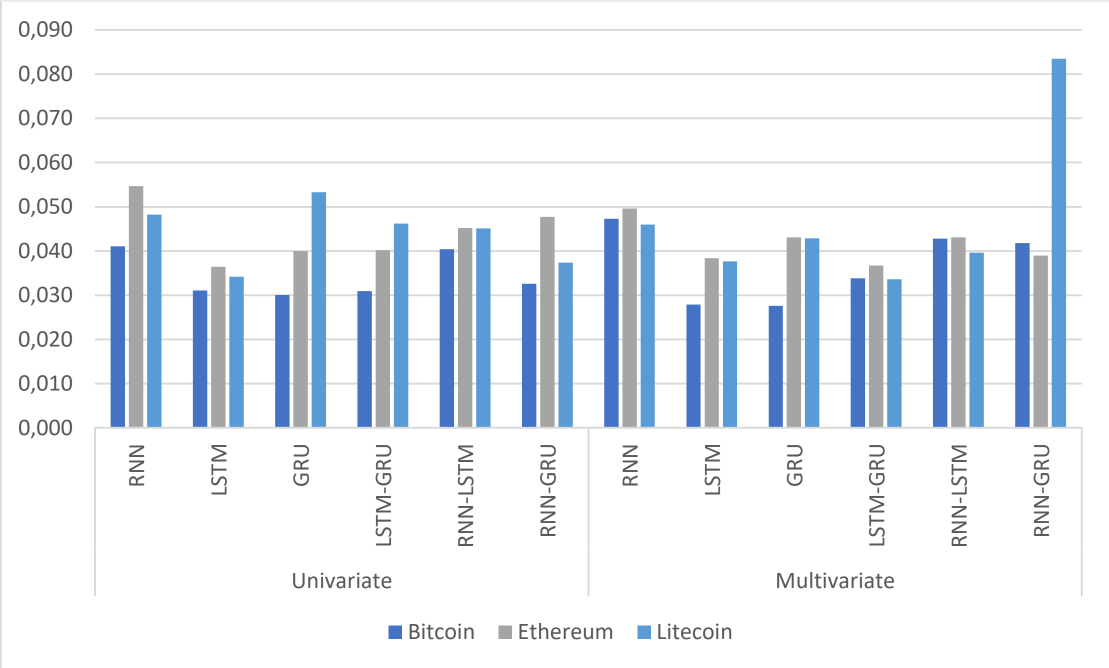

Figure 3. The MAPE values for cryptocurrency forecasts in coefficient form for the first half of 2020 (lower values indicate better accuracy).
Source: Authors' research.

Using multivariate models, the GRU model (0.0276) delivered the best performance for Bitcoin, with LSTM following closely behind (0.0279). RNN, however, had the worst performance with a MAPE of 0.0473. For Ethereum, the LSTM-GRU hybrid achieved the lowest error value among all methods tested (0.0367), although it did not surpass the best performance from the univariate models. RNN was the least accurate, with a MAPE of 0.0496. For Litecoin, the LSTM-GRU hybrid provided the most accurate forecast for the first half of 2020, with a MAPE of 0.0336, while the RNN-GRU hybrid was the least reliable, with an outlier MAPE of 0.0835. Notably, except for Ethereum, the lower bound of the MAPE values was reduced, indicating that multivariate models generally offered improved performance over univariate models.

In the third forecast period, the first half of 2022 (Figure 4), GRU achieved the most accurate predictive performance for Bitcoin in univariate simulations, with a MAPE of 0.0274, while RNN-LSTM performed significantly worse, with a MAPE of 0.0608. For Ethereum, the GRU algorithm also outperformed the other models, achieving a MAPE of 0.0357, while RNN-LSTM was the least accurate, with a MAPE of 0.0614. For Litecoin, the MAPE values ranged from 0.0367 to 0.0644, with LSTM delivering the best result and traditional RNN producing the weakest.

In the multivariate modelling, LSTM-GRU (0.0284) was the most efficient for Bitcoin, while RNN-LSTM had the worst performance (0.0441). No significant improvement was observed for Ethereum with the inclusion of additional explanatory variables. LSTM performed the best (0.0433), slightly ahead of LSTM-GRU (0.0438), with RNN-GRU being the least accurate

(0.0468). The difference between the lower and upper bounds of the interval was minimal in this case. For Litecoin, LSTM provided the most accurate prediction (0.0418), while GRU was the least accurate (0.0653). Overall, using multivariate methods did not lead to improved predictive performance.

