



Latest updates: <https://dl.acm.org/doi/10.1145/3699966>

RESEARCH-ARTICLE

Deep Learning Algorithms for Cryptocurrency Price Prediction: A Comparative Analysis

ARMIN MAZINANI, University of Parma, Parma, PR, Italy

LUCA DAVOLI, University of Parma, Parma, PR, Italy

GIANLUIGI FERRARI, University of Parma, Parma, PR, Italy

Open Access Support provided by:

University of Parma



PDF Download
3699966.pdf
30 December 2025
Total Citations: 3
Total Downloads: 4376

Published: 08 February 2025
Online AM: 12 October 2024
Accepted: 27 September 2024
Revised: 11 September 2024
Received: 08 December 2023

[Citation in BibTeX format](#)

Deep Learning Algorithms for Cryptocurrency Price Prediction: A Comparative Analysis

ARMIN MAZINANI, LUCA DAVOLI, and GIANLUIGI FERRARI, Internet of Things (IoT) Lab,
Department of Engineering and Architecture, University of Parma, Parma, Italy

Over the past years, cryptocurrencies have experienced a surge in popularity within the financial markets. As of today, besides being considered for investment purposes, they also serve as a widely accepted form of currency for everyday transactions. Due to the intricate characteristics of financial markets and their dependence on various factors to determine the prices of stocks and assets, the ability to predict such prices is crucial to make investment choices, especially in terms of cryptocurrencies. In this work, a comparative analysis on the suitability of Deep Learning (DL) algorithms (effective for time series forecasting) in predicting the price of three cryptocurrencies (namely Bitcoin, BTC; Ethereum, ETH; and Ripple, XRP) is assessed in terms of both short-term and long-term prediction accuracy. The results, evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (denoted as R^2), reveal that: Transformer is generally more effective for short-term forecasts and also performs well for long-term predictions; Convolutional Neural Network-Recurrent Neural Network (CNN-RNN) demonstrates the lowest complexity in terms of number of Multiply and Accumulate (MAC) operations; SimpleRNN has the fewest parameters and the smallest FLASH memory requirement. Overall, CNN-Gated Recurrent Unit (CNN-GRU) provides the best joint accuracy-complexity for predicting BTC and ETH prices, whereas CNN-RNN yields superior results for XRP price prediction.

CCS Concepts: • Computing methodologies → Machine learning; Neural networks; • Mathematics of computing → Time series analysis;

Additional Key Words and Phrases: Cryptocurrency, Deep Learning (DL), Forecasting, Price Prediction, CNN, RNN, GRU

ACM Reference format:

Armin Mazinani, Luca Davoli, and Gianluigi Ferrari. 2025. Deep Learning Algorithms for Cryptocurrency Price Prediction: A Comparative Analysis. *Distrib. Ledger Technol.* 4, 1, Article 5 (February 2025), 38 pages.
<https://doi.org/10.1145/3699966>

1 Introduction

Unlike a historical model of economy where traditional currencies (managed by governments) are dominant, new ways to exchange and transfer money have appeared, especially owing to cryptocurrencies. In general, cryptocurrencies can be seen as decentralized encrypted digital coins designed to be traded without an intermediate monetary authority (e.g., a central bank) and used on the Internet [40]. Thanks to their decentralized, secure, and self-managed nature, nearly 9,000 cryptocurrencies have been created until 2023, eventually gaining significant

Authors' Contact Information: Armin Mazinani, Internet of Things (IoT) Lab, Department of Engineering and Architecture, University of Parma, Parma, Italy; e-mail: armin.mazinani@unipr.it; Luca Davoli (corresponding author), Internet of Things (IoT) Lab, Department of Engineering and Architecture, University of Parma, Parma, Italy; e-mail: luca.davoli@unipr.it; Gianluigi Ferrari, Internet of Things (IoT) Lab, Department of Engineering and Architecture, University of Parma, Parma, Italy; e-mail: gianluigi.ferrari@unipr.it.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2025 Copyright held by the owner/author(s).

ACM 2769-6480/2025/2-ART5

<https://doi.org/10.1145/3699966>

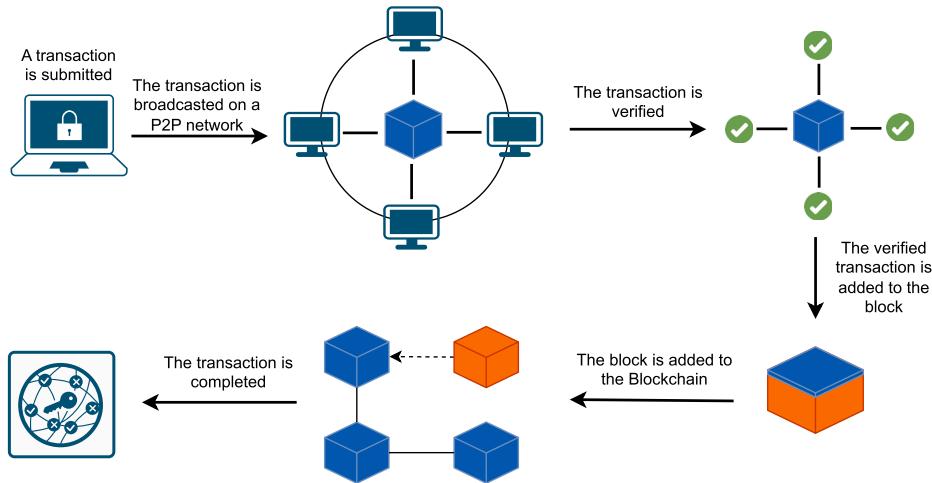


Fig. 1. High-level representation of a generic blockchain-based transaction flow.

attention from a wide plethora of investors and having a strong impact on the global financial market. As an example, the cryptocurrencies' market value overcame \$1.48 trillion in December 2023 (with its highest market cap, about \$2.82 trillion, in November 2021) [13]. Among the existing cryptocurrencies, **BiTCoin (BTC)** is so far one of the most popular and represents, as a Peer-to-Peer electronic currency exploiting core cryptographic mechanisms, the first cryptocurrency (introduced in 2008 [32]). Another technology strictly related to several cryptocurrencies (and required in order to govern them) is **Distributed Ledger Technology (DLT)** [46], based on the well-known paradigm embodied by blockchain. In fact, blockchain was originally designed to be exploited as an efficient (and trustable) way to record financial transactions. As shown in Figure 1, blockchain exploits cryptographic hashing techniques and a distributed network to force the persistence of all digital transactions as immutable, in the end ensuring (i) reliable transaction tracing, (ii) security maximization, and (iii) removal of reliance on third-party intermediaries [56].

More in detail, the benefits of blockchain and DLT—which transform data management from centralized to decentralized—support a wide range of applications, including banking, agriculture, intelligent transportation, energy, and supply chain management [17]. As an example, in [41], the authors introduce a blockchain-based framework designed for managing patients' medical records and the corresponding medical supply chain, in particular exploiting a Hyperledger Fabric network to ensure node confidentiality. Similarly, a blockchain-based Web application for generating medical certificates is proposed in [45], allowing users to manage their healthcare information through **Internet of Things (IoT)** devices and operating on an **ETHereum (ETH)** blockchain to prevent fraud in medical records. To this end, patients are provided credentials by a hospital authority and their data are verified on the blockchain before medical services are provided. Focusing on **Intelligent Transportation Systems (ITSs)**, in [30], an architecture securing data sharing within ITSs, using a combination of edge computing and DLT, is presented. This system, built on an ETH network and processing data closer to their sources, allows to reduce the volume of data to be sent to a central cloud.

As mentioned before, DLTs offer significant benefits, including high transparency, reduced supply chain costs, and enhanced safety, by tracking products from the origin to the customers [48]. As an example, in [4, 33], the authors provide an effective strategy for updating livestock agriculture by combining IoT and blockchain technologies. In detail, a cloud-based Smart Livestock Farming-oriented management system, composed by IoT wearable sensors to track livestock health in real time, is proposed, together with a blockchain-powered platform

(i) providing secure and transparent electronic record-keeping and (ii) removing the necessity for middlemen, and (iii) boosting trust between the involved parties.

As can be understood from the variability in the aforementioned market cap, cryptocurrency-based financial markets follow arbitrary and uncertain trends: this can only attract and make investors (of every nature) eager to find models to detect and predict financial market's trends. On a more technical side, this is due to the fact that the value of a cryptocurrency heavily depends on several factors, identifiable as both *internal* and *external*: demand and supply represent the most important *internal factors* affecting the price of cryptocurrencies (and, in general, of every kind of standard currency); political decisions and the degree of attractiveness of the cryptocurrency market, as well as the states of other markets (such as gold, regular stock, oil, etc.), are in general envisioned as *external factors* [38]. In the end, this highlights—one more time—how, on the basis of the price fluctuations in the cryptocurrency market, accurate price prediction is crucial to reduce the investment risk.

From a more statistical point of view, predicting the price of a cryptocurrency can be considered as a time series prediction, with forecasts being based on historical events over a proper time interval. Thus, given that time series prediction is currently used in a variety of applications in several fields—e.g., signal processing [54], weather forecasting [18], disease prediction [2], computer networks [7], and business planning [5]—the adoption of **Deep Learning (DL)** algorithms for prediction purposes can be considered a natural fit for time series. This is further motivated by the fact that, in recent years, DL algorithms have been widely used to extract features from financial time series and make predictions, since they (i) can be trained and operate with large amounts of multidimensional data and (ii) can learn non-linear dependencies between variables [55]. Indeed, financial time series are well-known as being non-linear, non-stationary, and depending on multiple variables, in the end featuring intricate trends that should be taken into account by an accurate (and complicated) forecasting model. This is what makes forecasting financial time series challenging.

Besides DL, traditional and Machine Learning algorithms are also commonly used to forecast time series. More in detail, *traditional approaches* (e.g., Automatic Regression Integrated Moving Average, ARIMA, and Auto-Regressive Moving Average, ARMA [29]) use statistical methods to obtain patterns in time series, but they are affected by some limitations. As an example, considering a linear relationship between the variables makes them ineffective in predicting and analyzing time series with non-linear data, in the end reducing the accuracy of long-term forecasts with low accuracy and making them unsuitable for multidimensional big data [23].

In this work, a comparative performance analysis and evaluation of different DL algorithms in predicting the price of three well-known cryptocurrencies (namely BTC, ETH, and Ripple, XRP) is proposed. In detail, several “single” DL algorithms (namely **Multilayer Perceptron (MLP)**; **Simple Recurrent Neural Network (SimpleRNN)**; **Long Short Term Memory (LSTM)**; **Bidirectional LSTM (BiLSTM)**; **Gated Recurrent Unit (GRU)**; **Bidirectional GRU (BiGRU)**; **Convolutional Neural Network (CNN)**; Transformer) as well as concatenated DL mechanisms (namely CNN-RNN and CNN-GRU) are considered. Their prediction accuracy is investigated in terms of **Root Mean Square Error (RMSE)**, **Mean Absolute Error (MAE)**, **Mean Absolute Percentage Error (MAPE)**, and coefficient of determination (denoted as R^2). The performance is analyzed considering various values of time lag, corresponding to the number of past observations considered to predict the price at the next time epoch. The Adam optimizer [24] is applied to improve the estimation accuracy of the considered DL methods for both short- and long-term cryptocurrencies' prices predictions.

The rest of this article is organized as follows. Section 2 summarizes various related works, while in Section 3 an overview on the considered DL algorithms is given. Section 4 and Section 5 present and discuss the proposed models and the experimental results, respectively. Finally, in Section 6, conclusions are drawn.

2 Related Works

Looking at the adoption of AI-based techniques in the context of cryptocurrencies, in [15], different AI mechanisms are used to improve the privacy and the effectiveness of the **Physical Unclonable Function (PUF)** electronic

cash system. In detail, the authors in [15] focus on this PUF system due to the fact that it makes use of physical unduplicatable functions in order to facilitate authentication and encryption procedures in the case different entities and multiple trusted third parties should be involved. Similarly, in [39], the authors consider various Artificial Neural Networks-based mechanisms aiming at the next day's BTC price prediction, training and validating their approach through the use of historical daily data (related to the time interval 2013–2017): a Back Propagation Neural Network is shown to return the best prediction performance. Furthermore, in [35], LSTM is evaluated and compared with different ARIMA-based hybrid models, showing that LSTM outperforms, in terms of the BTC price's prediction accuracy, the other methods. A similar evaluation (with similar outcomes) is proposed in [26], in which different Generalized Regression Neural Network models are compared with LSTM aiming at predicting the price of BTC, Digital Cash, and XRP, on the basis of a dataset related to the time interval July 2010–October 2018.

In [34], a BiLSTM is used on a BTC daily price dataset (containing data referring to the time interval January 2012–September 2020) in order to predict BTC prices: the final results show that the BiLSTM model provides more accurate predictions than linear regression-based models. In a similar way, the authors in [28] use a daily price dataset (referring to the time interval January 2014–October 2017) to evaluate the impact of the number of LSTM features on the BTC price prediction, considering single and multiple features. In the end, by taking into account multiple features, LSTM is shown to provide more accurate predictions at a significantly reduced error rate.

In [44], LSTM, GRU and BiLSTM models are adopted to predict the prices of BTC, ETH, and Litecoin, on the basis of their market capitalization, obtaining that BiLSTM returns the most precise forecasts for all the considered cryptocurrencies—in detail, with reference to RMSE and MAPE, considering data in the time interval January 2021–January 2023. Similarly, a Multi-Layer GRU model forecasting the prices of BTC, ETH, and Dogecoin, is introduced in [37]. In detail, this model consists of three hidden layers (each containing 200, 100, and 50 neurons, respectively): according to the presented results, the Multi-Layer GRU returns (on average) an RMSE 23 times lower than that of the LSTM model and nearly 4 times lower than that of the GRU model.

A comparison on the efficiencies of LSTM and ARIMA for short-term BTC price prediction (using data in the time interval December 2020–December 2021) is investigated in [27], obtaining that LSTM significantly enhances RMSE and MAE by 83.84% and 84.84%, respectively. Instead, encoder-decoder (also denoted as **AutoEncoders (AEs)**) predictive models for financial markets price prediction are proposed in [14]. In detail, the performance of AE-GRU and AE-LSTM is analyzed in forecasting stock prices, stock indexes, and cryptocurrencies—in particular, BTC pricing stretching in the time interval 2014–2022. Based on the achieved results, AE-GRU outperforms AE-LSTM in lowering MAPE by 50%.

The use of RNNs and MLP for predicting short- and long-term BTC prices, exploiting a 7-year dataset collected in the time interval August 2010–October 2017 at 2-day intervals, is discussed in [3], returning that MLP outperforms RNNs in the long-term BTC price prediction.

In [31], a Transformer model is employed to predict stock prices on the Dhaka Stock Exchange, in particular using Time2Vec encoding [19] to manage time series data (namely, both historical daily and weekly data). The experimental results indicate that the Transformer model returns promising predictions, achieving a lower RMSE with respect to ARIMA. Finally, in [52], the authors compare various DL models (including LSTM, RNN, CNN, and Transformer) for predicting major stock market indices (namely CSI 300, S&P 500, Hang Seng Index, Nikkei 225). The obtained results show that Transformer outperforms the other considered DL models, providing the lowest errors in terms of MSE, MAE, and MAPE.

3 Background

For the sake of clarity, before carrying out a comparative performance analysis of different DL algorithms applied for cryptocurrencies price prediction, it is useful to overview the considered DL models, highlighting their strengths and weaknesses.

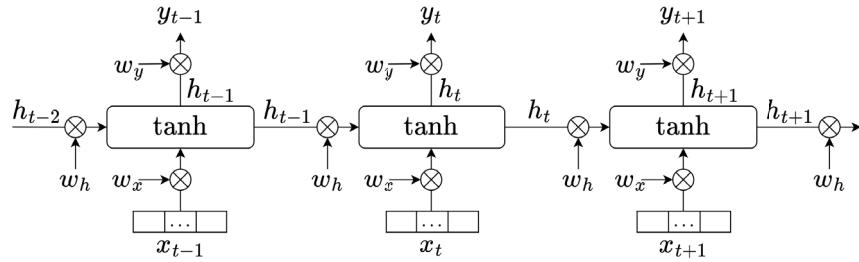


Fig. 2. Simplified mathematical model of a SimpleRNN.

3.1 SimpleRNNs

On the basis of the fact that (in the past) the price prediction of financial markets has been based on the trend of their price changes, there is a strong need to exploit algorithms with memory and able to store past prices. Since **Convolution Neural Networks** (CNNs) do not give this possibility, RNNs have been considered. More in detail, a SimpleRNN is an NN used to process and capture relationships among sequential data (e.g., time series). Then, in order to understand the relationship between previous and current data, SimpleRNN performs recursively, operating one operation (per series' element) to calculate (per neuron) the current (at time t) output (y_t) and the current hidden state value (h_t), depending on the current input features (x_t) and the previous (at time $t - 1$) hidden state value (h_{t-1}). We remark that all these quantities can be, in the most general case, vectors. In fact, the hidden state acts as a memory to store previous information for future predictions. In detail, as shown in Figure 2, a SimpleRNN performs the following operations:

$$h_t = \tanh(\mathbb{W}_h h_{t-1} + \mathbb{W}_x x_t), \quad (1)$$

$$y_t = \mathbb{W}_y h_t, \quad (2)$$

where \mathbb{W}_h , \mathbb{W}_y , and \mathbb{W}_x are weight matrices for hidden, output, and input states, respectively; and $\tanh(x) \triangleq (e^{2x} - 1)/(e^{2x} + 1)$ is used for data normalization (element-wise for the vector), returning a predicted vector h_t with elements in the (real) interval $[-1, 1]$.

SimpleRNNs are not efficient in learning long-term correlations because of vanishing or exploding gradient problems in long sequences [6]—these problems correspond to drastic decreases or increases observed at each layer during back-propagation. In order to overcome these SimpleRNNs' drawbacks and limitations, LSTM [16] and GRU [9] have been proposed.

3.2 LSTM Networks

An LSTM network corresponds to an advanced version of an RNN able to detect long-term dependencies using cell state, in addition to the hidden state used in RNNs. As detailed in Section 3.1, the short-term memory and the long-term memory are represented by the *hidden state* and the *cell state*, respectively, with each LSTM cell consisting of three types of gates determining the importance of the information to be remembered or forgotten: (i) the *forget gate*, determining whether information from previous time steps is retained or forgotten; (ii) the *input gate*, specifying which new information has to be stored in the cell state; and (iii) the *output gate*, determining the final output of the LSTM cell at each time step t . In detail, an LSTM network performs the following operations:

$$f_t = \sigma_g(\mathbb{W}_f x_t + \mathbb{U}_f h_{t-1} + b_f), \quad (3)$$

$$i_t = \sigma_g(\mathbb{W}_i x_t + \mathbb{U}_i h_{t-1} + b_i), \quad (4)$$

$$\mathbf{o}_t = \sigma_g(\mathbb{W}_o \mathbf{x}_t + \mathbb{U}_o h_{t-1} + \mathbf{b}_o), \quad (5)$$

$$\tilde{\mathbf{c}}_t = \sigma_c(\mathbb{W}_c \mathbf{x}_t + \mathbb{U}_c h_{t-1} + \mathbf{b}_c), \quad (6)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot c_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \quad (7)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \sigma_h(\mathbf{c}_t), \quad (8)$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is the input vector of the LSTM cell; $\mathbf{f}_t \in (0, 1)^h$ is the forget gate's activation vector; $\mathbf{i}_t \in (0, 1)^h$ is the input/update gate's activation vector; $\mathbf{o}_t \in (0, 1)^h$ is the output gate's activation vector; $\mathbf{h}_t \in (-1, 1)^h$ is the hidden state vector; $\tilde{\mathbf{c}}_t \in (-1, 1)^h$ is the cell input activation vector; $\mathbf{c}_t \in \mathbb{R}^h$ is the cell state vector; $\mathbb{W} \in \mathbb{R}^{h \times d}$, $\mathbb{U} \in \mathbb{R}^{h \times h}$, and $\mathbf{b} \in \mathbb{R}^h$ are weight matrices and bias vector; d and h refer to the number of input features and the number of hidden units, respectively; the initial values are $c_0 = 0$ and $h_0 = 0$; and the operator \odot denotes the element-wise product.

