Short-term prediction for Ethereum with Deep Neural Networks and Statistical Validation Tests

Eduardo José Costa Lopes, Reinaldo Augusto da Costa Bianchi

[1]Centro Universitário FEI
CEP 09850-901 – São Bernardo do Campo – SP – Brazil

eduardo.lopes@fei.edu.br, rbianchi@fei.edu.br

Abstract. *Cryptocurrency has become a popular asset in global financial markets, meaning that individual investors and asset management companies worldwide are considering this new investment class. The main contribution of this research is to address an intra-day forecasting problem with hourly granularity by comparing deep network architectures, including ones with attention mechanisms for the Ethereum intrinsic cryptocurrency (ETH). Since variations on the deep learning model parameter values may also introduce variability in the results produced by the models, different statistical validations were considered part of the comparison process. Finally, this work shows that the Temporal Convolutional Network model (TCN) outperformed other architectures considered for a short-term forecast period in terms of processing time. The TCN deep learning model is also amongst the most accurate models, using an autoregressive integrated moving average model (ARIMA) as a baseline.*

1. Introduction

Since ancient times, humanity has begun to wonder what lies in the essence of changes in natural events and everything cyclical that happens during their lifetime. It is intrinsic to the human being to foresee future events by analyzing past data. ”The uncertainty surrounding the future is exciting and challenging, with individuals and organizations seeking to minimize risks and maximize utilities. Many forecasting applications call for diverse forecasting methods to tackle real-life challenges”[Petropoulos et al. 2022].

Part of these challenges is related to financial events and decisions. A new class of financial asset that recently came into the picture is a blockchain-based technology called cryptocurrency, which has already been used in a wide range of use cases [Basu 2022]. As another example, individual cryptocurrency miners and financial institutions may have the same need: accurately predict short-term asset price value. There is an unbalance of time availability and processing power between these two sorts of investors. This study is motivated to provide the best possible resources to them when performing timely and accurate time series short-term forecasting activities.

The goal of this research is to address an intra-day forecasting problem with hourly granularity by comparing deep network architectures, including ones with attention mechanisms for the Ethereum intrinsic cryptocurrency (ETH), taking into account different statistical validations models in the models comparison process.

Machine learning and deep learning algorithms have obtained much attention in recent years due to their applicability to many real-life problems, such as fraud detection, spam email filtering, finance, and medical diagnosis. Deep learning models can automatically detect arbitrary complex mappings from inputs to outputs and support multiple inputs and outputs. Considering such powerful technology in this new type of investment class took time.

1.1. Statistical Tests

More than single value accuracy metric results are required to consider whether one deep learning model performs better. Hyper-parameter and model initialization values may significantly affect the results obtained by these models, therefore introducing result variability per model. Typically, the paired T-student test [Student 1908] that is used to compare mean samples is based on particular assumptions or parameters: the data samples meeting those parameters are randomly drawn from an average population, based on independent observations, measured with an interval or ratio scale, possess an adequate sample size, and approximately resemble a normal distribution [Corder and Foreman 2014]. Since these conditions are unknown, the Kolmogorov-Smirnov goodness of fit test [Massey Jr 1951] will be used to determine whether data samples obtained meet acceptable levels of normality, demanding further nonparametric tests..

1.1.1. Kolmogorov-Smirnov one-sample test

It is a procedure to examine the agreement between two sets of values by comparing two cumulative frequency distributions. A cumulative frequency distribution helps find the number of observations above or below a particular value in a data sample. It is calculated by taking a given frequency and adding all the preceding frequencies to the list. Creating cumulative frequency distributions of the observed and empirical frequency distributions allows us to find the point at which these two distributions show the most significant divergence. Then, the test uses the most considerable divergence to identify a two-tailed probability estimate ”p”to determine if the samples are statistically similar or different [Corder and Foreman 2014]. Two-sided tests consider the null hypothesis as if two distributions are identical, F(x)=G(x) for all x, and the alternative is that they are not identical. This study compares the sets of accuracy metric results with normal distribution.

1.1.2. Wilcoxon test

Wilcoxon's nonparametric test is a hypothesis test used to compare pooled samples to determine whether the population mean ranks differ [Derrick and White 2017]. In the original work, the null hypothesis H_0 was defined as the difference of pairs with a symmetric distribution around zero, while H_1 does not. The one-sided test considered in this work has the null hypothesis that the median is positive against the alternative that it is negative.