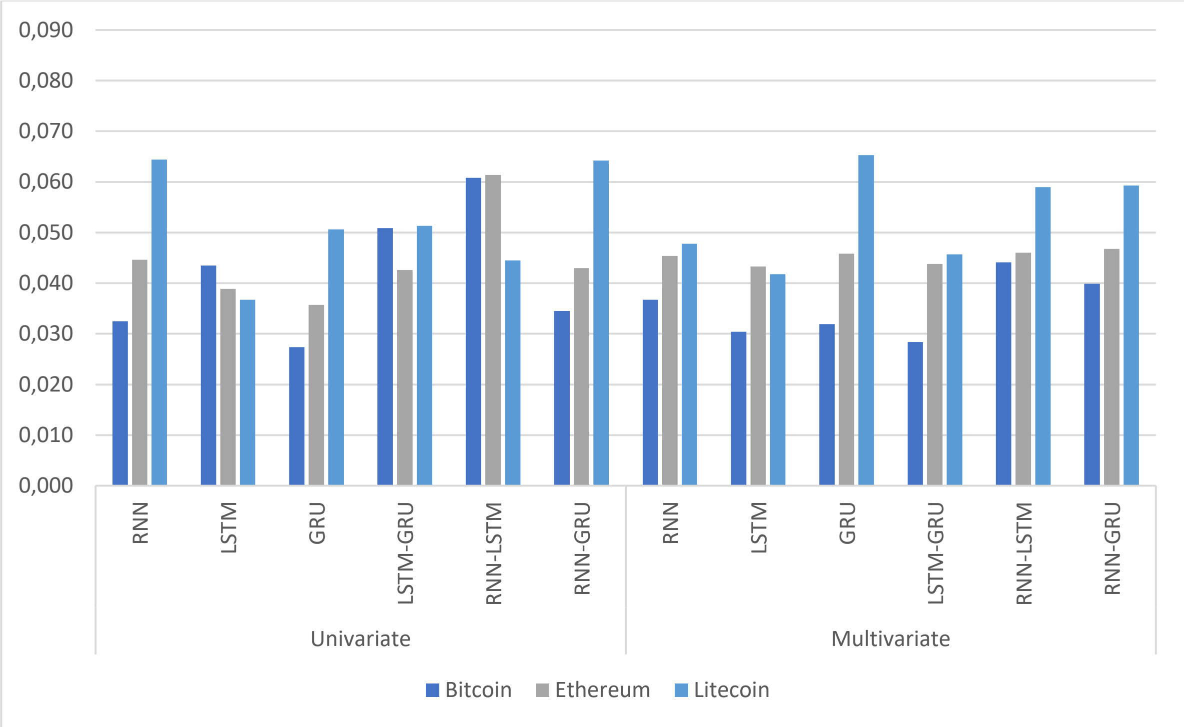

Figure 4. The MAPE values for cryptocurrency forecasts in coefficient form for the first half of 2022 (lower values indicate better accuracy)
Source: Authors' research.

Figure 5 displays the average estimation performance for cryptocurrencies across the three periods, categorized by the models used. The results indicate that what was expected to be a relatively calm period (2018) produced sextreme outcomes in the cryptocurrency market, making it the most challenging period for model reliability. Among all the methods tested, the multivariate GRU algorithm provided the most accurate predictions overall, while the univariate RNN-GRU hybrid method performed the worst. The most stable MAPE values were observed in 2020, during the peak of the COVID-19 pandemic, when cryptocurrency markets experienced lower volatility than in other periods. On average, the LSTM method performed the best, while RNN-GRU lagged. The 2022 period showed less extreme predictive performance than 2018, with the univariate GRU model proving the most reliable. In contrast, the RNN-LSTM hybrid algorithm, which used a single explanatory variable, delivered the least accurate predictions.