3.3 GRU Networks

A GRU network can be seen as an improved version of an LSTM network, but with a simpler structure, since (unlike LSTM) the memory cell is not used, and the forget gate is removed to reduce complexity. Therefore, a GRU cell includes the following two gates: (i) the *update gate*, determining how much of the past information (from previous time steps) needs to be passed to the next GRU cell; (ii) the *reset gate*, deciding how much of the past information (from the previous steps) will be disregarded. In detail, a GRU network performs the following operations:

$$\mathbf{z}_t = \sigma(\mathbb{W}_z \mathbf{x}_t + \mathbb{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z), \quad (9)$$

$$\mathbf{r}_t = \sigma(\mathbb{W}_r \mathbf{x}_t + \mathbb{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \quad (10)$$

$$\hat{\mathbf{h}}_t = \phi(\mathbb{W}_h \mathbf{x}_t + \mathbb{U}_h (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \quad (11)$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \hat{\mathbf{h}}_t, \quad (12)$$

where \mathbf{x}_t and \mathbf{h}_t are input and output vectors, respectively; $\hat{\mathbf{h}}_t$ is the candidate activation vector; \mathbf{z}_t is the update gate vector; \mathbf{r}_t is the reset gate vector; \mathbb{W} , \mathbb{U} , and \mathbf{b} are weight matrices and bias vector.

3.4 CNNs

CNNs are known for their extensive image processing capabilities, thanks to their well-known automatic feature extraction capability from high-dimensional data (e.g., images and videos). More in detail (and from a more general point of view), a CNN consists of two main layers, denoted as *convolution* and *pooling*. Convolution is applied to the input data using a kernel to produce a feature map, with the kernel size usually smaller than the input data. Then, after a convolution operation, pooling is performed to reduce the dimension. In general, 1D CNNs are mostly used for time series data, as shown in Figure 3.

3.5 Bidirectional RNNs

Unlike the DL models discussed in Subsection 3.1–3.3 (namely SimpleRNN, LSTM, and GRU), in a **Bidirectional RNN (BiRNN)** data sequences are processed in two directions, namely *forward* and *backward*. This requires to adopt and use two layers: the *first* layer uses the input data from start to end, while the *second* layer processes the input data reversely. Therefore, using two RNN layers to bidirectionally process the input data is expected to enhance the prediction accuracy.

3.6 Transformers

The Transformer model [36, 51], introduced in 2017, revolutionized the processing of sequential data, with particular regard to **Natural Language Processing (NLP)** tasks. This is due to the fact that, unlike traditional

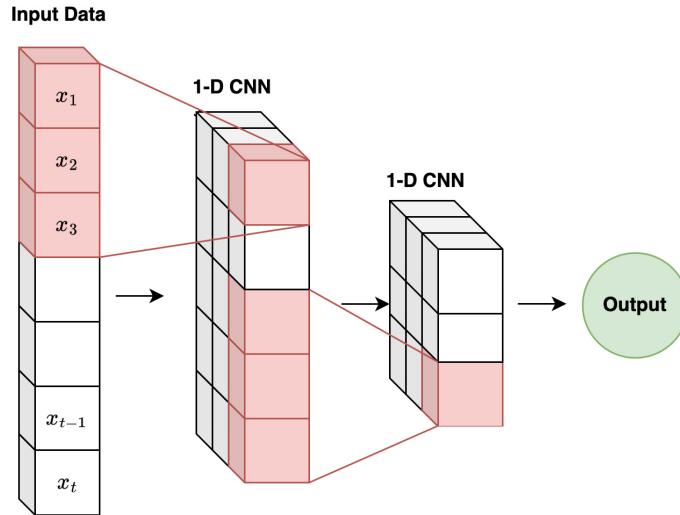


Fig. 3. 1D CNN architecture featuring two convolutional layers.

sequence models (e.g., RNNs), Transformers are better suited to handle long-range dependencies. This is mainly due to their reliance on the so-called *attention mechanisms* rather than sequential processing. In particular, a key component of the Transformer is the *scaled dot-product attention mechanism*, allowing this model to dynamically weigh the importance of different tokens in the input sequence—e.g., single words in a sentence to be analyzed by an NLP system.

From an analytical point of view, a Transformer operates by transforming the input into three matrices: *query* (denoted as \mathbb{Q}), *key* (denoted as \mathbb{K}), and *value* (denoted as \mathbb{V}) matrices. In detail, these matrices are expedient to calculate the following attention function:

$$\text{Attention}(\mathbb{Q}, \mathbb{K}, \mathbb{V}) = \text{softmax}\left(\frac{\mathbb{Q} \cdot \mathbb{K}^T}{\sqrt{d_k}}\right) \mathbb{V}, \quad (13)$$

where \mathbb{Q} represents the query matrix, corresponding to a token in the input sequence; \mathbb{K} corresponds to the key matrix; \mathbb{V} represents the value matrix; the term $1/\sqrt{d_k}$ (where d_k represents the dimensionality of the keys) serves as scaling factor, ensuring stable gradients by normalizing the dot product between \mathbb{Q} and \mathbb{K}^T .

It should be highlighted how Transformers use multiple instances of scaled dot-product attention mechanisms, obtaining the so-called *multi-head attention*. More in detail, on the basis of this mechanism, several attention functions (denoted as attention heads) are calculated in parallel, each using a different projection of the $(\mathbb{Q}, \mathbb{K}, \mathbb{V})$ matrices tuple. Thus, each attention head operates in a lower-dimensional subspace, enabling the Transformer model to capture different aspects of the input sequence, while the outputs from these parallel attention functions are then concatenated and passed through a linear layer to produce the final output. Hence, the multi-head attention mechanism can be analytically represented as

$$\text{MultiHead}(\mathbb{Q}, \mathbb{K}, \mathbb{V}) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) \mathbb{W}^{(O)}, \quad (14)$$

where h corresponds to the number of attention heads, and $\mathbb{W}^{(O)}$ corresponds to a learnable output weight matrix [1, 47, 53].

For the sake of clarity, a graphical representation of scaled dot-product attention and multi-head attention mechanisms is shown in Figure 4.

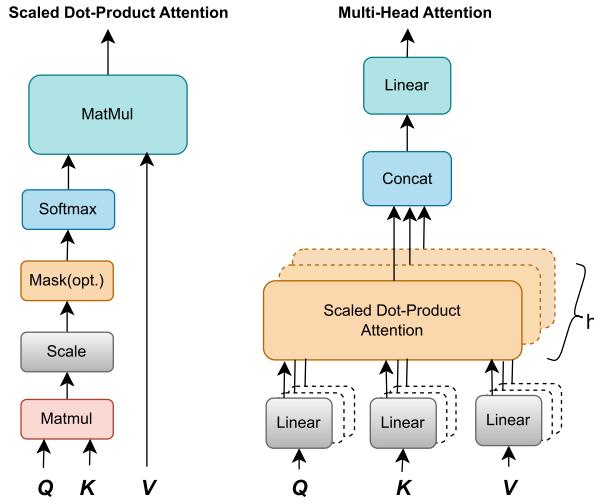


Fig. 4. Scaled dot-product (left) and multi-head attention (right) mechanisms.

Table 1. Summary of the Dataset Information

Observations interval [hour]	Training data size [record]	Test data size [record]	Features [num]
4	4,000	1,000	4

4 Evaluation

As anticipated in Section 1, the DL models presented in Section 3 are evaluated in a comparative way. Their performance is evaluated in terms of prediction accuracy of BTC, ETH, and XRP cryptocurrencies’ price, considering different lag sizes for short-term and long-term forecasting.

4.1 Dataset and Proposed Model

In order to conduct this analysis, three datasets, each containing 5,000 samples resulting from three cryptocurrencies data obtained from the twelvedata cryptocurrency trading platform [50], have been created. In particular, the data have been obtained exploiting the RESTful APIs provided by twelvedata, which allow to retrieve historical price data on the basis of proper query parameters using a predefined interval.¹ The ETH and XRP price information refer to the time interval August 2020–November 2022, while the BTC price information refer to the time interval November 2020–March 2023, with a 4-hour observation interval in each dataset. More in detail, as shown in Table 1, each dataset has been divided into a training set, consisting of 4,000 records, and a test set, composed by 1,000 records. Both sets contain the following 4 features: (i) the price at the start of each interval, denoted as Open Price; (ii) the highest price in each interval, denoted as High Price; (iii) the lowest price in each interval, denoted as Low Price; and (iv) the closing price at each interval, denoted as Close Price.

The DL models (namely MLP, SimpleRNN, LSTM, GRU, BiGRU, BiLSTM) share the same network architecture composed of three layers, with 50, 30, and 20 neurons, respectively. Instead, the CNN model is composed of two 1D-convolution windows (with a size equal to 2), each layer having 64 and 32 feature detectors, respectively, while a max pooling layer with size equal to 2 is used between the convolutional layers. Then, the **Rectified Linear Unit (ReLU)** activation function [10–12] (defined as $x^+ \triangleq \max(0, x)$, where x is the input of a neuron) is applied to input and hidden layers of the models. Finally, with regard to the proposed Transformer model, it includes (i)

¹Additional information on twelvedata APIs can be found at <https://twelvedata.com/docs>

Table 2. Network Architectures of the Evaluated DL Models

MLP	RNN	LSTM
Dense (layer_size = 50, activation = relu) Dropout (0.2)	SimpleRNN (layer_size = 50, activation = relu) Dropout (0.2)	LSTM (layer_size = 50, activation = relu) Dropout (0.2)
Dense (layer_size = 30, activation = relu) Dropout (0.2)	SimpleRNN (layer_size = 30, activation = relu) Dropout (0.2)	LSTM (layer_size = 30, activation = relu) Dropout (0.2)
Dense (layer_size = 20, activation = relu) Dropout (0.2)	SimpleRNN (layer_size = 20, activation = relu) Dropout (0.2)	LSTM (layer_size = 20, activation = relu) Dropout (0.2)
Dense (layer_size = s, activation = selu, kernel_regularizer = l2)	Dense (layer_size = s, activation = selu, kernel_regularizer = l2)	Dense (layer_size = s, activation = selu, kernel_regularizer = l2)
GRU	BiLSTM	BiGRU
GRU (layer_size = 50, activation = relu) Dropout (0.2)	BiLSTM (layer_size = 25, activation = relu) Dropout (0.2)	BiGRU (layer_size = 25, activation = relu) Dropout (0.2)
GRU (layer_size = 30, activation = relu) Dropout (0.2)	BiLSTM (layer_size = 25, activation = relu) Dropout (0.2)	BiGRU (layer_size = 15, activation = relu) Dropout (0.2)
GRU (layer_size = 20, activation = relu) Dropout (0.2)	BiLSTM (layer_size = 10, activation = relu) Dropout (0.2)	BiGRU (layer_size = 10, activation = relu) Dropout (0.2)
Dense (layer_size = s, activation = selu, kernel_regularizer = l2)	Dense (layer_size = s, activation = selu, kernel_regularizer = l2)	Dense (layer_size = s, activation = selu, kernel_regularizer = l2)
CNN	CNN-RNN	CNN-GRU
Conv1D (filters = 64, kernel_size = 2, activation = relu) Dropout (0.2)	Conv1D (filters = 64, kernel_size = 2, activation = relu) Dropout (0.2)	Conv1D (filters = 64, kernel_size = 2, activation = relu) Dropout (0.2)
Maxpooling1D Dropout (0.2)	Maxpooling1D Dropout (0.2)	Maxpooling1D Dropout (0.2)
Conv1D (filters = 32, kernel_size = 2, activation = relu) Dropout (0.2)	Conv1D (filters = 32, kernel_size = 2, activation = relu) Dropout (0.2)	Conv1D (filters = 32, kernel_size = 2, activation = relu) Dropout (0.2)
Dense (layer_size = s, activation = selu, kernel_regularizer = l2)	SimpleRNN (layer_size = 32, activation = relu) Dense (layer_size = s, activation = selu, kernel_regularizer = l2)	GRU (layer_size = 32, activation = relu) Dense (layer_size = s, activation = selu, kernel_regularizer = l2)

SELU, scaled exponential linear unit.

an embedding layer with 128 units to project input data into a suitable dimension, and (ii) a Transformer block with a multi-head attention mechanism using 4 attention heads and an embedding dimension equal to 128. Then, the attention mechanism is followed by dropout regularization, residual connections, and normalization layer. Finally, the block also contains a **Feed Forward Network (FFN)** with two dense layers (128 units each), using ReLU for the first layer and projecting back to the embedding dimension, while dropout is applied after the FFN to regularize the output.

Furthermore, the **Scaled Exponential Linear Unit (SELU)** activation function, defined as

$$f(\alpha, x) = \lambda \begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases},$$

with $\alpha \simeq 1.6733$ and $\lambda \simeq 1.0507$ [25], is used for the output layer of all the previous algorithms, while the Adam optimizer (detailed in Section 4.1.1) is used to minimize the prediction errors of the DL models. Finally, as listed in the detailed architectures of the proposed models in Tables 2 and 3, on the basis of experimental tuning, a learning rate equal to 0.001 and a batch size equal to 32 have been identified and tuned to train the NNs, as shown in Table 4. Then, in order to prevent overfitting during the training phase, dropout layers [20], L2 regularization [22], and early stopping [21] techniques have been applied.

Table 3. Network Architecture of the Evaluated Transformer Model, Detailing Its Composing Layers (Left) and Their Corresponding Description (Right)

Embedding Layer	
Dense (layer_size = 128)	Projects input data to embedding dimension
Transformer Block	
MultiHead ($h = 4$, key_dim = 128)	Computes self-attention
Dropout (0.1)	Regularizes the attention output
LayerNormalization()	Normalizes the sum of input and attention output
Dense (layer_size = 128, activation = relu)	First layer in FFN
Dense (layer_size = 128)	Projects back to embedding dimension
Dropout (0.1)	Regularizes the FFN output
LayerNormalization()	Normalizes the sum of input and FFN output
Fully Connected Layers	
Flatten()	Flattens the output for the final dense layers
Dropout (0.3)	Regularizes before the output layer
Dense (layer_size = s , activation = selu, kernel_regularizer = l2)	Output layer with L2 regularization

Table 4. Summary of the Adopted Hyperparameters

Parameters	Learning rate	Optimizer	Loss function	Batch size
Value	0.001	Adam	MSE	32

From an operational point of view, all the considered DL algorithms perform the following main prediction steps. In the *first* pre-processing step, missing values in each column of the dataset are filled with the closest previous value. *Then*, a MinMaxScaler normalization [43] is applied to fit data in the range [0, 1]—this takes into account the fact that the data of different columns have different distributions. As a matter of fact, this normalization action accelerates the learning process within the DL models. At this point, the dataset must be framed on the basis of the number of forecasting steps and the lag size, in order to perform time series forecasting. Then, as mentioned before, the dataset is divided into two sets (denoted as training and test), where the first is used to train the DL models, while the latter is used to perform cryptocurrencies' prices predictions. *Finally*, predicted and actual data are converted back to the original scale (using the inverse of the MinMaxScaler), and the difference between predicted and actual values is computed.

4.1.1 Adaptive Moment Estimation (Adam) Optimizer. Adam [24] is a computationally efficient optimization algorithm with low memory requirements that combines the benefits of RMSProp and momentum [42]. On the analytical side, in order to compute the learning rate for each weight, Adam exploits the mean (denoted as m_t) and the uncentered variance (denoted as v_t) of the gradients, corresponding to the first and second moments, respectively, as follows:

$$\mathbf{g}_t = \nabla \theta f_t(\theta_{t-1}), \quad (15)$$

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t, \quad (16)$$

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2, \quad (17)$$

where \mathbf{g}_t represents the gradient of the loss function f_t with respect to the parameters θ (weights and biases) at time step t , which are then updated during the training phase to optimize the model's performance; β_1 and β_2 are hyperparameters of the Adam algorithm, typically set to 0.9 and 0.999, respectively.

Then, in order to counteract biases in the moment estimates, especially during initial time steps, bias-corrected estimates are calculated as follows:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad (18)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}. \quad (19)$$

The parameter update rule in the Adam algorithm is the following:

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \quad (20)$$

where α is the learning rate and ϵ is a small constant (e.g., 10^{-8}) to prevent any division by zero.

4.2 Combination of CNN and RNN

The integration of CNNs with RNNs represents a highly effective approach in the field of the DL, since (as mentioned in Section 3.1 and Section 3.4) they are specifically designed for problems involving spatial and temporal inter-dependence—e.g., sequential data processing, such as generating captions for images, analyzing videos, and processing data in a sequential manner. Due to their strengths (especially if combined together), in this work, two additional hybrid DL models, involving a combination of 1D-CNN, SimpleRNN, and GRU are considered.

4.3 Loss Functions as Evaluation Metrics

In order to evaluate the prediction accuracy of a DL algorithm, a comparison between predicted and actual values needs to be carried out with proper performance metrics. In order to do this, we select, as loss functions, MAE, RMSE, MAPE, and R^2 , which are defined as follows:

$$\text{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) \triangleq \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}, \quad (21)$$

$$\text{MAE}(\mathbf{y}, \hat{\mathbf{y}}) \triangleq \frac{1}{N} \sum_{i=0}^{N-1} |y_i - \hat{y}_i|, \quad (22)$$

$$\text{MAPE}(\mathbf{y}, \hat{\mathbf{y}}) \triangleq \frac{1}{N} \sum_{i=0}^{N-1} \frac{|y_i - \hat{y}_i|}{\max(\epsilon, |y_i|)}, \quad (23)$$

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) \triangleq 1 - \frac{\text{SS}_{\text{res}}}{\text{SS}_{\text{tot}}} = 1 - \frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{N-1} (y_i - \bar{y})^2}, \quad (24)$$

where N is the total number of samples in the test set; $\mathbf{y} = (y_0, \dots, y_{N-1})$ is the actual (target) values vector; $\hat{\mathbf{y}} = (\hat{y}_0, \dots, \hat{y}_{N-1})$ is the vector of predicted values; and \bar{y} is the mean of the actual values. With regard to R^2 , when $\text{SS}_{\text{res}} > \text{SS}_{\text{tot}}$, then R^2 becomes negative: this indicates that the model's predictions are less accurate than a simple horizontal line representing the mean of the observed values [8]. Nevertheless, given that values scales' dependency might represent a significant drawback, it is important to highlight that data scaling would help in mitigating this undesired problem, ensuring that the metrics provide a fair assessment regardless of the magnitudes of the values in the dataset. This is the reason why, as detailed in Section 4.1, a MinMaxScaler normalization has been applied to fit data in the range $[0, 1]$. Then, while RMSE, MAE, and MAPE are known to be valuable for understanding error magnitude and relative error, R^2 was also included because it offers a

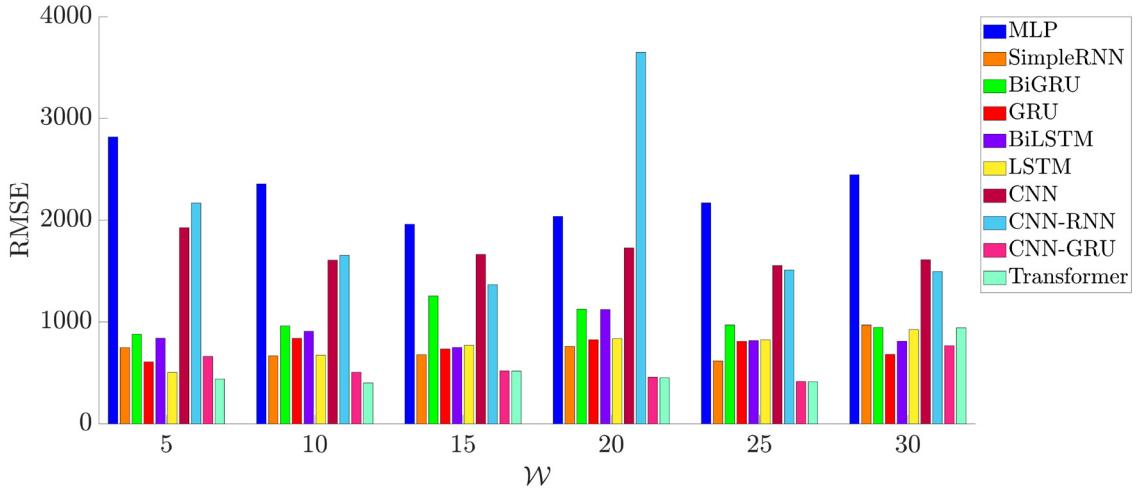


Fig. 5. Impact, in terms of RMSE, of the value of the time lag \mathcal{W} on the BTC price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

scale-invariant measure of how well the model captures the variance in the data. The combined use of all these evaluation metrics provides a comprehensive and balanced evaluation of the model's performance.