2. Related work

Most empirical time series (e.g., stock price series and, more recently, cryptocurrencies) behave as though they had no fixed mean. Even so, they exhibit homogeneity in that apart from a local level, or perhaps local level and trend, one part of the series behaves much like any other. Models that describe such homogeneous non-stationary behavior can be obtained by assuming that some suitable process difference is stationary. These facts suggest the use of deep neural networks and an important class of models for which the d^{th} difference of the series is a stationary mixed auto-regressive moving average process. These models are called Auto- Regressive Integrated Moving average (ARIMA) processes [Box et al. 2008] and have been widely used for time series forecasting.

Multiple machine learning methods to predict the Ethereum (ETH) price change direction have recently been studied [Chen et al. 2019]. Deep learning networks are used extensively to forecast the movement of the crypto market. They are "considered to be the most powerful and the most effective methods in approximating extremely complex and non-linear classification and regression problems" [Pintelas et al. 2020b]. According to this author, deep neural networks can deliver modest improvements over traditional machine learning methods. Long Short-term memory model (LSTM) is the most successful and widely used algorithm for prediction, so many authors have used it and improved the prediction accuracy [Tanwar et al. 2021]. A variation of the LSTM model called Gated Recurrent Unit (GRU) [Cho et al. 2014] has also been considered for forecasting tasks. In a study performed by [Awoke et al. 2021], more accurate results were obtained for specific scenarios regarding sliding window size and forecast periods compared to the LSTM deep learning model.

A Convolutional neural network (CNN) is also a class of deep neural networks, most commonly applied to computer vision tasks. Time series can also be handled like a one-dimensional image that a CNN model can analyze [Goodfellow et al. 2016]. Novel approaches applying CNN concepts can be found, like Temporal Convolutional Networks (TCN) [Chen et al. 2020]. It is a convolutional-based probabilistic forecasting framework for multiple related time series that shows both nonparametric and parametric approaches to model the probabilistic distribution based on neural networks. Results from industrial and public data sets show that the framework yields superior performance compared to other state-of-the-art methods in both point and probabilistic forecasting.

Modern Transformer architectures, relying on attention mechanisms are designed to handle sequential data but not necessarily in order. Instead, the attention mechanism provides context for any position in the input sequence. An example of such architecture is Temporal Fusion Transformer (TFT) model that follows an encoder-decoder structure, adds a time encoder to account for sequential data, and does not rely on recurrence and convolutions in order to generate an output. It also combines high-performance multi-horizon forecasting with interpretable insights into temporal dynamics. In order to learn temporal relationships at different scales, TFT uses recurrent layers for local processing and interpretable self-attention layers for long-term dependencies, utilizes specialized components to select relevant features, and a series of gating layers to suppress unnecessary components [Lim et al. 2019]. Prediction for Dogecoin crypto-currency described in [Sridhar and Sanagavarapu 2021] performs an hour-by-hour closing prices model calculation using the Temporal Fusion Transformer architecture. It presented better performance metrics than other deep learning methods applied to other cryptocurrencies. Online Advertising Revenue Forecasting has also benefited from Temporal Fusion Transformer model predictions [Würfel et al. 2021]. This study delivered better performance metrics than LSTM deep learning models for more extended forecast periods.

Pure deep learning models such as N-BEATS added another option for time-series models. It is a deep neural architecture based on backward and forward residual links and a very deep stack of fully-connected layers. This architecture treats forecasting as a non-linear regression problem and outperforms statistical, classical machine learning, and hybrid methods for mid-term electricity load forecasting. It also focuses on solving the univariate times-series point forecasting problem using deep learning [Oreshkin et al. 2021].

The work performed on this paper extends the works by [Hamayel and Owda 2021] and [Agarwal et al. 2021]. Hamayel and Owda (2021) performed a study considering recurrent networks, serving as an initial baseline for this research. The novelty of this study is twofold: (1) the addition of convolutional networks, networks with attention, and residual networks for cryptocurrency price prediction; (2) add a statistical view of the results, given that deep learning model outputs may vary greatly depending on the hyperparameters and seed values considered. Table 1 shows the research performed on the literature for the subject.