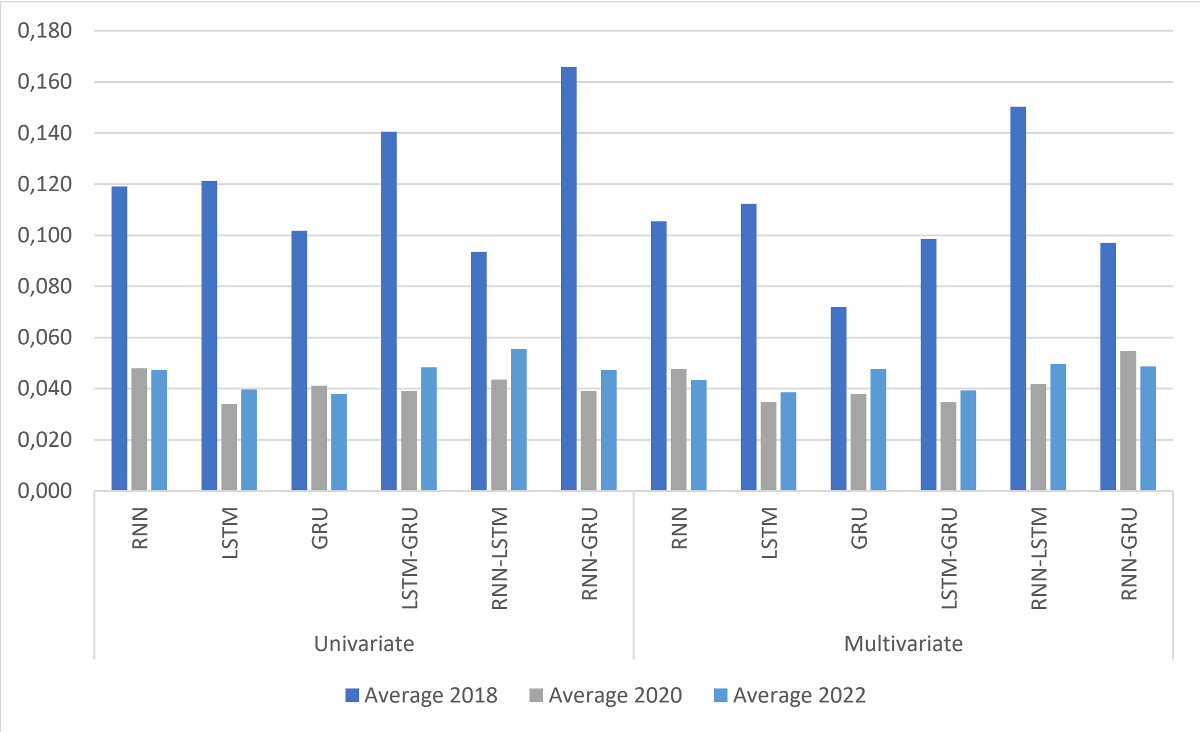

Figure 5. The values of the average MAPE indicators for the first half of 2018, 2020 and 2022 in coefficient form (lowers are better)
Source: Authors' research.

4.2. Optimisation of Activation Functions

The literature emphasizes the importance of activation functions in various predictive models [6-9]. Therefore, we sought to replace the commonly used ReLU function, primarily used in regression models in our study, with two alternatives to test whether this substitution affects predictive performance. In subsequent tests, both univariate and multivariate estimators for the three periods were evaluated using the Tanh and sigmoid functions. The modelling results and comparisons to the original model runs are presented below. We first illustrate the changes in MAPE values for 2018, followed by 2020, and finally, 2022.

Table 3 presents the MAPE values for all three Bitcoin activation functions in 2018. Among the univariate models, GRU was the only one that did not show improvement over the previous forecast performance. Proportionally, the largest improvement was observed with RNN using the sigmoid function. Although the MAPE values for the other models also decreased, the absolute best result for GRU (0.0457) remained unchanged. After the multivariate model runs, neither GRU nor RNN-GRU demonstrated a better fit to the real price. The best MAPE value remained unchanged, while positive changes were observed for the other models.

Table 3. Bitcoin Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2018 Forecast

Type of Model	Activation Function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.1771	0.1101	**0.0457**	0.1107	0.0625	0.1002
	Tanh	0.1078 (6.93)	0.0493 (6.08)	Not improved	0.0805 (3.02)	0.0617 (0.08)	0.0774 (2.28)
	Sigmoid	0.0656 (11.15)	Not improved	Not improved	Not improved	Not improved	Not improved
Multivariate	ReLu	0.0989	0.0735	**0.0448**	0.0488	0.0696	0.0767
	Tanh	0.0818 (1.71)	0.0532 (2.03)	Not improved	0.0480 (0.08)	Not improved	Not improved
	Sigmoid	0.0827 (1.62)	Not improved	Not improved	Not improved	0.0688 (0.08)	Not improved

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: Authors' research.

For Ethereum in 2018 (Table 4), we observed that in the univariate models, neither LSTM-GRU nor RNN-GRU responded positively to the newly introduced activation functions. Significant improvement was only achieved with the application of the Tanh function. LSTM and GRU showed increased fitting accuracy, with GRU demonstrating the most improvement, as its MAPE value dropped to 0.0474. In the multivariate tests, LSTM-GRU and RNN-GRU hybrids showed no improvement. The GRU-sigmoid combination performed best, with a MAPE of 0.0502.

Table 4. Ethereum Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2018 Forecast

Type of Model	Activation Function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.0674	0.0626	0.0692	0.0620	0.1015	0.0586
	Tanh	0.0610 (0.64)	0.0486 (1.40)	**0.0474** (2.18)	Not improved	0.0993 (0.22)	Not improved
	Sigmoid	Not improved	Not improved	0.0635 (0.57)	Not improved	Not improved	Not improved
Multivariate	ReLu	0.0570	0.0994	0.0707	0.0729	0.0908	0.0570
	Tanh	Not improved	0.0683 (3.11)	0.0646 (0.61)	Not improved	Not improved	Not improved
	Sigmoid	0.0524 (0.46)	Not improved	**0.0502** (2.05)	Not improved	0.0752 (1.56)	Not improved

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: Authors' research.

The effects of the changes made to the Litecoin 2018 forecasting models on the MAPE values are shown in Table 5. None of the newly applied activation functions for the univariate LSTM and RNN-LSTM models led to improvement. However, the MAPE values were reduced to 0.0819 for RNN (sigmoid) and 0.0755 for GRU (sigmoid). The largest improvements were observed in the LSTM-GRU and RNN-GRU hybrid models, where the Tanh function yielded significantly positive changes. The MAPE was reduced to 0.1055 for LSTM-GRU and to 0.0861 for RNN-GRU. In the multivariate models, the performance of LSTM and GRU did not improve further. For RNN, however, the MAPE was reduced to 0.0854 due to the application of the sigmoid function. All hybrid algorithms improved in the new simulations, with MAPE values of 0.0829 for LSTM-GRU, 0.1809 for RNN-LSTM, and 0.1049 for RNN-GRU using the Tanh activation function.