5 Experimental Results

On the basis of the evaluation metrics detailed in Section 4, in the following we compare and discuss the experimental results (in terms of cryptocurrencies' prices predictions) obtained by exploiting the different DL algorithms presented in the previous sections (namely, MLP, SimpleRNN, LSTM, GRU, BiLSTM, BiGRU, CNN, CNN-RNN, CNN-GRU, Transformer), for both short- and long-term forecasting. For the sake of completeness, since CNN-LSTM, CNN-BiLSTM, and CSS-BiGRU do not outperform the performance results discussed in the following, it has been preferred to present the results of these three additional algorithms (denoted as A1–A24) in Appendix A.

5.1 Accuracy

More in detail, Figures 5 and 6 represent the RMSE of the considered DL models for the BTC price prediction, considering a 3-step ahead short-term prediction horizon ($s = 3$) and various values of time lag (denoted as \mathcal{W}). In detail, \mathcal{W} refers to the number of previous time steps (or observations) considered as input for predicting the cryptocurrency values in next time steps. For the sake of clarity, with $\mathcal{W} = 5$ and $s = 3$, and considering that the reference dataset has a 4-hour data collection interval (as detailed in Section 4.1), the next 12-hour currency prices' prediction is based on the information from the past 20 hours. According to the obtained results, Transformer provides (in general) the lowest RMSE and MAE, while MLP and CNN-RNN return the highest RMSE and MAE across different lags. Moreover, as shown in Figure 7, Transformer can return a small MAPE considering large time lags (e.g., $\mathcal{W} = \{5, 10, 15, 20, 25\}$). The results of R^2 as the final evaluation metric are shown in Figure 8, where Transformer provides the highest value.

Similarly, the experimental performance results for the ETH price prediction, in terms of RMSE and MAE, considering a 3-step ahead short-term prediction horizon ($s = 3$) and various values of time lag \mathcal{W} , are shown in Figures 9 and 10, respectively. In general, on the basis of the experimental results, Transformer returns the lowest error rates if compared to the other DL methods for various time lags \mathcal{W} . The same conclusion arises from the results shown in Figure 11, with Transformer returning the lowest MAPE for $\mathcal{W} = \{5, 15, 20, 25, 30\}$, while

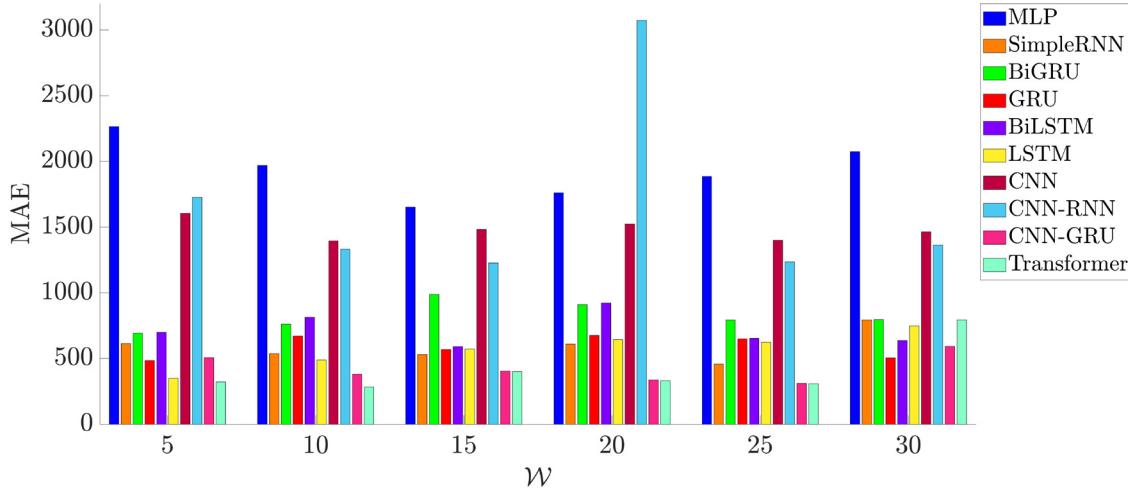


Fig. 6. Impact, in terms of MAE, of the value of the time lag \mathcal{W} on the BTC price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

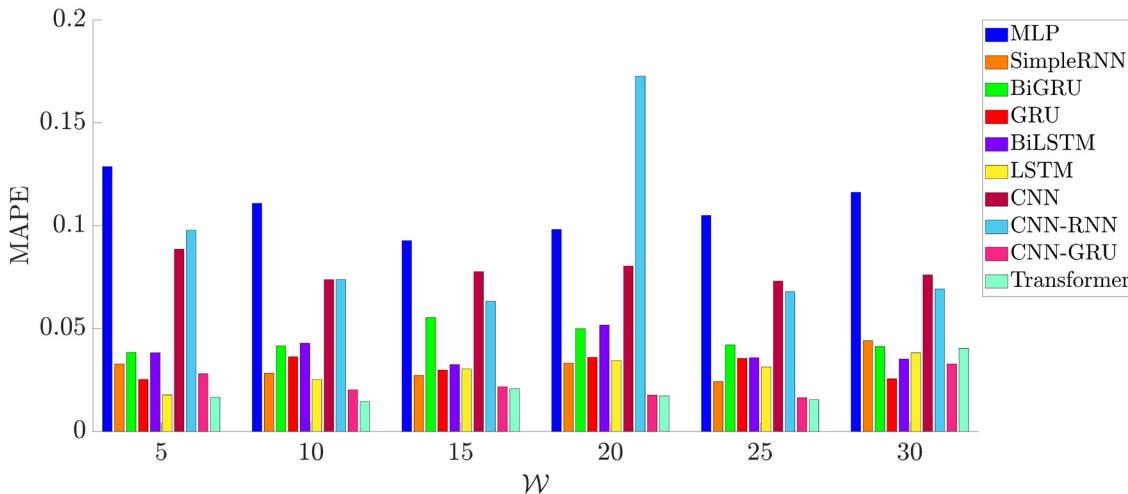


Fig. 7. Impact, in terms of MAPE, of the value of the time lag \mathcal{W} on the BTC price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

BiLSTM performing better for $\mathcal{W} = \{10\}$. The results of R^2 as the final evaluation metric are shown in Figure 12, where Transformer provides the highest value.

Finally, the experimental performance results for the XRP price prediction (in terms of RMSE and MAE) considering a 3-step ahead short-term prediction horizon ($s = 3$) and various values of time lag \mathcal{W} are shown in Figures 13 and 14, respectively. On the basis of the achieved results, Transformer seems to outperform the other considered DL algorithms. In addition, when examining the results shown in Figures 15, it can be observed that Transformer achieves the lowest MAPE for $\mathcal{W} \in \{5, 15, 20\}$, whereas CNN-GRU and BiLSTM outperform Transformer for $\mathcal{W} \in \{25, 30\}$ and $\mathcal{W} = 10$, respectively. As shown in Figure 16, the Transformer generally

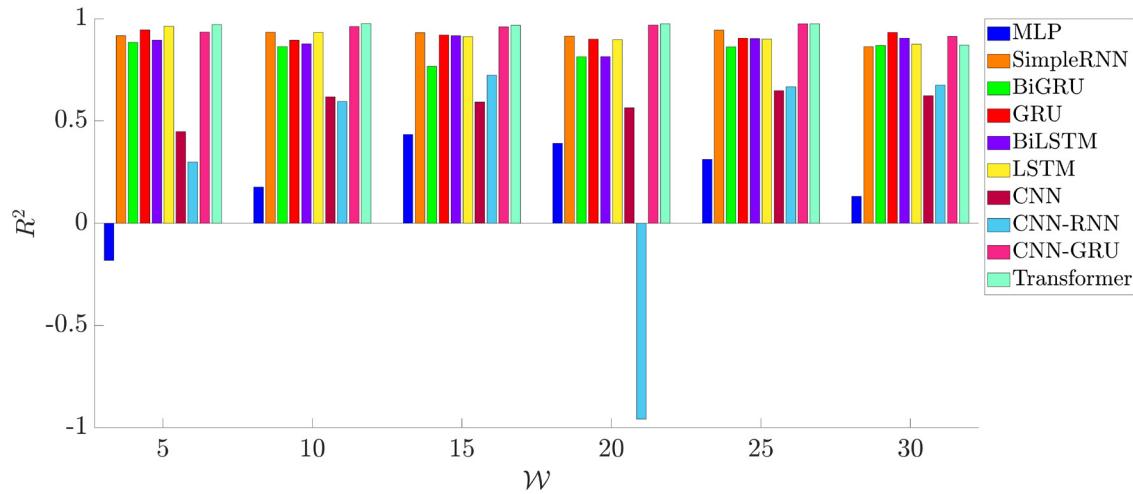


Fig. 8. Impact, in terms of R^2 , of the value of the time lag \mathcal{W} on the BTC price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

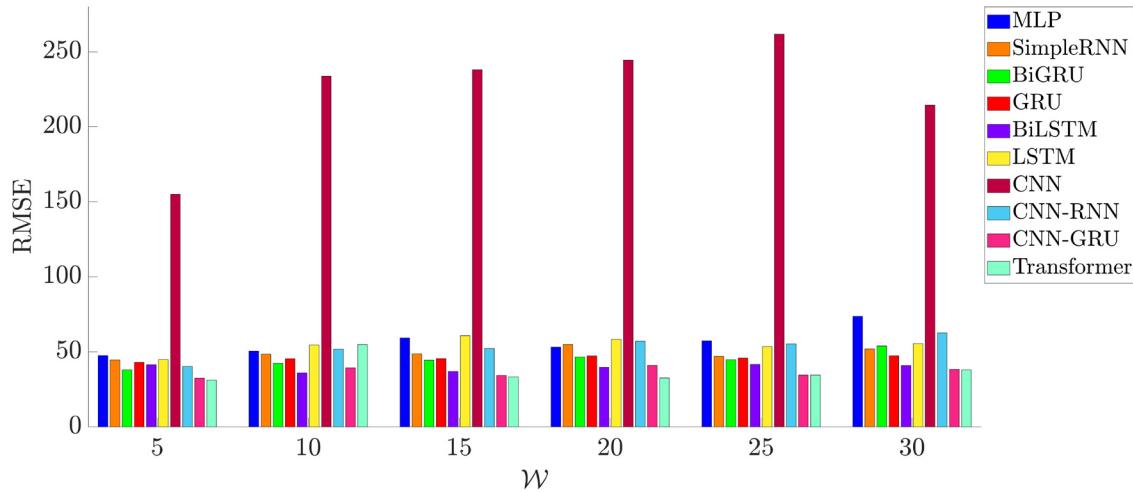


Fig. 9. Impact, in terms of RMSE, of the value of the time lag \mathcal{W} on the ETH price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

outperforms the other models, except when $\mathcal{W} \in \{10, 25, 30\}$, where BiLSTM and CNN-GRU achieve higher R^2 values.

For the sake of clarity, a summary of the best-performing DL algorithms, in terms of RMSE, is shown in Table 5, which highlights how Transformer consistently outperforms the other models in the majority of scenarios, confirming its robustness in capturing temporal dependencies for short-term predictions. To better highlight the performance and generalization capability of the considered models in predicting BTC, ETH, and XRP prices, the training and validation losses are shown in Figures 17, 18, 19, and 20.

Then, in order to evaluate the accuracy in the case of long-term cryptocurrencies prices forecasting, the prediction horizon has been extended to 10-, 15-, and 20-step ahead ($s \in \{10, 15, 20\}$), considering a fixed time lag

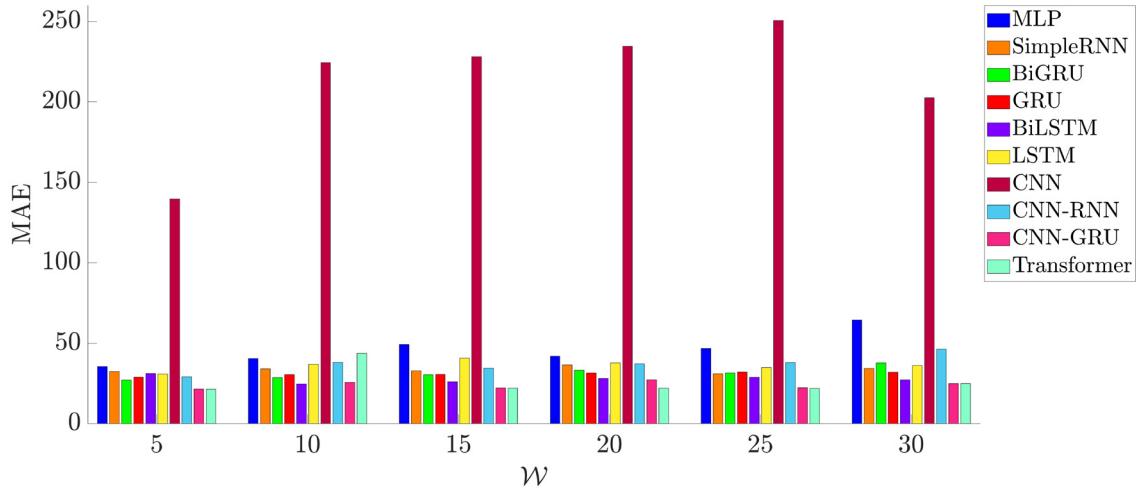


Fig. 10. Impact, in terms of MAE, of the value of the time lag \mathcal{W} on the ETH price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

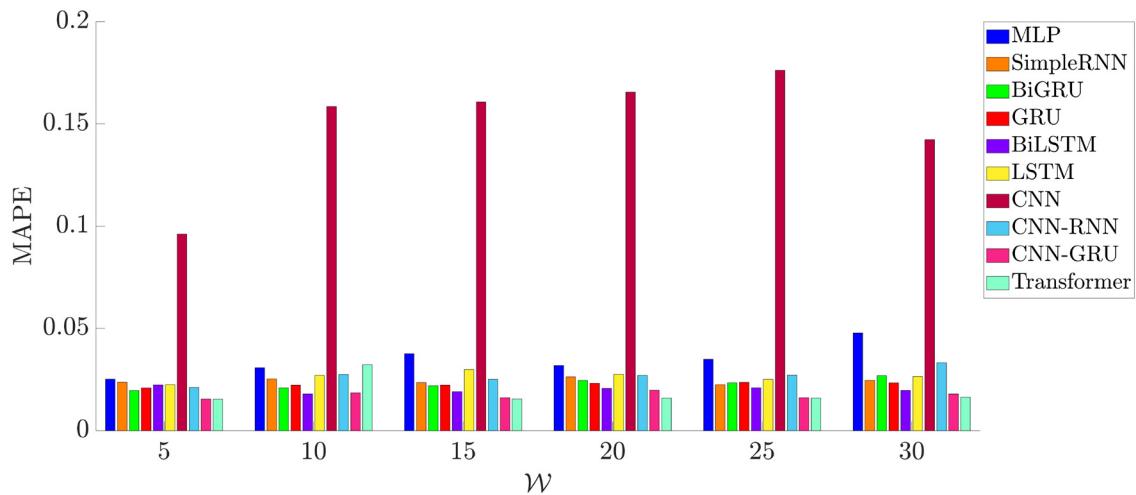


Fig. 11. Impact, in terms of MAPE, of the value of the time lag \mathcal{W} on the ETH price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

$\mathcal{W} = 25$. In particular, after experimental tuning activities, $\mathcal{W} = 25$ has been selected since it represents a good tradeoff between low values of the error metrics (MSE, RMSE, MAE, MAPE) and high value of R^2 , thus indicating accurate and reliable predictions.

At this point, recalling that the collection time interval in the dataset is equal to 4 hours, the considered values of s correspond to cryptocurrencies' price predictions for the next 40, 60, and 80 hours, respectively. The experimental findings achieved with this long-term prediction assessment in terms of RMSE for BTC, ETH, and XRP, are presented in Figures 21, 25, and 29, respectively. On the basis of these results and as described in Table 6, Transformer is more accurate (if compared to the other evaluated DL models) for long-term price prediction. Furthermore, the results of the comparison, in terms of MAE, of BTC, ETH, and XPR are shown in Figures 22, 26,

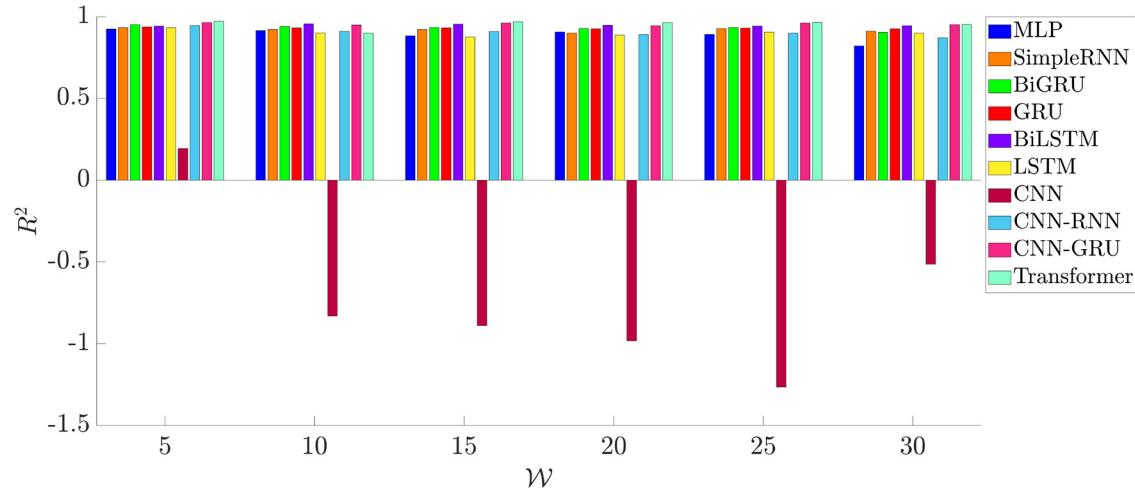


Fig. 12. Impact, in terms of R^2 , of the value of the time lag W on the ETH price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

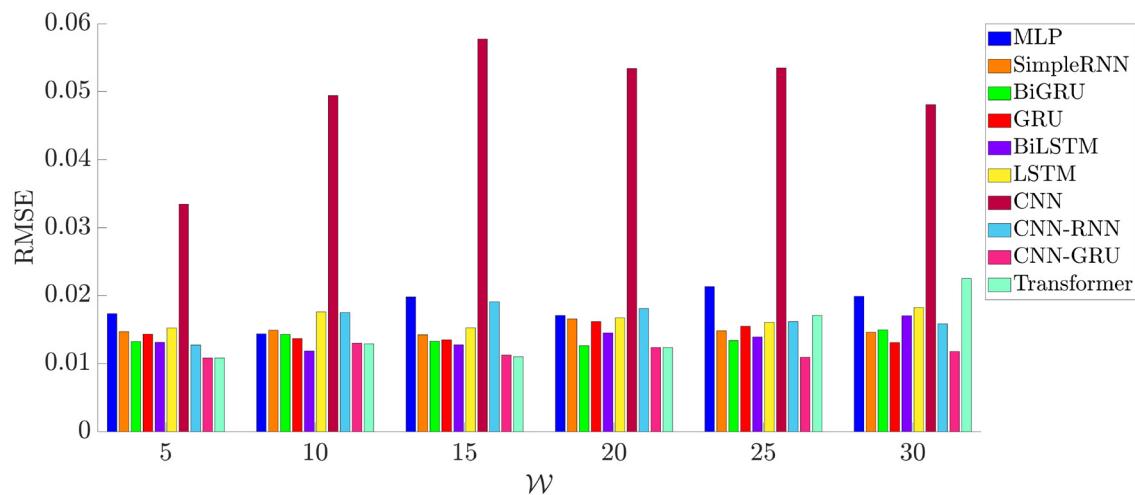


Fig. 13. Impact, in terms of RMSE, of the value of the time lag W on the XRP price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

and 30, respectively. More exactly, Transformer returns the lowest MAE rate in all of the long-term predictions for ETH and BTC, as well as trials with XRP ($s \in \{10, 20\}$). The evaluations in respect to the MAPE are shown in Figures 23, 27, and 31, respectively, where the superiority of Transformer is clear. Finally, R^2 is the last evaluated metric that illustrated for BTC, ETH and XRP in Figures 24, 28, and 32, respectively. On the basis of the collected data, similar to the previous results, Transformer outperforms the other models in terms of R^2 .