3. Methods and Materials

In order to achieve the results of this work, five distinct architectures using Ethereum data were considered. Then, a performance evaluation was conducted with an accuracy comparison against an ARIMA model. In summary, the process consists of five stages: (1) collecting historical cryptocurrency data; (2) data exploration and visualization; (3) training five types of models; (4) generating forecast data using the models; (5) extracting and comparing model accuracy metric results with statistical validation.

3.1. Evaluation Criteria

A combination of both statistical tests and model accuracy metrics were considered.

3.1.1. Accuracy Metrics

According to [Pintelas et al. 2020a], Mean Average Error (MAE) and Root Mean Squared Error (RMSE) are the most used performance metrics to evaluate the regression performance of the forecasting models. In summary, MAE measures the average of the residuals in the dataset, while RMSE measures the standard deviation of residuals. Given that:

- n = number of samples.
- x_i = real value for time (i).
- y_i = forecasted value for time (i).

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - x_i| \tag{1}$$

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2}$$

Since there is no one-size-fits-all performance indicator to evaluate cryptocurrency regression models, mean average percentage error (MAPE), and mean squared error (MSE) are also considered in the analysis. MSE measures the variance of the residuals and the coefficient of determination, and MAPE is used in the hyperparameter tuning process. It has a clear interpretation since percentages are more accessible for people to understand and visualize. At the same time, it does not present shortcomings for our study because the target variable is always a positive value.

$$MSE = \sum_{i=1}^{n}(x_i - y_i)^2 \tag{3}$$

$$MAPE = (\frac{1}{n})\sum_{i=1}^{n}(\frac{x_i - y_i}{x_i}) \tag{4}$$

A historical forecast procedure on the testing data set was selected to calculate the performance metrics, computing the historical forecasts that would have been obtained by the neural network models generated on the series. This process returns the mean value of all the selected accuracy metrics.

3.2. Data set

The data set was collected from glassnode.com, and the feature considered is Ethereum (ETH) closing value with one-hour data resolution. Resolution means the frequency data is updated and the time window over which a metric is aggregated. The metric timestamp is in UTC and always refers to the start of an interval (e.g., data with timestamp 2019-05-13 10:00 UTC includes data from 2019-05-13 10:00 UTC to 2019-05-13 10:59 UTC) [Glassnode 2022].

The entire data period for the ETH series is available from August 2015 until February 2022 and is displayed in Figure 1. January 2020, as the series's starting point, is justified by the increase in the number of transactions. Figure 2 displays the training, validation, and test sets considered in this research (training set - blue, validation set - orange, test set - grey). Table 2 details these sets regarding the number of samples, start and end dates, and the split ratios considered for the selected timeframe.

Tabela 1. Details on training, validation and test data sets.

Data set	Start	End	Samples	(%)
Selected set	01/01/20 00:00	02/07/22 18:00	18,451	100
Training	01/01/20 00:00	06/22/21 02:00	12,915	70
Validation	06/22/21 03:00	10/15/21 09:00	2,767	15
Test	10/15/21 10:00	02/07/22 18:00	2,769	15

3.3. Models and Hyper-parameters

Darts [Herzen et al. 2021] was selected among all the deep network Python packages evaluated for the work performed in this paper. It combines the forecast-related classes of PyTorch with several other packages and facilitates switching between forecast methods, pre-processing, and evaluation tasks.

Figura 1. Ethereum data set - Price on black and transaction volume on blue.