Table 5. Litecoin Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2018 Forecast

Type of Model	Activation Function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.1130	0.1912	0.1906	0.2490	0.1168	0.3389
	Tanh	Not improved	Not improved	0.1591 (3.15)	0.1055 (14.35)	Not improved	0.0861 (25.28)
	Sigmoid	0.0819 (3.11)	Not improved	**0.0755** (11.51)	Not improved	Not improved	Not improved
Multivariate	ReLu	0.1604	0.1642	0.1006	0.1739	0.2907	0.1573
	Tanh	0.1564 (0.40)	Not improved	Not improved	**0.0829** (9.10)	0.1809 (10.98)	0.1049 (5.24)
	Sigmoid	0.0854 (7.50)	Not improved	Not improved	0.1277 (4.62)	0.2013 (8.94)	0.1352 (2.21)

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: Authors' research.

It is important to note that the 2020 period showed a much more stable picture in terms of volatility compared to 2018, which is reflected in the error values. Table 6 presents the MAPE values optimised with Bitcoin activation functions. All algorithms except RNN-GRU successfully ran the new simulations in the univariate models. For the LSTM, LSTM-GRU, and RNN-LSTM models, both Tanh and sigmoid functions led to improved prediction performance. As in the control method, GRU achieved the lowest MAPE values. However, GRU did not show any improvement in the multivariate models, unlike the others. LSTM had the lowest error with the Tanh function (0.0269).

Table 6. Bitcoin Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2020 Forecast

Type of Model	Activation Function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.0411	0.0311	0.0301	0.0309	0.0404	0.0326
	Tanh	0.0331 (0.80)	0.0292 (0.19)	Not improved	0.0290 (0.19)	0.0393 (0.11)	Not improved
	Sigmoid	Not improved	0.0285 (0.26)	**0.0279** (0.22)	0.0300 (0.09)	0.0326 (0.78)	Not improved
Multivariate	ReLu	0.0473	0.0279	0.0276	0.0338	0.0428	0.0418
	Tanh	0.0402 (0.71)	**0.0269** (0.01)	Not improved	0.0332 (0.56)	0.0386 (0.42)	0.0380 (0.38)
	Sigmoid	0.0321 (1.52)	Not improved	Not improved	0.0325 (0.13)	Not improved	0.0342 (0.76)

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: Authors' research.

For Ethereum (Table 7), univariate tests for the first half of 2020 forecasts showed that four models—RNN, LSTM, LSTM-GRU, and RNN-LSTM—did not produce any significant improvements. RNN-GRU showed modest gains, while GRU achieved better-than-optimal results with MAPE values of 0.0377 using the Tanh function and 0.0363 using the sigmoid function, compared to its original 0.0400. In the multivariate model runs, only the combination of LSTM (0.0374) and LSTM-GRU (0.0363) with the Tanh function resulted in more accurate forecasts than the original models.

Table 7. Ethereum Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2020 Forecast

Type of Model	Activation Function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.0547	0.0364	0.0400	0.0402	0.0452	0.0477
	Tanh	Not improved	Not improved	0.0377 (0.23)	Not improved	Not improved	0.0469 (0.08)
	Sigmoid	Not improved	Not improved	**0.0363** (0.37)	Not improved	Not improved	Not improved
Multivariate	ReLu	0.0496	0.0384	0.0431	0.0367	0.0431	0.0390
	Tanh	Not improved	0.0374 (0.10)	Not improved	**0.0363** (0.04)	Not improved	Not improved
	Sigmoid	Not improved	Not improved	Not improved	Not improved	Not improved	Not improved

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: Authors' research.

The MAPE values for Litecoin's 2020 predictions are shown in Table 8. In the univariate tests, the performance of the LSTM and RNN-GRU algorithms did not improve further. Overall, LSTM-GRU produced the best results using the Tanh activation function (0.0336), surpassing the lowest error value recorded previously. GRU also showed significant improvement, with the sigmoid function responsible for the positive change (0.0369). However, in the multivariate models, only RNN-GRU achieved a better result than before (0.0490). Despite this improvement, RNN-GRU's original estimate had been considerably less accurate than the other algorithms, with nearly double the MAPE value (0.0835).

Table 8. Litecoin Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2020 Forecast

Type of model	Activation function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.0482	0.0342	0.0533	0.0462	0.0451	0.0374
	Tanh	0.0448 (0.34)	Not improved	0.0416 (1.17)	**0.0336** (1.26)	0.0446 (0.05)	Not improved
	Sigmoid	Not improved	Not improved	0.0369 (1.64)	Not improved	Not improved	Not improved
Multivariate	ReLu	0.0460	0.0377	0.0429	**0.0336**	0.0396	0.0835
	Tanh	Not improved	Not improved	Not improved	Not improved	Not improved	0.0490 (3.45)
	Sigmoid	Not improved	Not improved	Not improved	Not improved	Not improved	Not improved

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: authors' research.