Finally, for the sake of clarity, real prices and those predicted through the considered DL algorithms are shown for BTC, ETH, and XRP in Figures 33, 34, and 35, respectively.

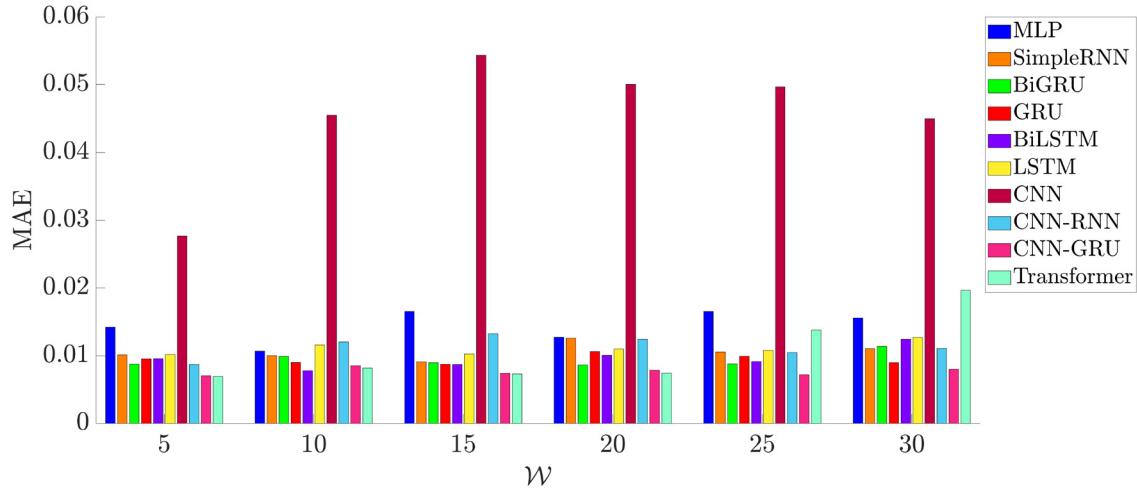


Fig. 14. Impact, in terms of MAE, of the value of the time lag \mathcal{W} on the XRP price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

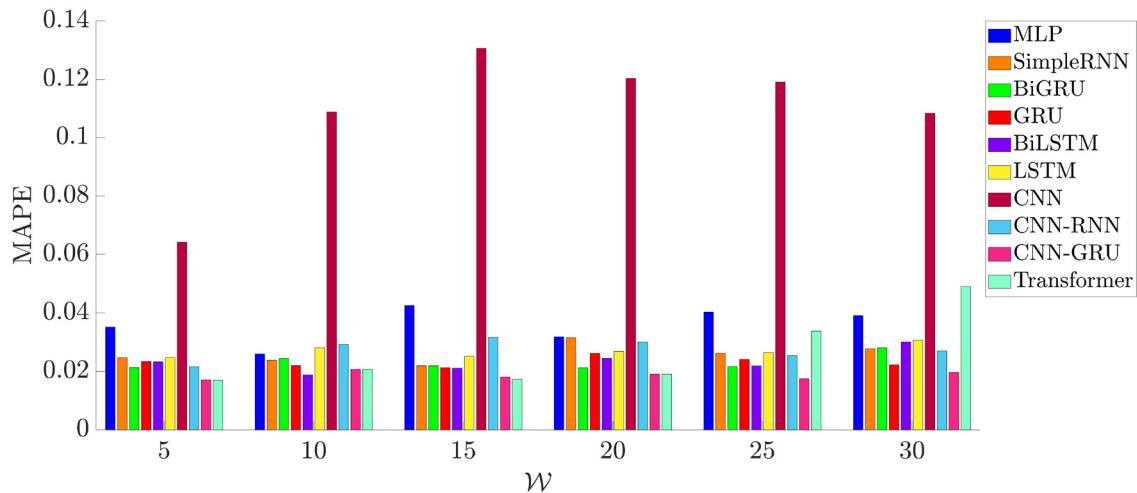


Fig. 15. Impact, in terms of MAPE, of the value of the time lag \mathcal{W} on the XRP price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

Table 5. Best-Performing DL Algorithms, According to RMSE, for the Cryptocurrencies' Prices Prediction, Considering a 3-Step Ahead Prediction Horizon ($s = 3$) and Various Values of \mathcal{W}

Cryptocurrency	$\mathcal{W} = 5$	$\mathcal{W} = 10$	$\mathcal{W} = 15$	$\mathcal{W} = 20$	$\mathcal{W} = 25$	$\mathcal{W} = 30$
BTC	Transformer	Transformer	Transformer	Transformer	Transformer	GRU
ETH	Transformer	BiLSTM	Transformer	Transformer	Transformer	Transformer
XRP	Transformer	BiLSTM	Transformer	Transformer	CNN-GRU	CNN-GRU

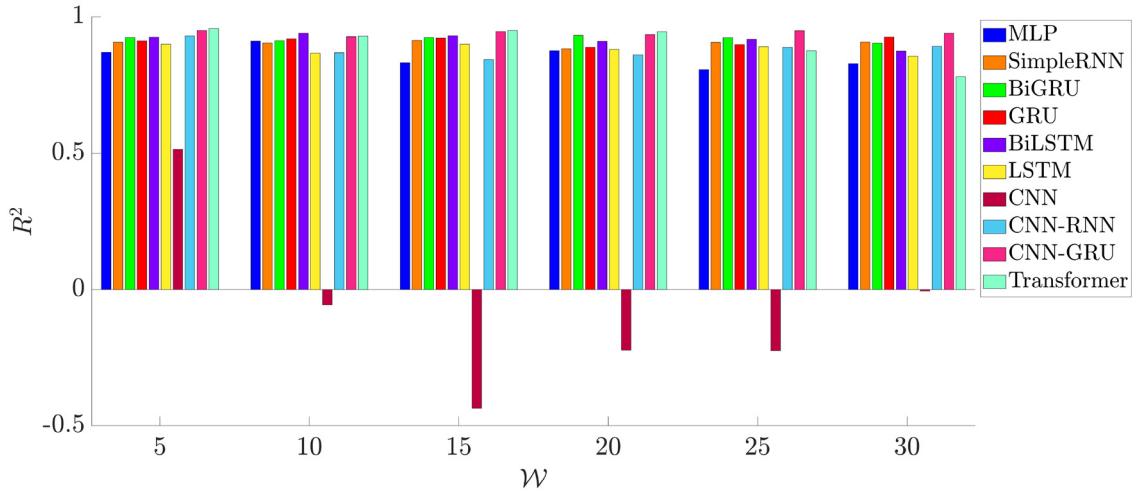


Fig. 16. Impact, in terms of R^2 , of the value of the time lag \mathcal{W} on the XRP price prediction with various DL algorithms, considering a 3-step ahead prediction horizon ($s = 3$).

5.2 Computational Complexity

In order to investigate the computational complexity of the considered DL algorithms, the number of **Multiply and ACCumulate (MAC)** operations, the number of parameters, and the FLASH memory size returned by each DL model, have been considered as performance metrics exploiting the ST Edge AI Developer Cloud [49] tool. As a result, the obtained performance is shown in Table 7, where it can be observed that CNN-RNN returns the lowest number of MAC operations and parameters in comparison to the other discussed RNNs, while SimpleRNN offers the smallest FLASH memory size.

5.3 Joint Accuracy-Complexity Tradeoff

Finally, in order to evaluate the overall computational-accuracy performance tradeoff of the considered DL algorithms in predicting BTC, ETH, and XRP prices, it is crucial to assess a performance metric that considers both the computing complexity and the accuracy. To this end, we define the following metric:

$$\xi \triangleq \text{MACs Number} * \text{Average MAPE}. \quad (25)$$

Then, the average MAPE has been calculated considering various time lags $\mathcal{W} \in \{5, 10, 15, 20, 25, 30\}$ and a prediction horizon $s = 3$ for the different DL models of interest. According to the results shown in Table 8, CNN-GRU provides the lowest value of ξ when focusing on the BTC and ETH price prediction, while CNN-RNN returns the lowest value of ξ for XPR price prediction. These results indicate that the time lag substantially impacts the prediction accuracy: in fact, shorter values of the time lag capture immediate effects and short-term patterns, whereas longer time lags capture stable trends and cyclic behaviors. Moreover, the selected time lag sizes were customized to differentiate the volatility and momentum patterns of various cryptocurrencies, thus improving the models' capacity to derive insights from past data. Then, by fine-tuning these time lags, the prediction precision was enhanced, providing traders and investors with more reliable forecasts to make well-informed decisions in practical situations.

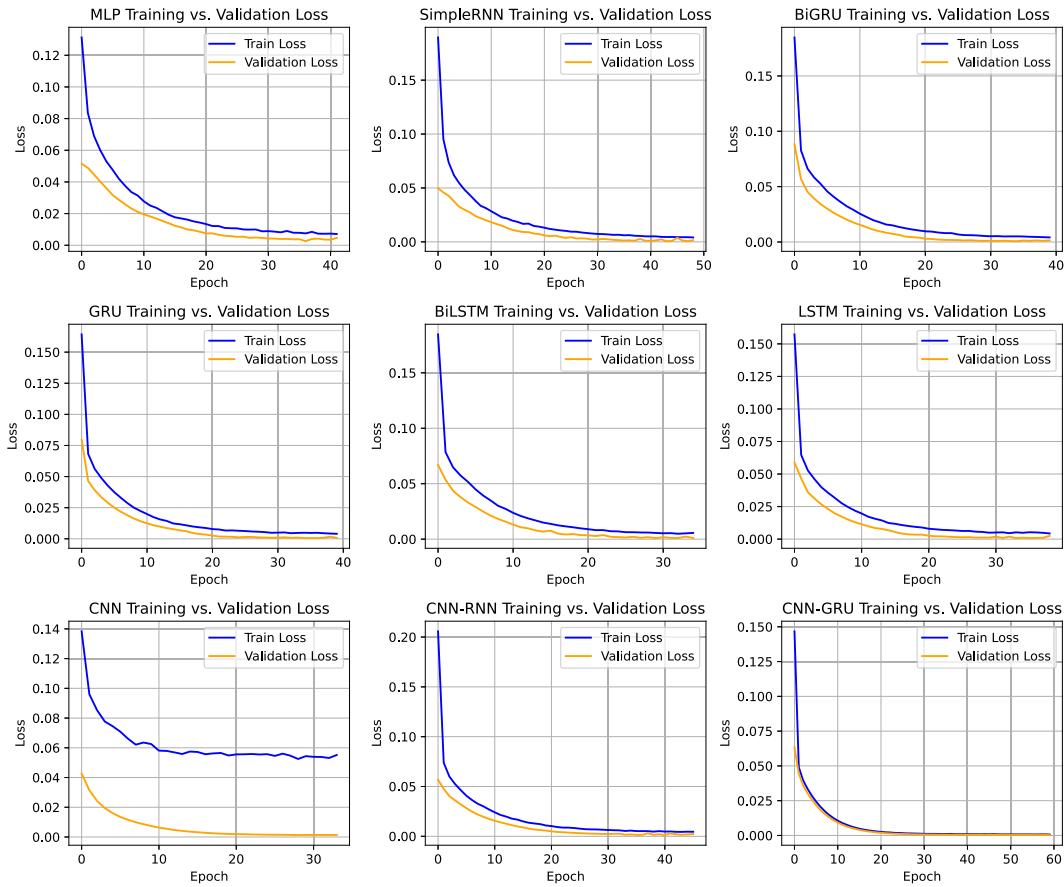


Fig. 17. Training and validation loss for BTC using the proposed DL models.

5.4 Research Limitations

Finally, for the sake of completeness, in the following a few potential limitations of the proposed approach (which can thus lead to further improvements and research directions) are highlighted and discussed.

- *Determining the appropriate time lag*: identifying the most suitable time lag for prediction requires substantial experimentation. In fact, the selection process is empirical and not straightforward, involving multiple comparisons in order to determine the appropriate value for accurate forecasts and understand the relationship between prediction accuracy and varying lag sizes.
- *Model generalization across cryptocurrencies*: developing a model able to predict values for all the existing cryptocurrencies with the lowest error is challenging, given the fact that each cryptocurrency has unique features and market behaviors. This makes the (ideal) goal of constructing a generally applicable forecasting model exceedingly difficult.
- *Limited historical data*: the availability of short historical data for new cryptocurrencies can complicate the creation of credible forecasting models, since accurate models generally depend on significant historical data in order to be able to capture trends and patterns properly.

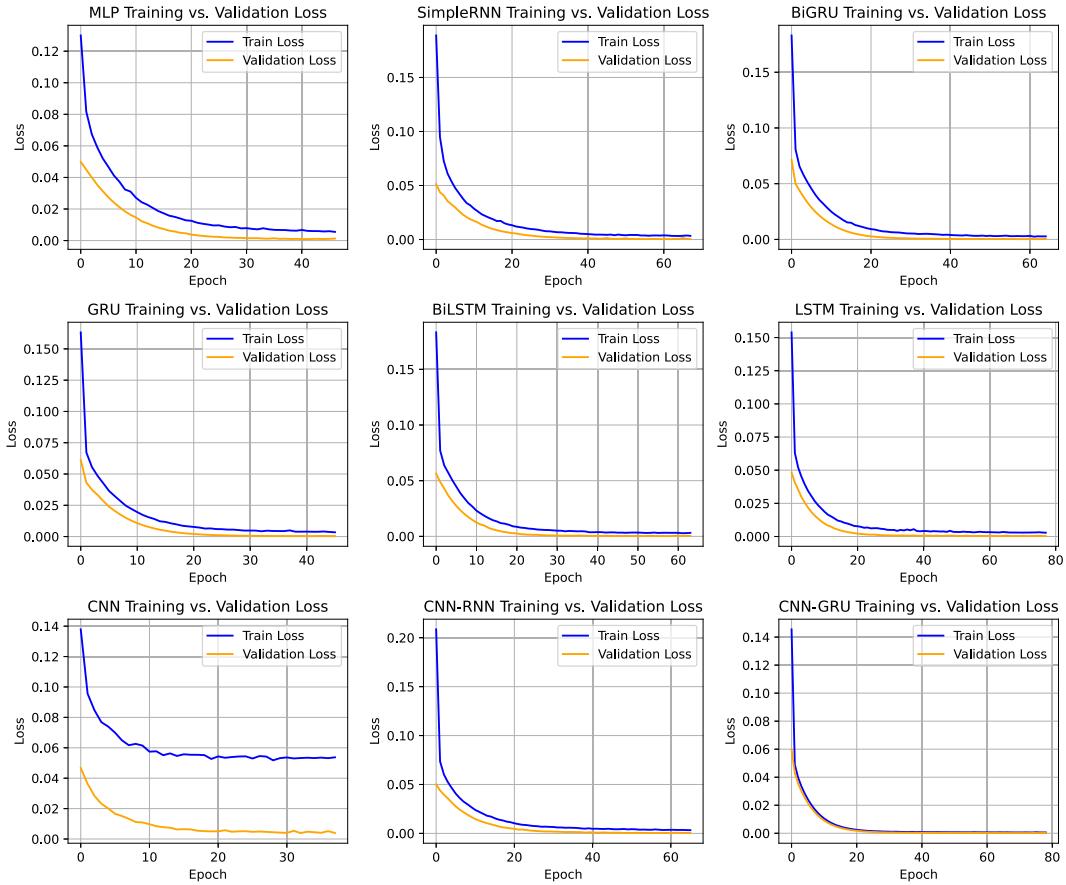


Fig. 18. Training and validation loss for ETH using the proposed DL models.

- Market volatility and unpredictability*: the tremendous volatility and unpredictability of the cryptocurrency market can severely impair prediction accuracy. This is due to sudden market movements, regulatory developments, and other external factors, that can lead to considerable failures in predictions.
- Data quality*: the precision of predictions is contingent on the data quality. Therefore, it is crucial to ensure that the data collection process is robust and efficient.
- Overfitting*: DL models are susceptible to overfitting, which makes it essential to carefully tune the model and apply appropriate regularization techniques.

6 Conclusions

In this work, we have presented and discussed an experimental comparative analysis of different DL algorithms (namely MLP, SimpleRNN, LSTM, BiLSTM, GRU, BiGRU, CNN, CNN-RNN, CNN-GRU, Transformer) for short- and long-term price prediction of three well-known and widely used cryptocurrencies (namely BTC, ETH, and XRP). The obtained results highlight that, for both *short-term* and *long-term* predictions, the Transformer model outperforms the other considered DL algorithms in all the cases, returning the lowest prediction errors (including RMSE, MAE, and MAPE) and achieving the highest R^2 values across various experiments. Furthermore, on the basis of experimental findings, CNN-RNN exhibits the lowest complexity level, measured in terms of the

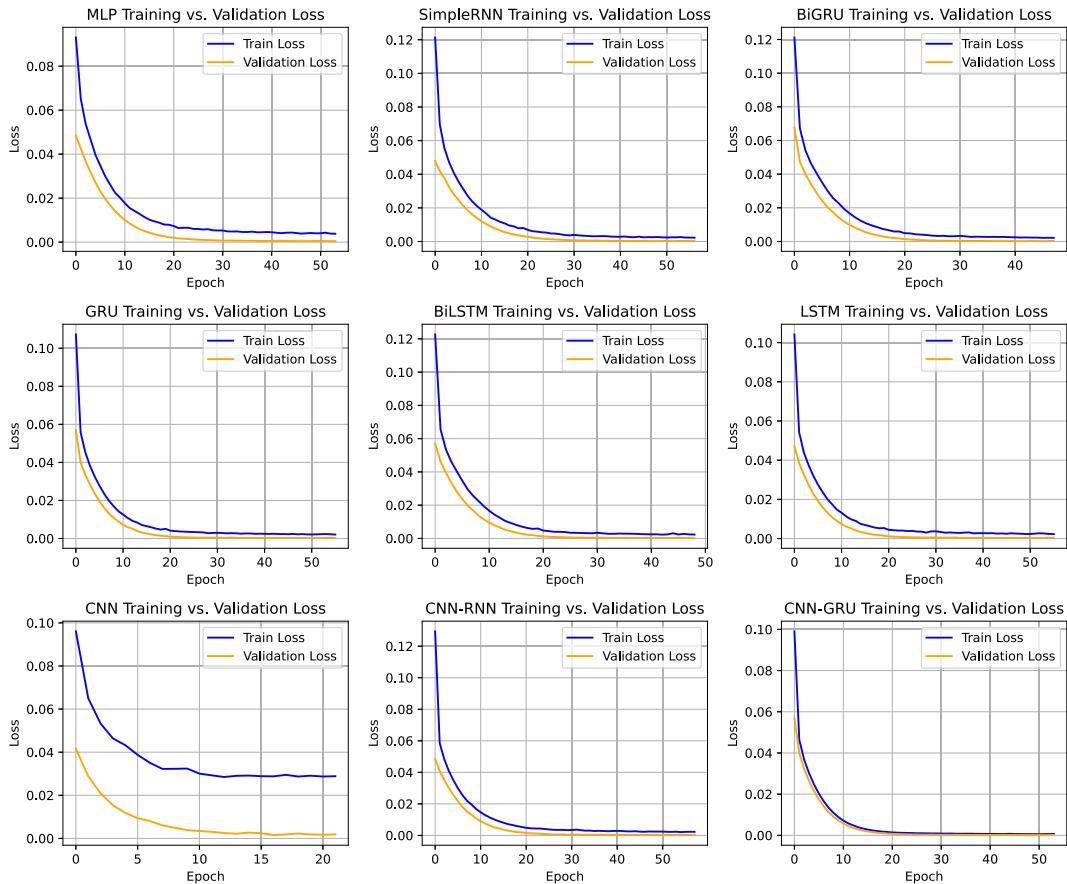


Fig. 19. Training and validation loss for XRP using the proposed DL models.

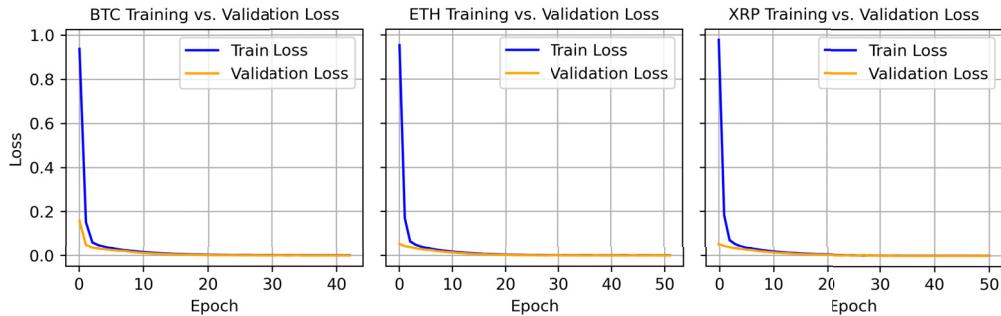


Fig. 20. Training and validation loss for BTC, ETH, and XRP using the proposed Transformer model.

number of MAC operations. On the other hand, SimpleRNN has the lowest number of parameters and requires the smallest amount of FLASH memory (compared to the other DL models). Finally, by investigating the joint accuracy-complexity through the metric ξ introduced in Equation (25), our results show that CNN-GRU provides the lowest (i.e., the best) value of ξ when predicting future BTC and ETH prices, while CNN-RNN yields the best

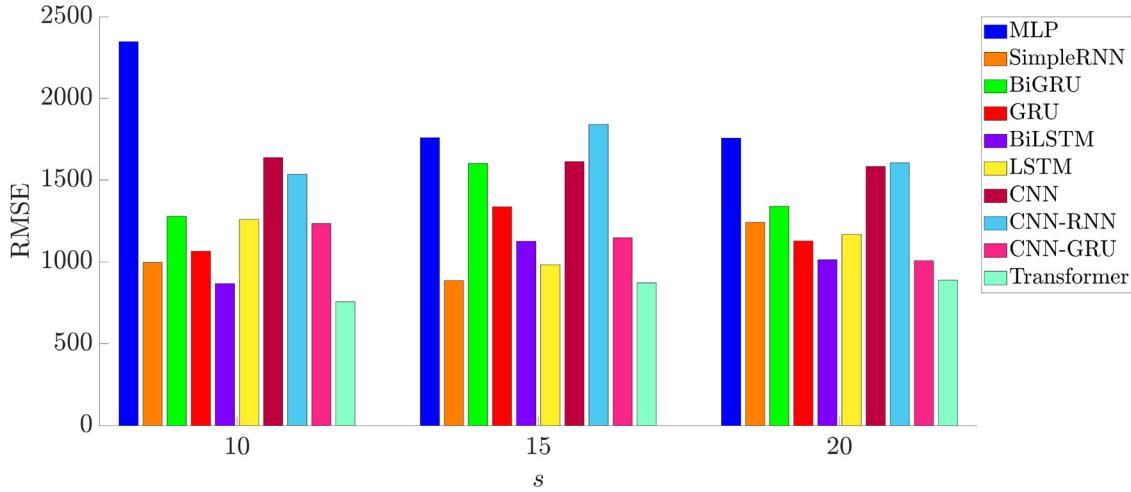


Fig. 21. Impact, in terms of RMSE, of the value of the prediction horizon s on the BTC price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

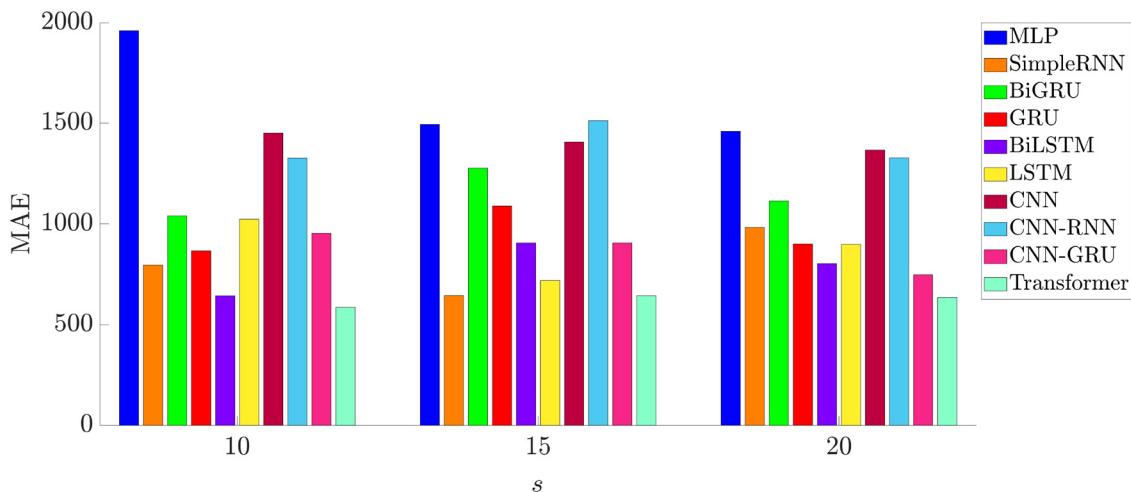


Fig. 22. Impact, in terms of MAE, of the value of the prediction horizon s on the BTC price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

performance (i.e., the lowest value of ξ) when the prediction focuses on XRP. In fact, although the Transformer model returns the highest prediction accuracy, it is also the most computationally expensive and is not the best considering the metric ξ . The complexity of the Transformer may be a limitation depending on the available resources on the chosen processing platform. Future research could focus on integrating sentiment analysis-based approaches and the influence of macroeconomic factors into predictive models to enhance the accuracy of cryptocurrency price predictions.

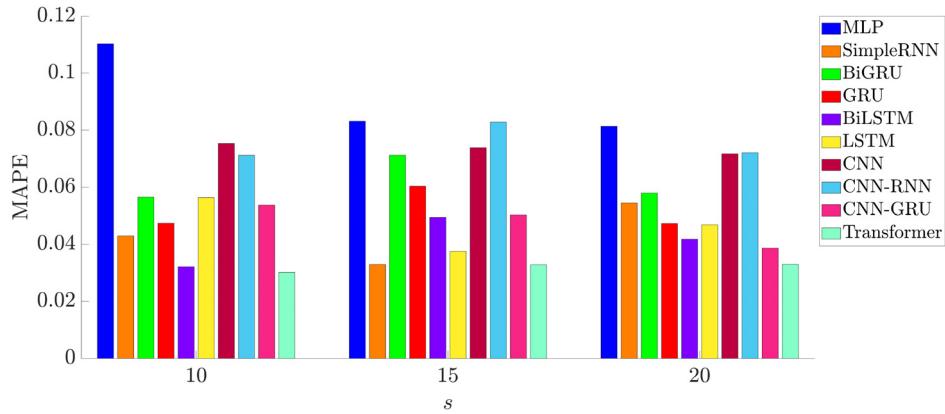


Fig. 23. Impact, in terms of MAPE, of the value of the prediction horizon s on the BTC price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

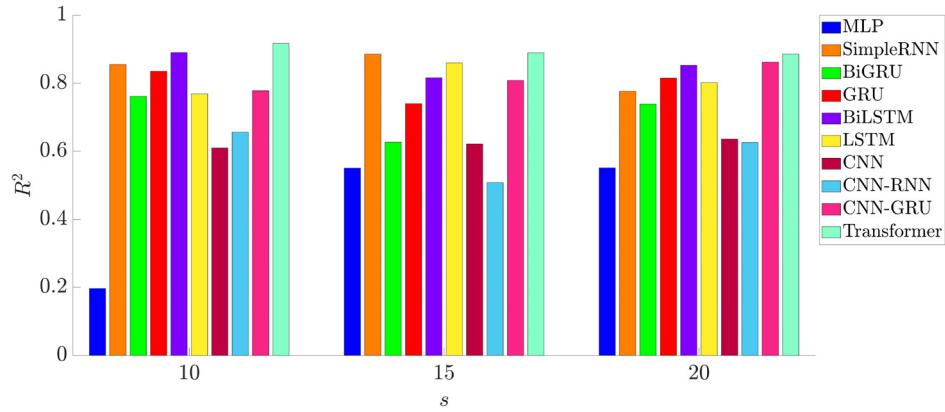


Fig. 24. Impact, in terms of R^2 , of the value of the prediction horizon s on the BTC price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

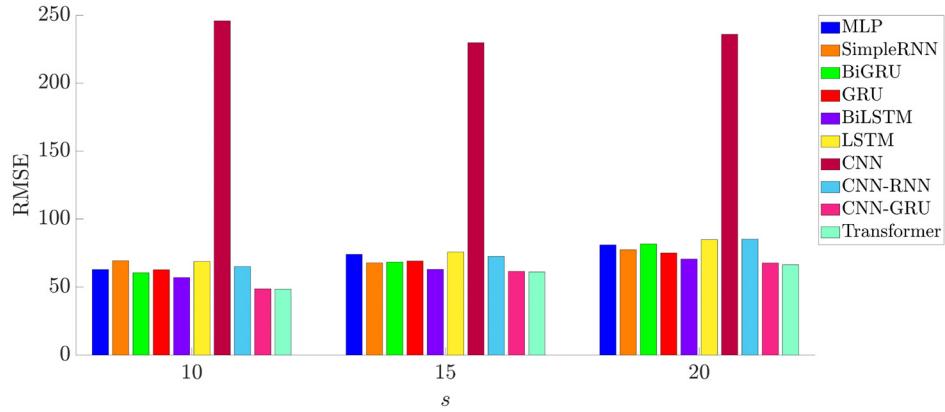


Fig. 25. Impact, in terms of RMSE, of the value of the prediction horizon s on the ETH price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

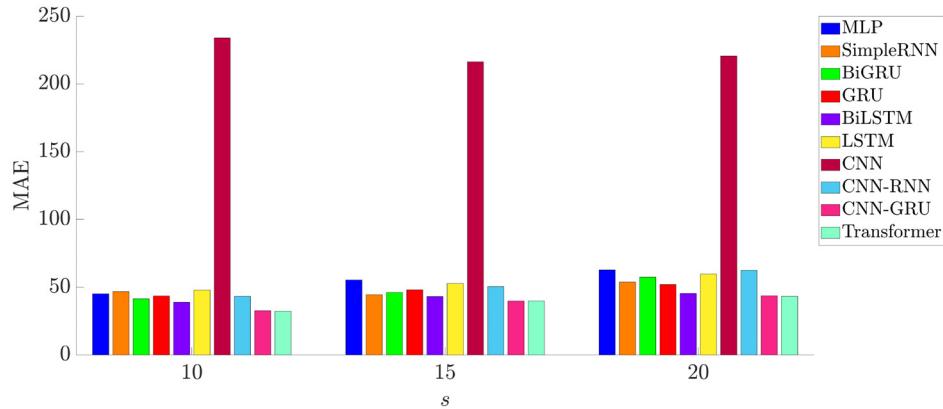


Fig. 26. Impact, in terms of MAE, of the value of the prediction horizon s on the ETH price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

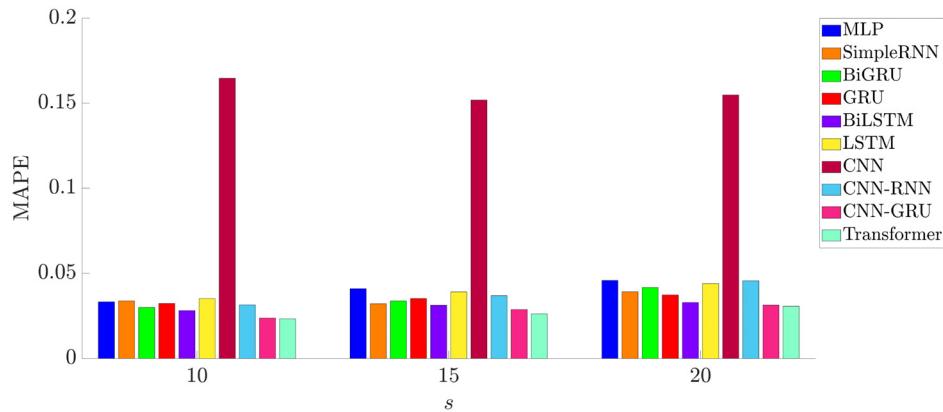


Fig. 27. Impact, in terms of MAPE, of the value of the prediction horizon s on the ETH price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

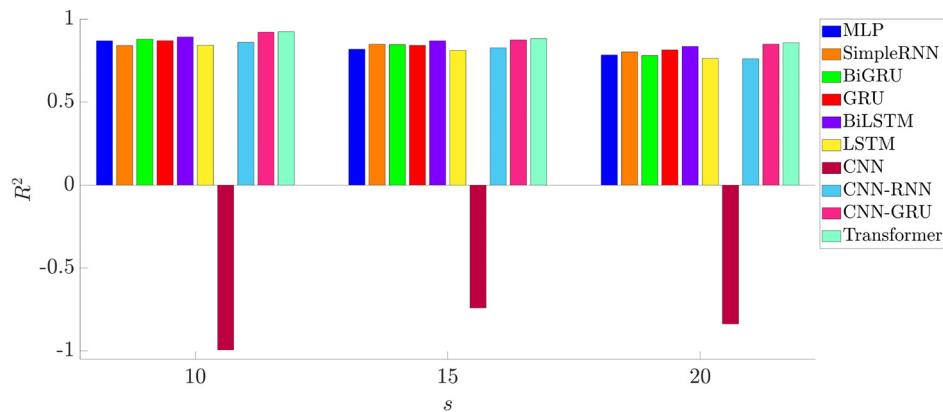


Fig. 28. Impact, in terms of R^2 , of the value of the prediction horizon s on the ETH price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

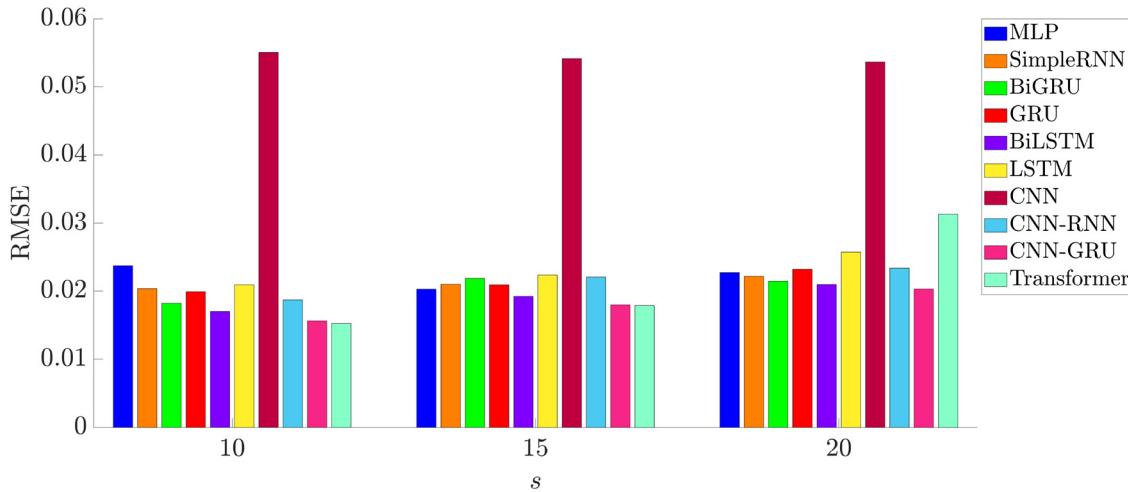


Fig. 29. Impact, in terms of RMSE, of the value of the prediction horizon s on the XRP price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

Table 6. Best-Performing DL Algorithms, According to RMSE, for the Cryptocurrencies' Prices Prediction, Considering Various Long-Term Step Ahead Prediction Horizons $s \in \{10, 15, 20\}$ and a Fixed Time Lag $\mathcal{W} = 25$

Cryptocurrency	$s = 10$	$s = 15$	$s = 20$
BTC	Transformer	Transformer	Transformer
ETH	Transformer	Transformer	Transformer
XRP	Transformer	Transformer	CNN-GRU