Table 9. Bitcoin Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2022 Forecast

Type of model	Activation function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.0325	0.0435	**0.0274**	0.0509	0.0608	0.0345
	Tanh	Not improved	0.0340 (0.95)	Not improved	0.0363 (1.46)	0.0462 (1.46)	0.0320 (0.25)
	Sigmoid	0.0296 (0.29)	0.0334 (1.01)	Not improved	0.0450 (0.59)	0.0366 (2.42)	Not improved
Multivariate	ReLu	0.0367	0.0304	0.0319	**0.0284**	0.0441	0.0399
	Tanh	0.0309 (0.58)	Not improved	0.0316 (0.03)	Not improved	Not improved	Not improved
	Sigmoid	0.0306 (0.61)	Not improved	Not improved	Not improved	Not improved	0.0333 (0.66)

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: Authors' research.

Bitcoin price forecasts for 2022 are presented in Table 9. In the univariate tests, none of GRU's newly applied activation functions resulted in more accurate estimates. It is worth noting, however, that none of the optimised models approached the initial MAPE value of 0.0274 obtained during the original modelling. The largest improvement was seen in the RNN-LSTM model, where the sigmoid function reduced the MAPE from 0.0608 to 0.0366. In the multivariate models, LSTM, LSTM-GRU, and RNN-LSTM showed no improvement over their original performance. Other algorithms produced modest improvements of 0.5-0.6%. For GRU, only minimal improvement was observed.

The MAPE values for the activation function optimisation of the models used to forecast Ethereum prices in the 2022 period are presented in Table 10. The univariate modelling results found that the RNN, GRU, and RNN-GRU models did not achieve further performance improvements. Only the LSTM-Tanh combination approached the accuracy achieved by the GRU-ReLU pair. In the multivariate simulations, the MAPE performance of LSTM and RNN-GRU showed no favorable changes. However, a more significant reduction of 0.77% for GRU compared to the initial condition was observed. Additionally, both hybrid models (LSTM-GRU and RNN-LSTM) were able to reduce the error to below 4% in the deviation between actual and predicted prices.

Table 10. Ethereum Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2022 Forecast

Type of Model	Activation Function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.0446	0.0389	**0.0357**	0.0426	0.0614	0.0430
	Tanh	Not improved	0.0361 (0.28)	Not improved	0.0399 (0.27)	0.0558 (0.56)	Not improved
	Sigmoid	Not improved	Not improved	Not improved	0.0397 (0.29)	0.0439 (1.75)	Not improved
Multivariate	ReLu	0.0454	0.0433	0.0458	0.0438	0.0460	0.0468
	Tanh	0.0437 (0.17)	Not improved	**0.0381** (0.77)	Not improved	Not improved	Not improved
	Sigmoid	Not improved	Not improved	0.0429 (0.29)	0.0395 (0.43)	0.0398 (0.62)	Not improved

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: Authors' research.

The MAPE values of the modified models for Litecoin price forecasts up to 2022 are presented in Table 11. No further accuracy improvements were achieved with the univariate RNN, LSTM, RNN-LSTM, and RNN-GRU models. However, GRU with the Tanh function improved to 0.0487, and LSTM-RNN with the sigmoid function improved to 0.0388. The multivariate RNN, LSTM-GRU, and RNN-LSTM models did not yield better results. However, LSTM (0.0387), GRU (0.0389), and RNN-GRU (0.0581) showed improvement after applying the Tanh activation function. As a result, the previous best result (0.0418) was surpassed by the LSTM model.

Table 11. Litecoin Univariate and Multivariate MAPE Indicators Using Different Activation Functions for the First Half of 2022 Forecast

Type of Model	Activation Function	RNN	LSTM	GRU	LSTM-GRU	RNN-LSTM	RNN-GRU
Univariate	ReLu	0.0644	**0.0367**	0.0506	0.0513	0.0445	0.0642
	Tanh	Not improved	Not improved	0.0487 (0.19)	0.0404 (1.09)	Not improved	Not improved
	Sigmoid	Not improved	Not improved	Not improved	0.0388 (1.25)	Not improved	Not improved
Multivariate	ReLu	0.0478	0.0418	0.0653	0.0457	0.0590	0.0593
	Tanh	Not improved	**0.0387** (0.31)	0.0389 (2.64)	Not improved	Not improved	0.0581 (0.12)
	Sigmoid	Not improved	Not improved	Not improved	Not improved	Not improved	Not improved

Note: In parentheses, the performance improvement achieved as a result of changing the activation function in percentage compared to ReLU.
Source: Authors' research.