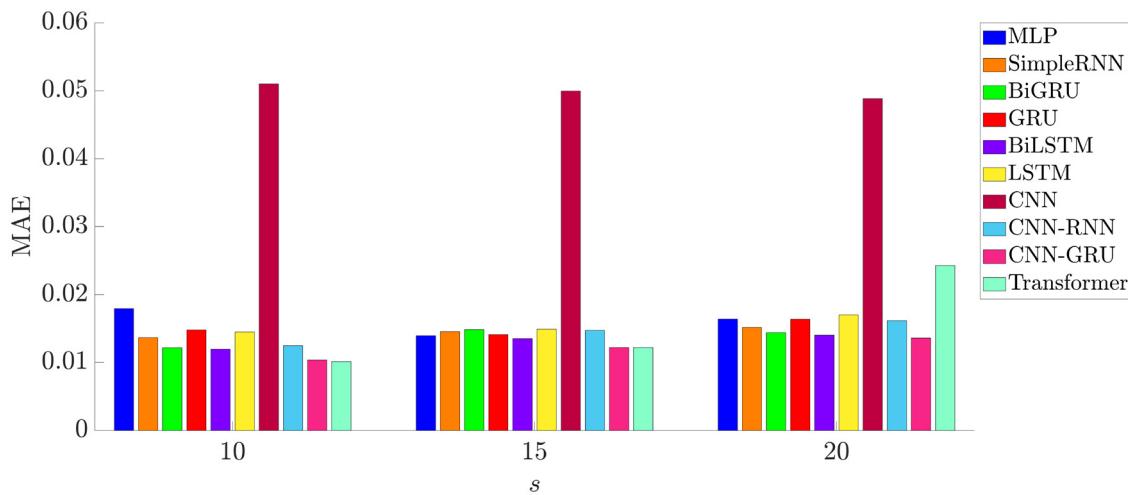


Fig. 30. Impact, in terms of MAE, of the value of the prediction horizon s on the XRP price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

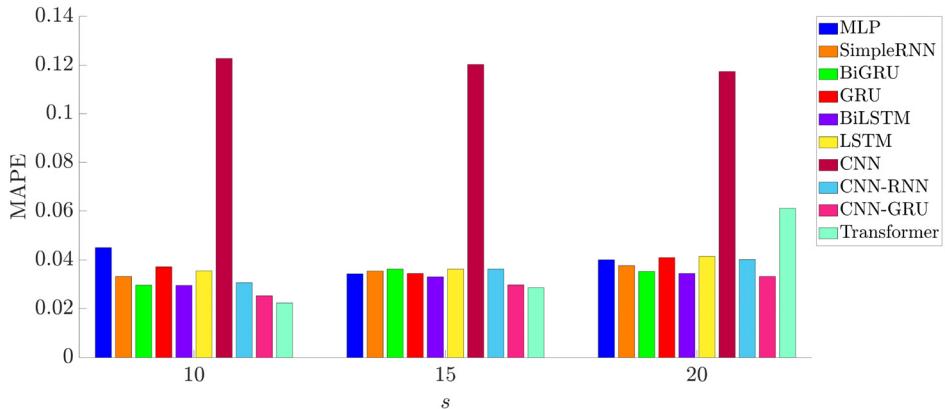


Fig. 31. Impact, in terms of MAPE, of the value of the prediction horizon s on the XRP price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

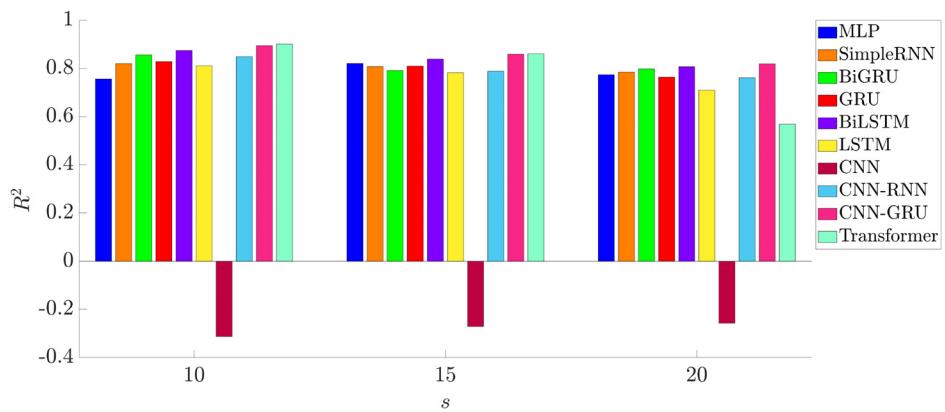


Fig. 32. Impact, in terms of R^2 , of the value of the prediction horizon s on the XRP price prediction with various DL algorithms, considering a time lag $\mathcal{W} = 25$.

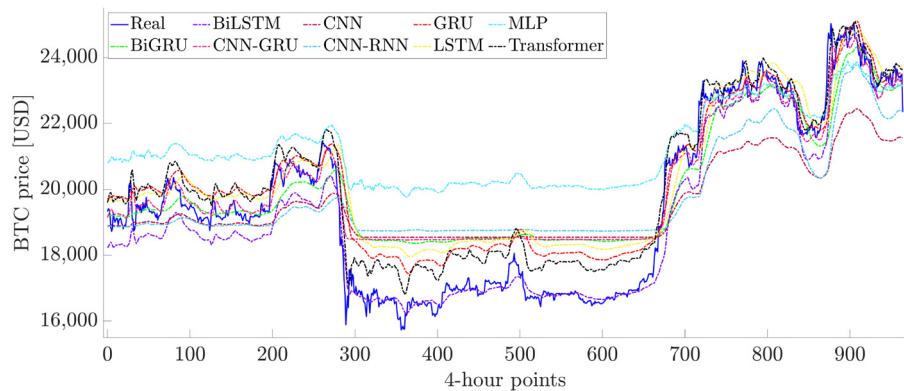


Fig. 33. BTC price long-term prediction results obtained by the considered DL models, with a time lag $\mathcal{W} = 25$, and considering a 10-step ahead prediction horizon ($s = 10$).

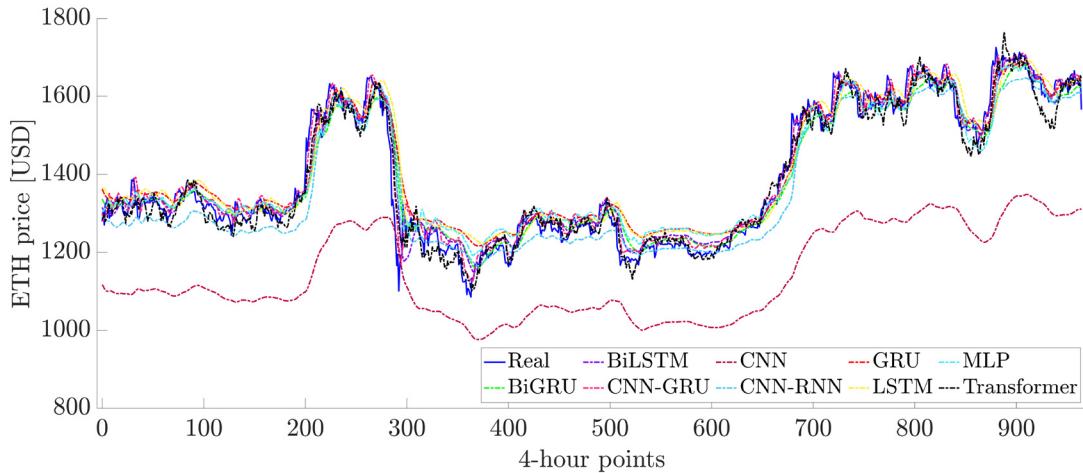


Fig. 34. ETH price long-term prediction results obtained by the considered DL models, with a time lag $\mathcal{W} = 25$, and considering a 10-step ahead prediction horizon ($s = 10$).

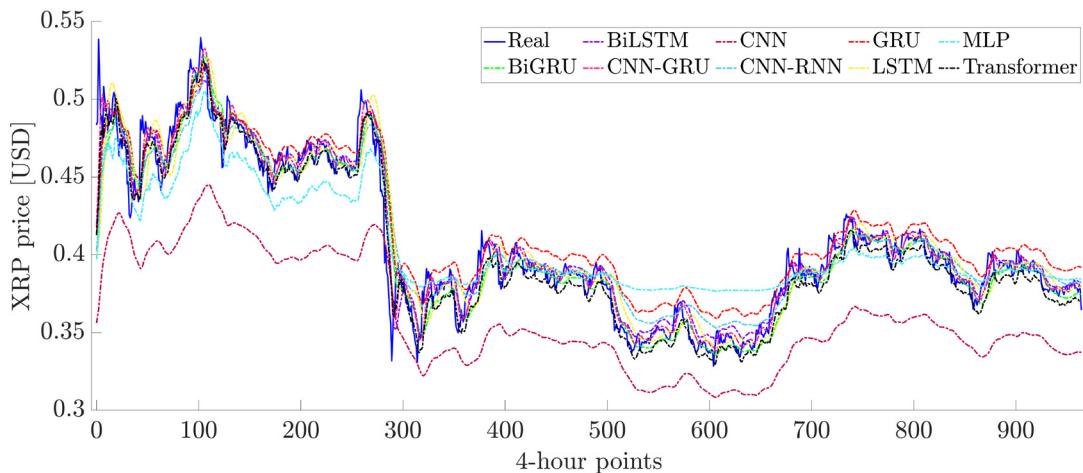


Fig. 35. XRP price long-term prediction results obtained by the considered DL models, with a time lag $\mathcal{W} = 25$, and considering a 10-step ahead prediction horizon ($s = 10$).

Table 7. Complexity Comparison of the Considered DL Models

	SimpleRNN	LSTM	BiLSTM	GRU	BiGRU	CNN-RNN	CNN-GRU	Transformer
MACs number	155,099	622,599	432,599	462,599	320,099	83,879	129,287	1,151,360
Parameters number	6,263	24,863	17,263	18,963	13,263	6,883	11,139	299,907
FLASH size [KiB]	37	116	89	93	74	46	69	970

Bold results representing the lowest values reached for each performance metric.

Table 8. Joint Accuracy-Complexity ξ of the Considered DL Models

	SimpleRNN	LSTM	BiLSTM	GRU	BiGRU	CNN-RNN	CNN-GRU	Transformer
BTC	4,898.29	18,371.39	17,027.93	14,509.03	14,309.22	7,608.38	2,945.79	23,976.89
ETH	3,761.29	16,428.58	8,695.51	10,435.69	7,323.34	2,248.59	2,235.88	21,324.44
XRP	4,032.3	16,803.34	10,063.94	10,724.07	7,399.53	2,305.26	2,412.08	30,091.47

Bold values representing the lowest values reached for each performance metric.

References

- [1] Sabeen Ahmed, Ian E. Nielsen, Aakash Tripathi, Shamoona Siddiqui, Ravi P. Ramachandran, and Ghulam Rasool. 2023. Transformers in Time-Series Analysis: A Tutorial. *Circuits, Systems, and Signal Processing* 42, 12 (2023), 7433–7466. DOI: <https://doi.org/10.1007/s00034-023-02454-8>
- [2] Talha Burak Alakus and Ibrahim Turkoglu. 2020. Comparison of Deep Learning Approaches to Predict COVID-19 Infection. *Chaos, Solitons & Fractals* 140 (2020), 110120. DOI: <https://doi.org/10.1016/j.chaos.2020.110120>
- [3] Rahmat Albariqi and Edi Winarko. 2020. Prediction of Bitcoin Price Change using Neural Networks. In *Proceedings of the International Conference on Smart Technology and Applications (ICoSTA)*. IEEE, 1–4. DOI: <https://doi.org/10.1109/ICoSTA48221.2020.1570610936>
- [4] Mohammed Alshehri. 2023. Blockchain-Assisted Internet of Things Framework in Smart Livestock Farming. *Internet of Things* 22 (2023), 100739. DOI: <https://doi.org/10.1016/j.iot.2023.100739>
- [5] Ibrahim M. Awad, Ghada K. Al-Jerashi, and Zaid Ahmad Alabadi. 2021. Determinants of Private Domestic Investment in Palestine: Time Series Analysis. *Journal of Business and Socio-Economic Development* 1, 1 (2021), 71–86. DOI: <https://doi.org/10.1108/JBSED-04-2021-0038>
- [6] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning Long-Term Dependencies with Gradient Descent Is Difficult. *IEEE Transactions on Neural Networks* 5, 2 (1994), 157–166. DOI: <https://doi.org/10.1109/72.279181>
- [7] Tania Cerquitelli, Michela Meo, Marilia Curado, Lea Skorin-Kapov, and Eirini Eleni Tsipropoulou. 2023. Machine Learning Empowered Computer Networks. *Computer Networks* 230 (2023), 109807. DOI: <https://doi.org/10.1016/j.comnet.2023.109807>
- [8] Davide Chicco, Matthijs J. Warrens, and Giuseppe Jurman. 2021. The Coefficient of Determination R-squared is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Computer Science* 7 (2021), e623. DOI: <https://doi.org/10.7717/peerj.cs.623>
- [9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, arXiv:1412.3555. Retrieved from <https://arxiv.org/abs/1412.3555>
- [10] Jenny Cifuentes, Geovanny Marulanda, Antonio Bello, and Javier Reneses. 2020. Air Temperature Forecasting Using Machine Learning Techniques: A Review. *Energies* 13, 16 (2020), 1–28. DOI: <https://doi.org/10.3390/en13164215>
- [11] Gaia Codeluppi, Antonio Cilfone, Luca Davoli, and Gianluigi Ferrari. 2020. AI at the Edge: A Smart Gateway for Greenhouse Air Temperature Forecasting. In *Proceedings of the IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*. IEEE, 348–353. DOI: <https://doi.org/10.1109/MetroAgriFor50201.2020.9277553>
- [12] Gaia Codeluppi, Luca Davoli, and Gianluigi Ferrari. 2021. Forecasting Air Temperature on Edge Devices with Embedded AI. *Sensors* 21, 12 (2021), 1–29. <https://doi.org/10.3390/s21123973>
- [13] CoinMarketCap. 2024. Global Cryptocurrency Charts – Total Market Cap. Retrieved August 29, 2024 from <https://coinmarketcap.com/charts/>
- [14] Joy Dip Das, Ruppa K. Thulasiram, Christopher Henry, and Aerambamoorthy Thavaneswaran. 2024. Encoder-Decoder Based LSTM and GRU Architectures for Stocks and Cryptocurrency Prediction. *Journal of Risk and Financial Management* 17, 5 (2024), 200. DOI: <https://doi.org/10.3390/jrfm17050200>
- [15] Georgios Fragkos, Cyrus Minwalla, Jim Plusquellic, and Eirini Eleni Tsipropoulou. 2020. Artificially Intelligent Electronic Money. *IEEE Consumer Electronics Magazine* 10, 4 (2020), 81–89. DOI: <https://doi.org/10.1109/MCE.2020.3024512>
- [16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. DOI: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [17] Parikshit Joshi, Vijaishri Tewari, Shailendra Kumar, and Anshu Singh. 2023. Blockchain Technology for Sustainable Development: A Systematic Literature Review. *Journal of Global Operations and Strategic Sourcing* 16, 3 (2023), 683–717. DOI: <https://doi.org/10.1108/JGOSS-06-2022-0054>
- [18] Zahra Karevan and Johan A. K. Suykens. 2020. Transductive LSTM for Time-Series Prediction: An Application to Weather Forecasting. *Neural Networks* 125 (2020), 1–9. DOI: <https://doi.org/10.1016/j.neunet.2019.12.030>
- [19] Seyed Mehran Kazemi, Rishabh Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. 2019. Time2Vec: Learning a Vector Representation of Time. arXiv:1907.05321. Retrieved from <https://doi.org/10.48550/arXiv.1907.05321>

- [20] Keras. 2024. Dropout Layer. Retrieved August 29, 2024 from https://keras.io/api/layers/regularization_layers/dropout/
- [21] Keras. 2024. EarlyStopping. Retrieved August 29, 2024 from https://keras.io/api/callbacks/early_stopping/
- [22] Keras. 2024. Layer Weight Regularizers. Retrieved August 29, 2024 from <https://keras.io/api/layers/regularizers/>
- [23] Mehdi Khashei and Mehdi Bijari. 2011. A Novel Hybridization of Artificial Neural Networks and ARIMA Models for Time Series Forecasting. *Applied Soft Computing* 11, 2 (2011), 2664–2675. DOI: <https://doi.org/10.1016/j.asoc.2010.10.015>
- [24] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from <https://doi.org/10.48550/arXiv.1412.6980>
- [25] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. arXiv:1706.02515. Retrieved from <https://doi.org/10.48550/arXiv.1706.02515>
- [26] Salim Lahmiri and Stelios Bekiros. 2019. Cryptocurrency Forecasting with Deep Learning Chaotic Neural Networks. *Chaos, Solitons & Fractals* 118 (2019), 35–40. DOI: <https://doi.org/10.1016/j.chaos.2018.11.014>
- [27] Navmeen Latif, Joseph Durai Selvam, Manohar Kapse, Vinod Sharma, and Vaishali Mahajan. 2023. Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices. *Australasian Accounting, Business and Finance Journal* 17, 1 (2023), 256–276. DOI: <https://doi.org/10.14453/aabfj.v17i1.15>
- [28] Tonghui Li. 2022. Prediction of Bitcoin Price Based on LSTM. In *Proceedings of the International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*. IEEE, 19–23. DOI: <https://doi.org/10.1109/MLISE57402.2022.00012>
- [29] Terence C. Mills. 1990. *Time Series Techniques for Economists*. Cambridge University Press, Cambridge, UK.
- [30] Douglas L. L. Moura, Andre L. L. Aquino, and Antonio A. F. Loureiro. 2024. An Edge Computing and Distributed Ledger Technology Architecture for Secure and Efficient Transportation. *Ad Hoc Networks* 164 (2024), 103633. DOI: <https://doi.org/10.1016/j.adhoc.2024.103633>
- [31] Tashreef Muhammad, Anika Bintee Aftab, Md. Mainul Ahsan, Maishameem Meherin Muhu, Muhammad Ibrahim, Shahidul Islam Khan, and Mohammad Shafiu1 Alam. 2023. Transformer-based Deep Learning Model for Stock Price Prediction: A Case Study on Bangladesh Stock Market. *International Journal of Computational Intelligence and Applications* 22, 3 (2023), 2350013. DOI: <https://doi.org/10.1142/S146902682350013X>
- [32] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved August 29, 2024 from <https://bitcoin.org/bitcoin.pdf>
- [33] Anum Nawaz, Laura Belli, Luca Davoli, and Gianluigi Ferrari. 2024. Hyperledger Fabric in Precision Agriculture: A Study on Data Integrity and Availability. In *Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE, 1–8. DOI: <https://doi.org/10.1109/CITS61189.2024.10608019>
- [34] P. Nithyakani, Rijo Jackson Tom, Piyush Gupta, A Shanthini, Vivia Mary John, and Vipul Sharma. 2021. Prediction of Bitcoin Price Using Bi-LSTM Network. In *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 1–5. DOI: <https://doi.org/10.1109/ICCCI50826.2021.9402427>
- [35] D. Olvera-Juarez and E. Huerta-Manzanilla. 2019. Forecasting Bitcoin Pricing with Hybrid models: A Review of the Literature. *International Journal of Advanced Engineering Research and Science* 6, 9 (2019), 161–164. DOI: <https://doi.org/10.22161/ijaers.69.18>
- [36] Maria Rita Palattella, Stefano Scanzio, and Sinem Coleri Ergen. 2019. Ad-Hoc, Mobile, and Wireless Networks. In *Proceedings of the 18th International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW '19)*. DOI: <https://doi.org/10.1007/978-3-030-31831-4>
- [37] Gyana Ranjan Patra and Mihir Narayan Mohanty. 2023. Price Prediction of Cryptocurrency using a Multi-Layer Gated Recurrent Unit Network with Multi Features. *Computational Economics* 62, 4 (2023), 1525–1544. DOI: <https://doi.org/10.1007/s10614-022-10310-1>
- [38] Obryan Poyer. 2017. Exploring the determinants of Bitcoin's price: An application of Bayesian structural time series. arXiv:1706.01437. Retrieved from <https://doi.org/10.48550/arXiv.1706.01437>
- [39] Arief Radityo, Qorib Munajat, and Indra Budi. 2018. Prediction of Bitcoin Exchange Rate to American Dollar Using Artificial Neural Network Methods. In *Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 433–438. DOI: <https://doi.org/10.1109/ICACSIS.2017.8355070>
- [40] Rajat Rathore, Deepti Mishra, Pawan Singh Mehra, Om Pal, AHMAD SOBRI HASHIM, Azrulhizam Shapi'i, T. Ciano, and Meshal Shutaywi. 2022. Real-World Model for Bitcoin Price Prediction. *Information Processing & Management* 59, 4 (2022), 102968. DOI: <https://doi.org/10.1016/j.ipm.2022.102968>
- [41] Alessandra Rizzardi, Sabrina Sicari, Jesus F. Cevallos M., Alberto Coen-Porisini. 2024. IoT-Driven Blockchain to Manage the Healthcare Supply Chain and Protect Medical Records. *Future Generation Computer Systems* 161 (2024), 415–431. DOI: <https://doi.org/10.1016/j.future.2024.07.039>
- [42] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. arXiv:1609.04747. Retrieved from <https://doi.org/10.48550/arXiv.1609.04747>
- [43] scikit-learn developers. 2024. MinMaxScaler. Retrieved August 29, 2024 from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>
- [44] Phumudzo Lloyd Seabe, Claude Rodrigue Bambe Moutsinga, and Edson Pindza. 2023. Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. *Fractal and Fractional* 7, 2 (2023), 203. DOI: <https://doi.org/10.3390/fractfrac7020203>

- [45] Pratima Sharma, Suyel Namasudra, Ruben Gonzalez Crespo, Javier Parra-Fuente, and Munesh Chandra Trivedi. 2023. EHDHE: Enhancing Security of Healthcare Documents in IoT-enabled Digital Healthcare Ecosystems using Blockchain. *Information Sciences* 629 (2023), 703–718. DOI: <https://doi.org/10.1016/j.ins.2023.01.148>
- [46] Sheetal Singh, Nikhil Kumar Rajput, Vipin Kumar Rathi, Hari Mohan Pandey, Amit Kumar Jaiswal, and Prayag Tiwari. 2020. Securing Blockchain Transactions Using Quantum Teleportation and Quantum Digital Signature. *Neural Processing Letters* 55, 4 (2020), 3827–3842. DOI: <https://doi.org/10.1007/s11063-020-10272-1>
- [47] Sushant Singh and Ausif Mahmood. 2021. The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures. *IEEE Access* 9 (2021), 68675–68702. DOI: <https://doi.org/10.1109/ACCESS.2021.3077350>
- [48] Tanishq Soni, Deepali Gupta, Monica Dutta, and Gifty Gupta. 2023. Tracing Food Products in Supply Chain Using DLT Enabled Blockchain Technology. In *Proceedings of the International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM)*. IEEE, 1–6. DOI: <https://doi.org/10.1109/ELEXCOM58812.2023.10370526>
- [49] STMicroelectronics. 2024. ST Edge AI Developer Cloud. Retrieved August 29, 2024 from <https://stm32ai-cs.st.com/home>
- [50] Twelve Data Pte. Ltd. 2024. Twelvedata Platform. Retrieved August 29, 2024 from <https://www.twelvedata.com>
- [51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. arXiv:1706.03762. Retrieved from <https://doi.org/10.48550/arXiv.1706.03762>
- [52] Chaojie Wang, Yuanyuan Chen, Shuqi Zhang, and Qiuhibi Zhang. 2022. Stock Market Index Prediction using Deep Transformer Model. *Expert Systems with Applications* 208 (2022), 118128. DOI: <https://doi.org/10.1016/j.eswa.2022.118128>
- [53] Caosen Xu, Jingyuan Li, Bing Feng, and Baoli Lu. 2023. A Financial Time-Series Prediction Model based on Multiplex Attention and Linear Transformer Structure. *Applied Sciences* 13, 8 (2023), 5175. DOI: <https://doi.org/10.3390/app13085175>
- [54] Hongling Xu, Ruizhe Ma, Li Yan, and Zongmin Ma. 2021. Two-Stage Prediction of Machinery Fault Trend Based on Deep Learning for Time Series Analysis. *Digital Signal Processing* 117 (2021), 103150. DOI: <https://doi.org/10.1016/j.dsp.2021.103150>
- [55] Hongju Yan and Hongbing Ouyang. 2018. Financial Time Series Prediction Based on Deep Learning. *Wireless Personal Communications* 102, 2 (2018), 683–700. DOI: <https://doi.org/10.1007/s11277-017-5086-2>
- [56] Yijun Zou, Ting Meng, Peng Zhang, Wenzhen Zhang, and Huiyang Li. 2020. Focus on Blockchain: A Comprehensive Survey on Academic and Application. *IEEE Access* 8 (2020), 187182–187201. DOI: <https://doi.org/10.1109/ACCESS.2020.3030491>

Appendix

A Experimental Results with CNN-LSTM, CNN-BiLSTM, and CSS-BiGRU Models

For the sake of completeness and in order to provide the reader with all the experimental performance results obtained considering BTC, ETH, and XRP as target cryptocurrencies, the following additional experimental results, obtained with CNN-LSTM, CNN-BiLSTM, and CSS-BiGRU, are shown. In particular, since these algorithms do not outperform those detailed and discussed in Section 5.1 (relative to nine DL models: MLP, SimpleRNN, LSTM, BiLSTM, GRU, BiGRU, CNN, CNN-RNN, and CNN-GRU), it has been preferred to present the performance of these additional algorithms in the appendix, to provide the reader with a complete overview, yet without overloading him/her with too many results in the main body of the manuscript.

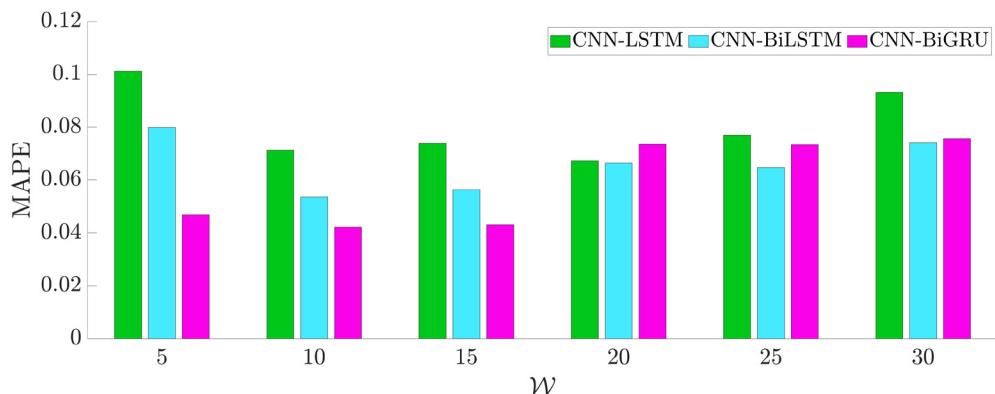


Fig. A1. Impact, in terms of MAPE, of the value of the time lag \mathcal{W} on the BTC price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

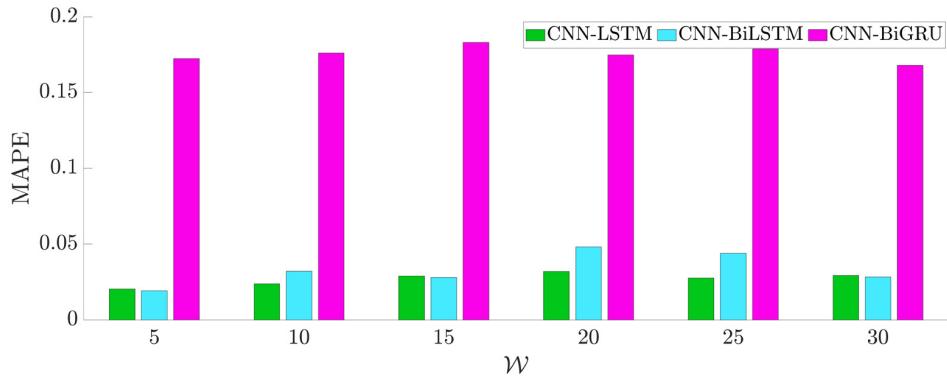


Fig. A2. Impact, in terms of MAPE, of the value of the time lag \mathcal{W} on the ETH price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

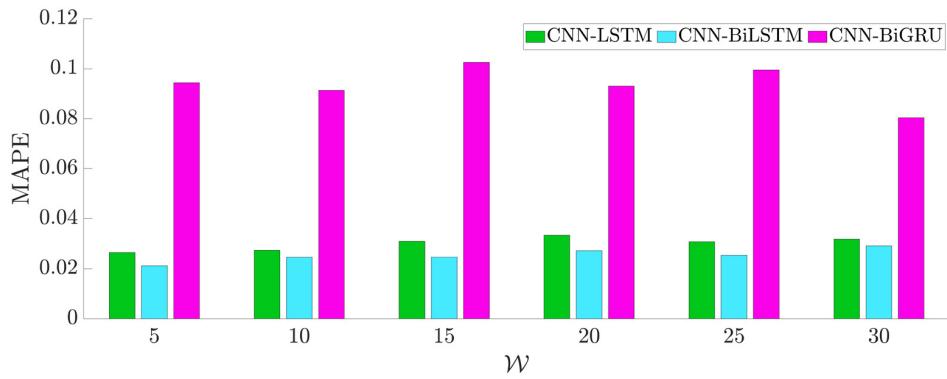


Fig. A3. Impact, in terms of MAPE, of the value of the time lag \mathcal{W} on the XRP price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

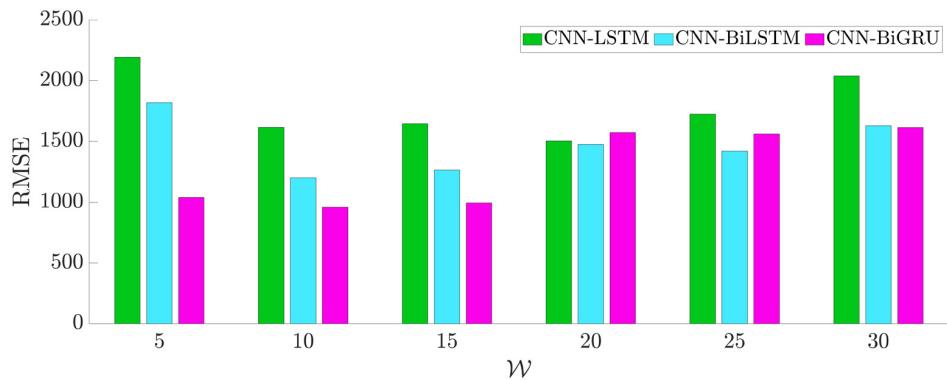


Fig. A4. Impact, in terms of RMSE, of the value of the time lag \mathcal{W} on the BTC price prediction with with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

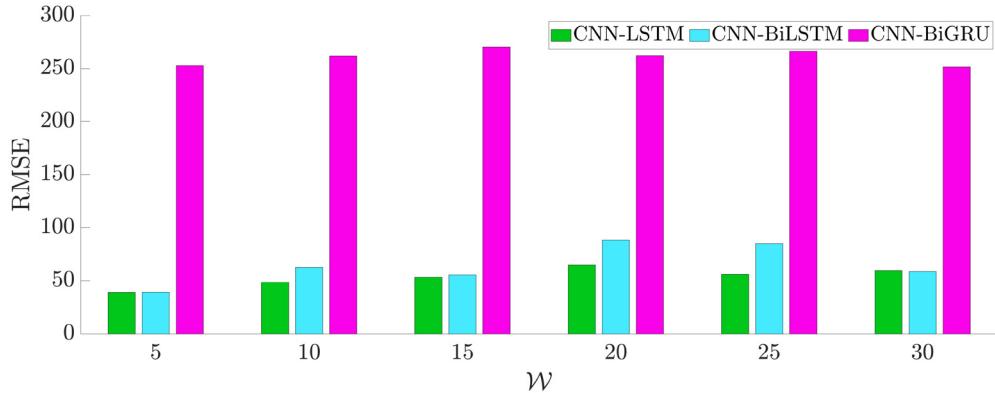


Fig. A5. Impact, in terms of RMSE, of the value of the time lag \mathcal{W} on the ETH price prediction with with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

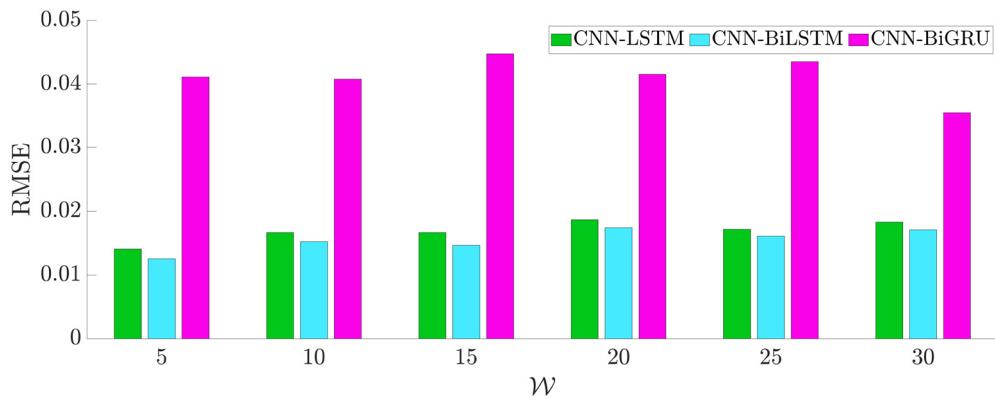


Fig. A6. Impact, in terms of RMSE, of the value of the time lag \mathcal{W} on the XRP price prediction with with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

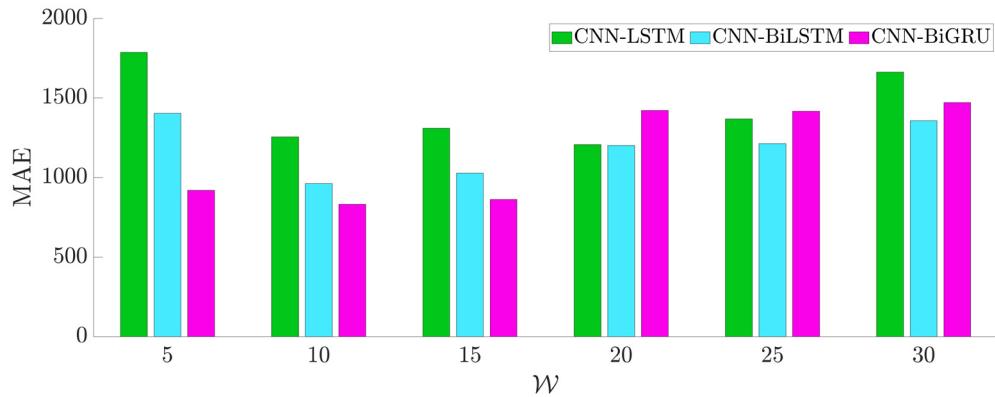


Fig. A7. Impact, in terms of MAE, of the value of the time lag \mathcal{W} on the BTC price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

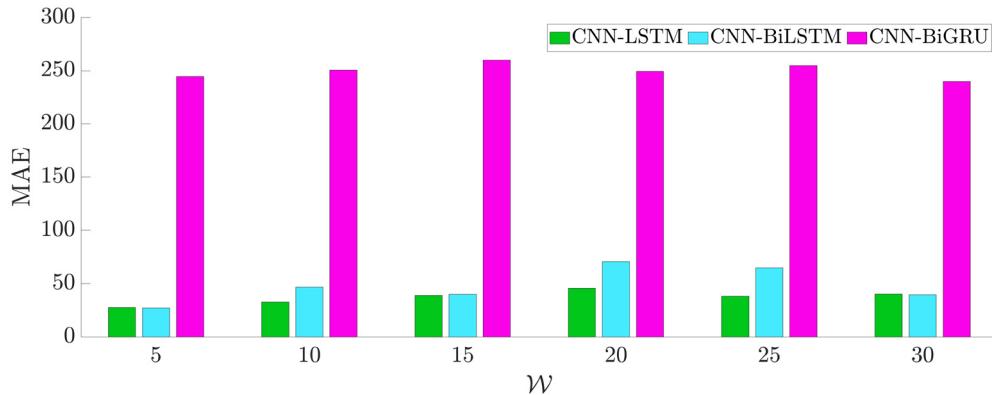


Fig. A8. Impact, in terms of MAE, of the value of the time lag \mathcal{W} on the ETH price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

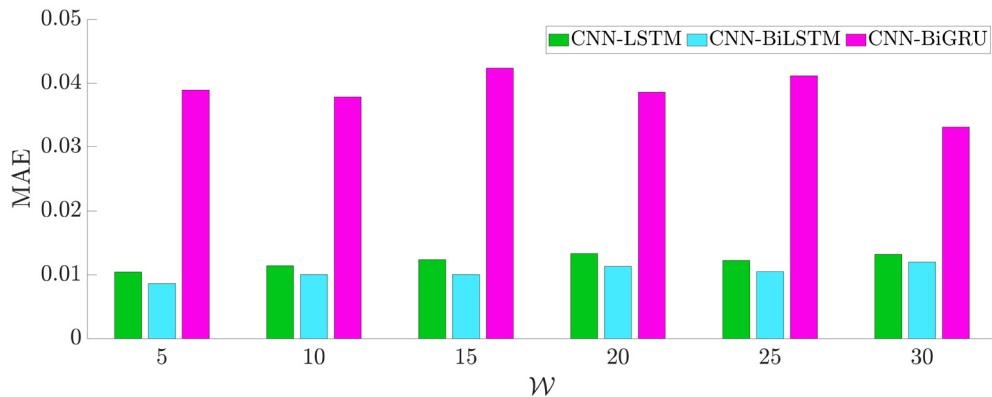


Fig. A9. Impact, in terms of MAE, of the value of the time lag \mathcal{W} on the XRP price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

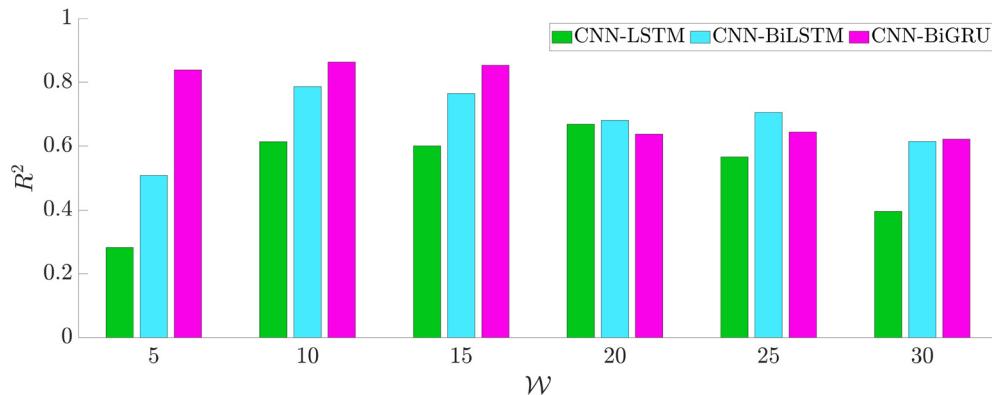


Fig. A10. Impact, in terms of R^2 , of the value of the time lag \mathcal{W} on the BTC price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

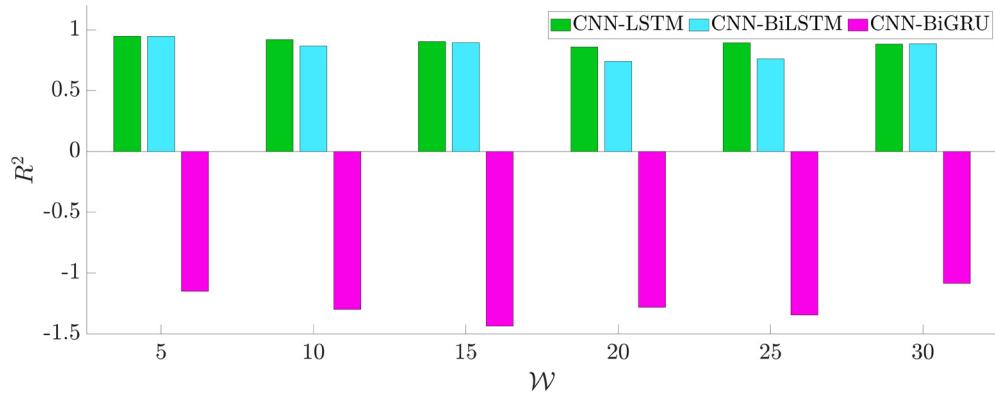


Fig. A11. Impact, in terms of R^2 , of the value of the time lag \mathcal{W} on the ETH price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

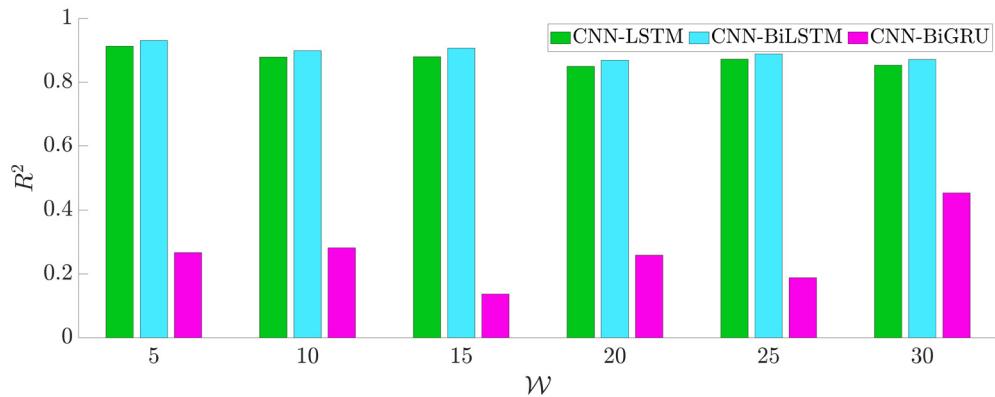


Fig. A12. Impact, in terms of R^2 , of the value of the time lag \mathcal{W} on the XRP price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a 3-step ahead prediction horizon ($s = 3$).