5. Conclusions

Accurate price forecasting in cryptocurrency markets is crucial due to the significant financial implications of prediction errors, which can lead to premature or delayed trading decisions and unnecessary transaction costs. This study examined the impact of different activation functions on neural network models—specifically RNN, LSTM, GRU, and hybrid models—in predicting the prices of Bitcoin, Ethereum, and Litecoin during critical economic periods.

Our findings indicate that optimising activation functions significantly enhances prediction accuracy. The GRU model consistently produced the most accurate forecasts, while the RNN model was the least efficient. Multivariate models, which incorporated additional price metrics such as opening, highest, and lowest prices, generally outperformed univariate models, demonstrating the benefit of utilizing more comprehensive data. Notably, the use of the Tanh activation function led to the greatest improvements, particularly in previously underperforming models like RNN.

These results suggest that careful selection and optimisation of activation functions can lead to more reliable cryptocurrency price forecasts, aiding practitioners in making better trading decisions and managing risks in the dynamic and often unpredictable cryptocurrency markets. For researchers, the study underscores the critical role of activation functions as a key parameter in neural network modelling for financial forecasting, encouraging further exploration in this area.

Future studies could expand on our findings by exploring different economic periods and cryptocurrency behaviors. Investigating the models' performance during various market conditions—such as extreme volatility, regulatory changes, or market bubbles—could provide deeper insights into their robustness and adaptability. Additionally, examining a broader range of cryptocurrencies, including emerging or less-traded ones, may reveal whether the benefits of activation function optimisation are consistent across different assets with varying market dynamics. Incorporating external factors like macroeconomic indicators, social media sentiment, or blockchain-specific metrics could also enhance model inputs and potentially improve forecasting accuracy.

Author Contributions: Conceptualization, L.V., T.T. and T.B.; methodology, L.V.; software, L.V.; validation, L.V., T.T. and T.B.; formal analysis, T.B.; investigation, T.T.; resources, L.V.; data curation, L.V.; writing-original draft preparation, L.V.; writing-review and editing, L.V.; visualization, L.V.; supervision, T.B.; project administration, T.T.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflictsof interest.

Appendix A

Abbreviation	Full Term
ADA	Adaptive Boosting (AdaBoost)
AMEM	Asymmetric Multiplicative Error Model
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
ARIMA	AutoRegressive Integrated Moving Average
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
DFNN	Deep Feedforward Neural Network
FNN	Feedforward Neural Networks
GA	Genetic Algorithm
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GBM	Gradient Boosting Machine
GRNN	Generalized Regression Neural Network
GRU	Gated Recurrent Unit
LDA	Linear Discriminant Analysis
LightGBM	Light Gradient Boosting Machine
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multilayer Perceptron
QDA	Quadratic Discriminant Analysis
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network

Abbreviation	Full Term
SLSTM	Stacked Long Short-Term Memory
SVM	Support Vector Machine
SVR	Support Vector Regression
Tanh	Hyperbolic Tangent (Activation Function)
WAMC	Weighted & Attentive Memory Channels
XGB/XGBoost	Extreme Gradient Boosting

References

1. Ji, Q., Bouri, E., Lau, C. K. M., & Roubaud, D. (2019). Dynamic connectedness and integration in cryptocurrency markets. *International Review of Financial Analysis, 63*, 257-272. https://doi.org/10.1016/j.irfa.2018.12.002
2. Zhang, H., Ho, T. B., Lin, M. S. (2004). A Non-Parametric Wavelet Feature Extractor for Time Series Classification. In Advances in Knowledge Discovery and Data Mining; Springer: Berlin/Heidelberg, Germany, 2004; pp. 595–603. https://doi.org/10.1007/978-3-540-24775-3_71
3. Ong, B., Lee, T. M., Li, G., & Chuen, D. L. K. (2024). Evaluating the potential of alternative cryptocurrencies. In *Handbook of digital currency* (pp. 79-132). Academic Press. https://doi.org/10.1016/B978-0-323-98973-2.00022-8
4. Ellahie, A. (2024). Disclosure in Initial Coin Offerings. In *The Palgrave Encyclopedia of Private Equity* (pp. 1-7). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-38738-9_217-1
5. Hoang, D., & Wiegratz, K. (2023). Machine learning methods in finance: Recent applications and prospects. *European Financial Management, 29*(5), 1657-1701. https://doi.org/10.1111/eufm.12408
6. Apicella, A., Donnarumma, F., Isgrò, F., & Prevete, R. (2021). A survey on modern trainable activation functions. *Neural Networks, 138*, 14-32. https://doi.org/10.1016/j.neunet.2021.01.026
7. Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing, 503*, 92-108. https://doi.org/10.1016/j.neucom.2022.06.111
8. Singh, B., Patel, S., Vijayvargiya, A., & Kumar, R. (2023). Analyzing the impact of activation functions on the performance of the data-driven gait model. *Results in Engineering, 18*, 101029. https://doi.org/10.1016/j.rineng.2023.101029
9. Szandała, T. (2021). Review and comparison of commonly used activation functions for deep neural networks. *Bio-inspired neurocomputing*, 203-224. https://doi.org/10.1007/978-981-15-5495-7_11
10. Sun, X., Liu, M., & Sima, Z. (2020). A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters, 32*, 101084. https://doi.org/10.1016/j.frl.2018.12.032
11. Wang, Y., Wang, C., Sensoy, A., Yao, S., & Cheng, F. (2022). Can Investors' Informed Trading Predict Cryptocurrency Returns? Evidence from Machine Learning. *Research in International Business and Finance*, 101683. https://doi.org/10.1016/j.ribaf.2022.101683
12. Oyedele, A. A., Ajayi, A. O., Oyedele, L. O., Bello, S. A., & Jimoh, K. O. (2023). Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. *Expert Systems with Applications, 213*, 119233. https://doi.org/10.1016/j.eswa.2022.119233
13. Akyildirim, E., Goncu, A., & Sensoy, A. (2021). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research, 297*(1), 3-36. https://doi.org/10.1007/s10479-020-03575-y
14. Borges, T. A., & Neves, R. F. (2020). Ensemble of machine learning algorithms for cryptocurrency investment with different data resampling methods. *Applied Soft Computing, 90*, 106187. https://doi.org/10.1016/j.asoc.2020.106187
15. Zhang, Z., Dai, H. N., Zhou, J., Mondal, S. K., García, M. M., & Wang, H. (2021). Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels. *Expert Systems with Applications, 183*, 115378. https://doi.org/10.1016/j.eswa.2021.115378
16. Alonso-Monsalve, S., Suárez-Cetrulo, A. L., Cervantes, A., & Quintana, D. (2020). Convolution on neural networks for high-frequency trend prediction of cryptocurrency prices using technical indicators. *Expert Systems with Applications, 149*, 113250. https://doi.org/10.1016/j.eswa.2020.113250