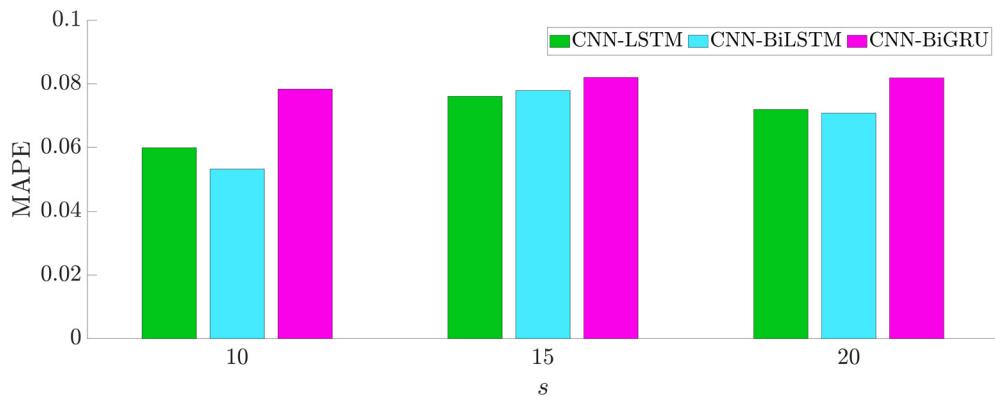


Fig. A13. Impact, in terms of MAPE, of the value of the prediction horizon s on the BTC price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

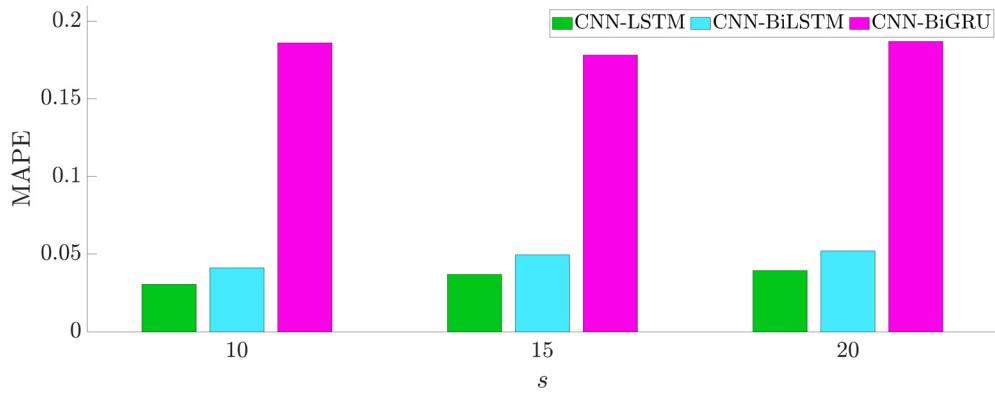


Fig. A14. Impact, in terms of MAPE, of the value of the prediction horizon s on the ETH price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

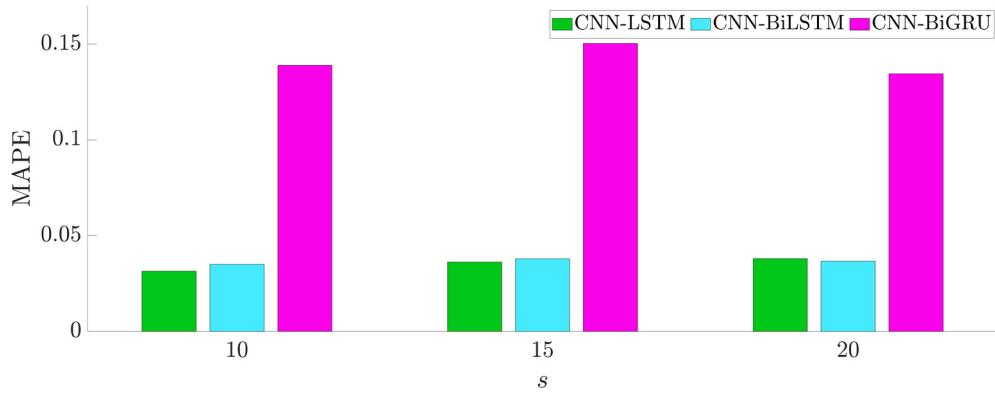


Fig. A15. Impact, in terms of MAPE, of the value of the prediction horizon s on the XRP price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

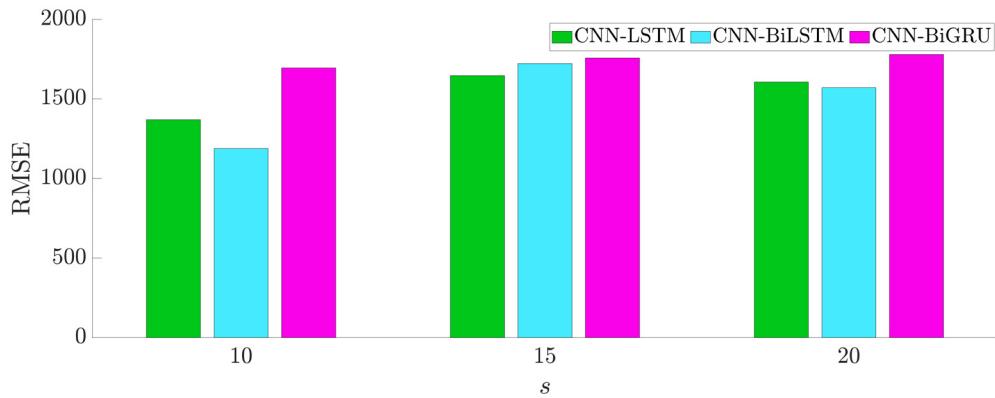


Fig. A16. Impact, in terms of RMSE, of the value of the prediction horizon s on the BTC price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

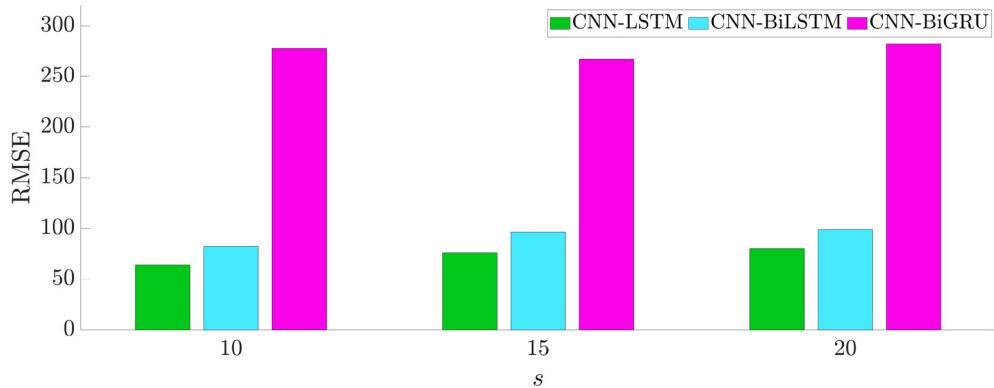


Fig. A17. Impact, in terms of RMSE, of the value of the prediction horizon s on the ETH price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

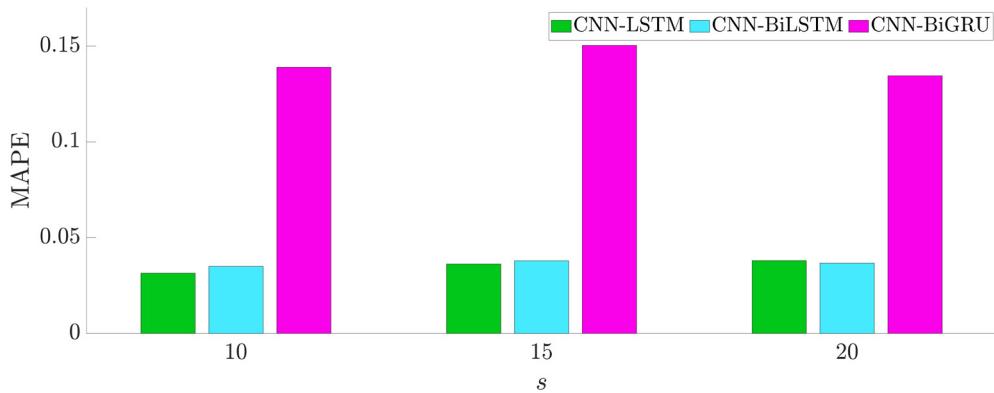


Fig. A18. Impact, in terms of RMSE, of the value of the prediction horizon s on the XRP price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

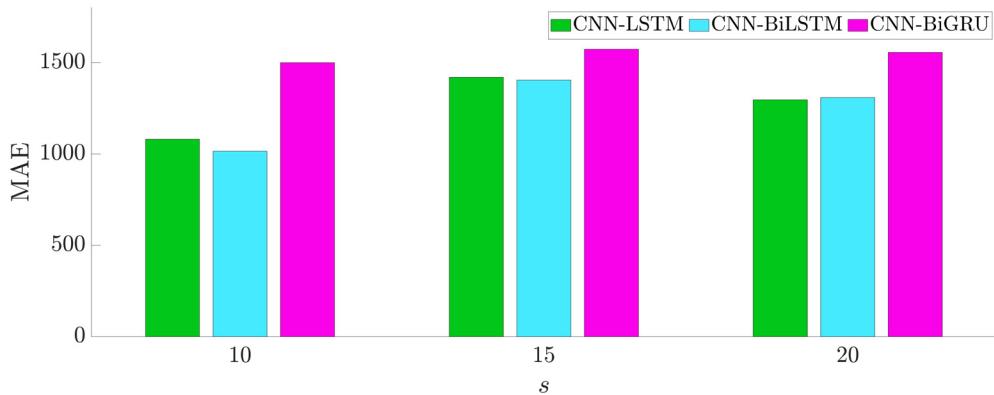


Fig. A19. Impact, in terms of MAE, of the value of the prediction horizon s on the BTC price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

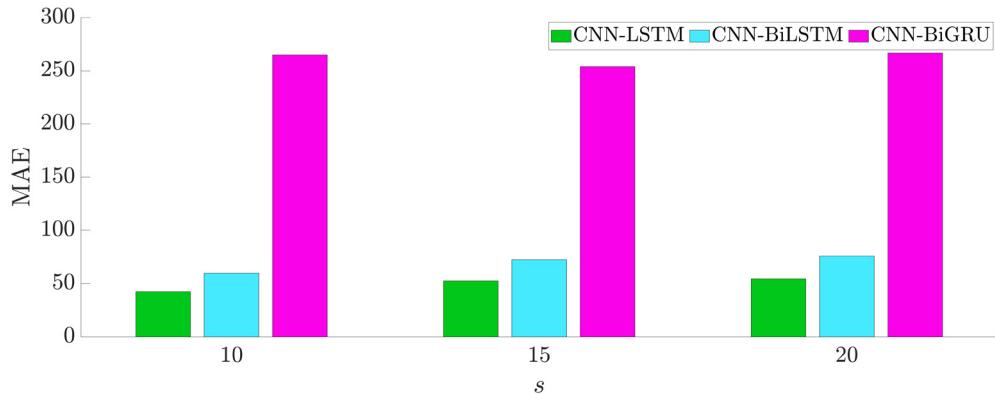


Fig. A20. Impact, in terms of MAE, of the value of the prediction horizon s on the ETH price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

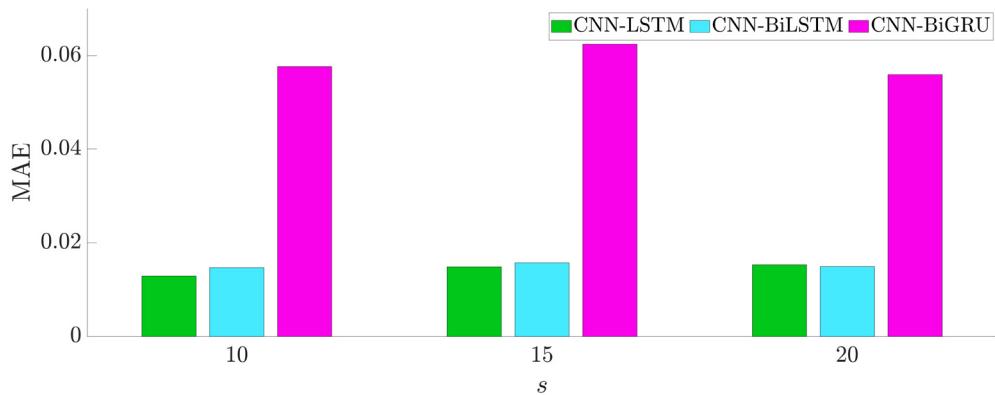


Fig. A21. Impact, in terms of MAE, of the value of the prediction horizon s on the XRP price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

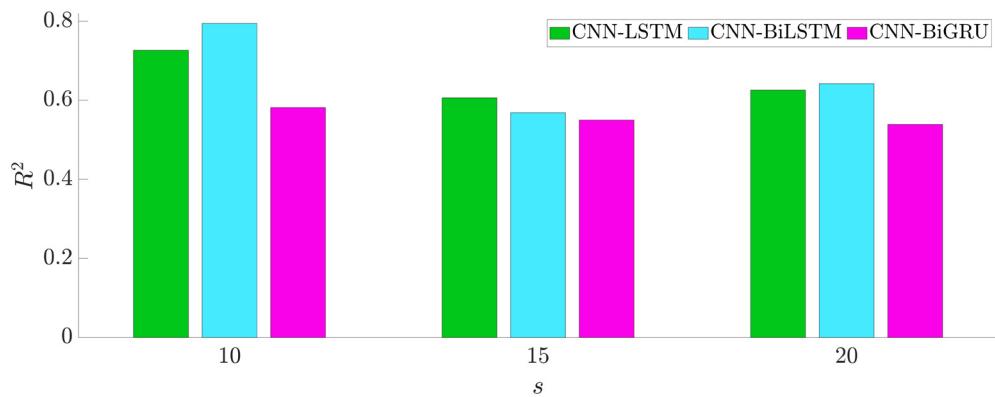


Fig. A22. Impact, in terms of R^2 , of the value of the prediction horizon s on the BTC price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

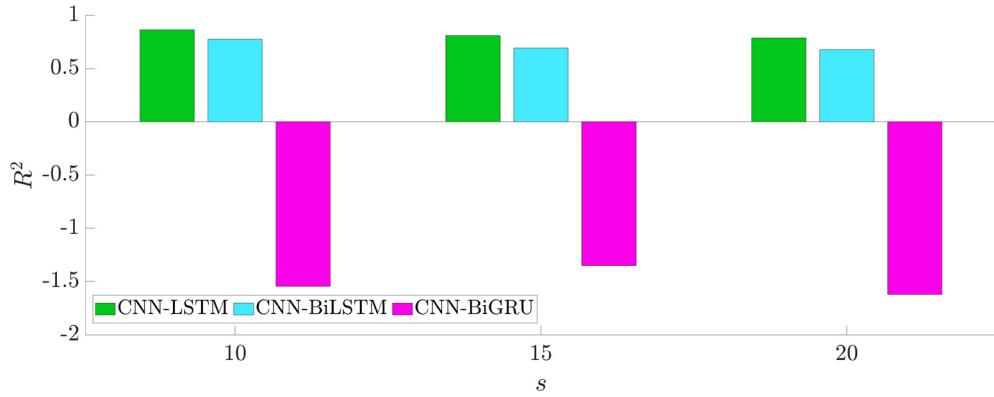


Fig. A23. Impact, in terms of R^2 , of the value of the prediction horizon s on the ETH price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

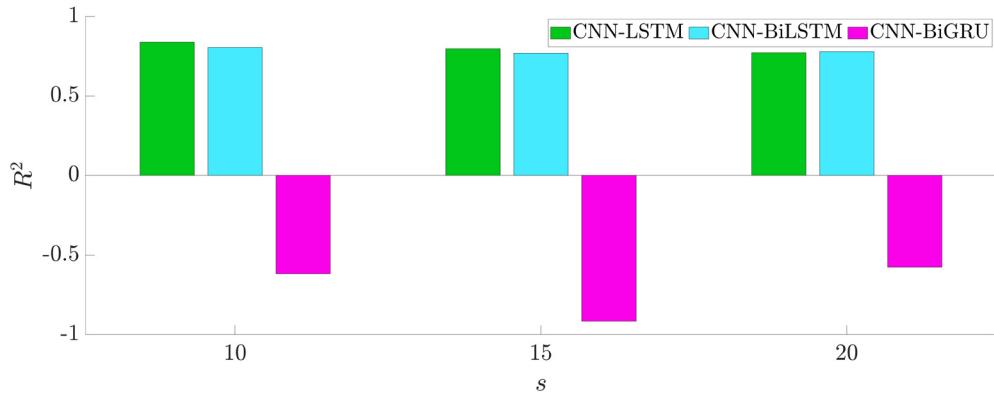


Fig. A24. Impact, in terms of R^2 , of the value of the prediction horizon s on the XRP price prediction with CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU, considering a time lag $\mathcal{W} = 25$.

Received 8 December 2023; revised 11 September 2024; accepted 27 September 2024