17. Cavalli, S., & Amoretti, M. (2021). CNN-based multivariate data analysis for bitcoin trend prediction. *Applied Soft Computing, 101*, 107065. https://doi.org/10.1016/j.asoc.2020.107065
18. Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics, 365*, 112395. https://doi.org/10.1016/j.cam.2019.112395
19. Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. *The journal of finance and data science, 7*, 45-66. https://doi.org/10.1016/j.jfds.2021.03.001
20. Alkhodhairi, R. K., Aljalhami, S. R., Rusayni, N. K., Alshobaili, J. F., Al-Shargabi, A. A., & Alabdulatif, A. (2021). Bitcoin Candlestick Prediction with Deep Neural Networks Based on Real Time Data. *CMCCOMPUTERS MATERIALS & CONTINUA, 68*(3), 3215-3233. https://doi.org/10.32604/cmc.2021.016881
21. Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for Bitcoin price prediction using economic and technology determinants. *International Journal of Forecasting, 37*(1), 28-43. https://doi.org/10.1016/j.ijforecast.2020.02.008
22. Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. *Neural computing and applications*, 1-15. https://doi.org/10.1007/s00521-020-05129-6
23. Mallqui, D. C., & Fernandes, R. A. (2019). Predicting the direction, maximum, minimum and closing prices of daily Bitcoin price using machine learning techniques. *Applied Soft Computing, 75*, 596-606. https://doi.org/10.1016/j.asoc.2018.11.038
24. Jang, H., & Lee, J. (2017). An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *IEEE Access*, *6*, 5427-5437. https://doi.org/10.1109/ACCESS.2017.2779181
25. Al-Nefaie, A. H., & Aldhyani, T. H. (2022). Bitcoin Price Forecasting and Trading: Data Analytics Approaches. *Electronics, 11*(24), 4088. https://doi.org/10.3390/electronics11244088
26. Cocco, L., Tonelli, R., & Marchesi, M. (2021). Predictions of bitcoin prices through machine learning based frameworks. *PeerJ Computer Science, 7*, e413. https://doi.org/10.7717/peerj-cs.413
27. Tapia, S., & Kristjanpoller, W. (2022). Framework based on multiplicative error and residual analysis to forecast bitcoin intraday-volatility. *Physica A: Statistical Mechanics and its Applications, 589*, 126613. https://doi.org/10.1016/j.physa.2021.126613
28. Dutta, A., Kumar, S., & Basu, M. (2020). A gated recurrent unit approach to bitcoin price prediction. *Journal of risk and financial management, 13*(2), 23. https://doi.org/10.3390/jrfm13020023
29. Lahmiri, S., & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals, 118*, 35-40. https://doi.org/10.1016/j.chaos.2018.11.014
30. Serrano, W. (2022). The random neural network in price predictions. *Neural Computing and Applications, 34*(2), 855-873. https://doi.org/10.1007/s00521-021-05903-0
31. Uras, N., Marchesi, L., Marchesi, M., & Tonelli, R. (2020). Forecasting Bitcoin closing price series using linear regression and neural networks models. *PeerJ Computer Science, 6*, e279. https://doi.org/10.7717/peerj-cs.279
32. Sebastião, H., & Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation, 7*(1), 1-30. https://doi.org/10.1186/s40854-020-00217-x
33. Poongodi, M., Sharma, A., Vijayakumar, V., Bhardwaj, V., Sharma, A. P., Iqbal, R., & Kumar, R. (2020). Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system. Computers & Electrical Engineering, 81, 106527. https://doi.org/10.1016/j.compeleceng.2019.106527
34. Zoumpekas, T., Houstis, E., & Vavalis, M. (2020). Eth analysis and predictions utilizing deep learning. *Expert Systems with Applications, 162*, 113866. https://doi.org/10.1016/j.eswa.2020.113866
35. Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A deep learning-based cryptocurrency price prediction scheme for financial institutions. *Journal of information security and applications*, *55*, 102583. https://doi.org/10.1016/j.jisa.2020.102583
36. Peng, Y., Albuquerque, P. H. M., de Sá, J. M. C., Padula, A. J. A., & Montenegro, M. R. (2018). The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression. *Expert Systems with Applications, 97*, 177-192. https://doi.org/10.1016/j.eswa.2017.12.004
37. Fabozzi, F. J., Fallahgoul, H., Franstianto, V., & Loeper, G. (2024). Asymptotic Properties of ReLU FFN Sieve Estimators. *Studies in Nonlinear Dynamics & Econometrics*, (0). https://doi.org/10.1515/snde-2023-0072

38. Konak, F., Bülbül, M. A., & Türkoğlu, D. (2024). Feature selection and hyperparameters optimization employing a hybrid model based on genetic algorithm and artificial neural network: Forecasting dividend payout ratio. *Computational Economics, 63*(4), 1673-1693. https://doi.org/10.1007/s10614-023-10530-z
39. Fraszka-Sobczyk, E., & Zakrzewska, A. (2024). The impact of foreign stock market indices on predictions volatility of the WIG20 index rates of return using neural networks. *Computational Economics*, 1-14. https://doi.org/10.1007/s10614-024-10662-w
40. Kayim, F., & Yilmaz, A. (2022). Time series forecasting with volatility activation function. *IEEE Access, 10*, 104000-104010. https://doi.org/10.1109/ACCESS.2022.3211312
41. Tripathi, B., & Sharma, R. K. (2023). Modeling bitcoin prices using signal processing methods, bayesian optimization, and deep neural networks. *Computational Economics, 62*(4), 1919-1945. https://doi.org/10.1007/s10614-022-10325-8
42. Sbrana, A., & Lima de Castro, P. A. (2024). N-BEATS perceiver: a novel approach for robust cryptocurrency portfolio forecasting. *Computational Economics, 64*(2), 1047-1081. https://doi.org/10.1007/s10614-023-10470-8
43. Bai, Y., Xie, J., Liu, C., Tao, Y., Zeng, B., & Li, C. (2021). Regression modeling for enterprise electricity consumption: A comparison of recurrent neural network and its variants. *International Journal of Electrical Power & Energy Systems, 126*, 106612. https://doi.org/10.1016/j.ijepes.2020.106612
44. Xiao, H., Chen, Z., Cao, R., Cao, Y., Zhao, L., & Zhao, Y. (2022). Prediction of shield machine posture using the GRU algorithm with adaptive boosting: A case study of Chengdu Subway project. *Transportation Geotechnics, 37*, 100837. https://doi.org/10.1016/j.trgeo.2022.100837
45. Ortu, M., Uras, N., Conversano, C., Bartolucci, S., & Destefanis, G. (2022). On technical trading and social media indicators for cryptocurrency price classification through deep learning. *Expert Systems with Applications, 198*, 116804. https://doi.org/10.1016/j.eswa.2022.116804
46. Dai, Y., Zhou, Q., Leng, M., Yang, X., & Wang, Y. (2022). Improving the Bi-LSTM model with XGBoost and attention mechanism: A combined approach for short-term power load prediction. *Applied Soft Computing,* 109632. https://doi.org/10.1016/j.asoc.2022.109632
47. Zhao, L., Li, Z., Qu, L., Zhang, J., & Teng, B. (2023). A hybrid VMD-LSTM/GRU model to predict non-stationary and irregular waves on the east coast of China. *Ocean Engineering, 276*, 114136. https://doi.org/10.1016/j.oceaneng.2023.114136
48. Islam, M. S., & Hossain, E. (2021). Foreign exchange currency rate prediction using a GRU-LSTM hybrid network. *Soft Computing Letters, 3*, 100009. https://doi.org/10.1016/j.socl.2020.100009
49. Faru, S. H., Waititu, A., & Nderu, L. (2023). A Hybrid Neural Network Model Based on Transfer Learning for Forecasting Forex Market. *Journal of Data Analysis and Information Processing, 11*(2), 103-120. https://doi.org/10.4236/jdaip.2023.112007
50. Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review, 53*(4), 3007-3057. https://doi.org/10.1007/s10462-019-09754-z
51. Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences, 10*(5), 1897. https://doi.org/10.3390/app10051897
52. CoinMarketCap. (2025). www.coinmarketcap.com