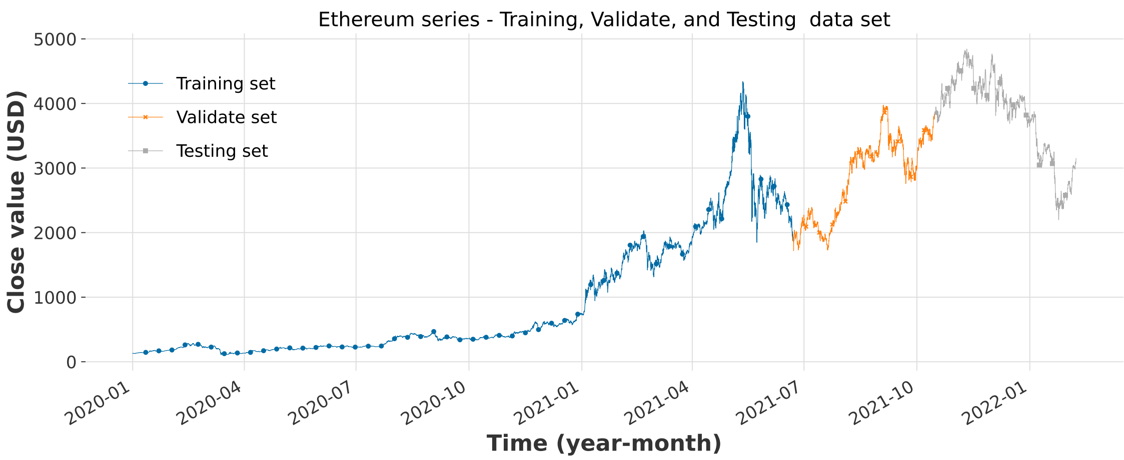

Figura 2. Selected Ethereum data set.

The neural network hyperparameter tuning process was performed using the training data set and the manual search technique. Figures 3 to 7 show the accuracy results obtained for the set of parameters listed as follows:

- Batch size: 16, 32, 48, 64 and 128.
- GRU and LSTM: number of hidden cells and RNN layers.
- TCN: kernel size and dilation base.
- TFT: number of attention heads and LSTM layer hidden size.
- NBEATS: number of stacks.
- ARIMA: number of lags (p).

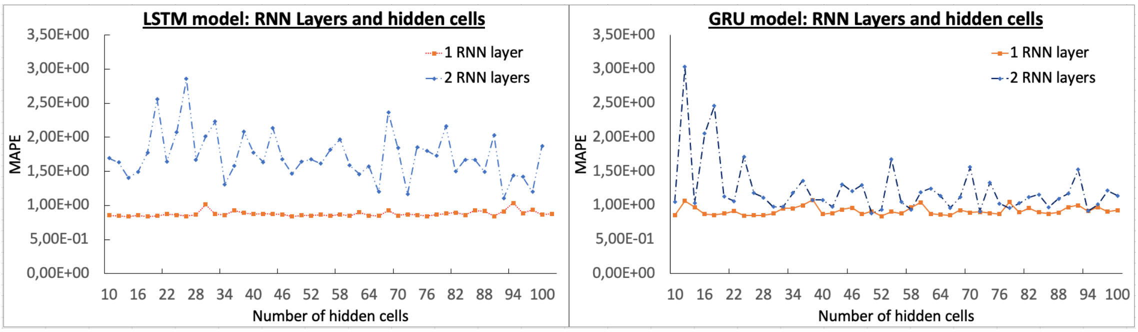

Figura 3. LSTM and GRU hyper-parameter tuning results.

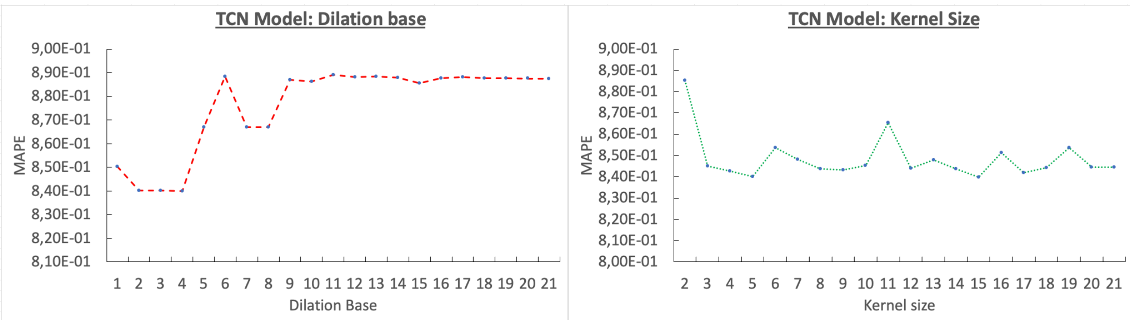

Figura 4. TCN hyper-parameter tuning results.

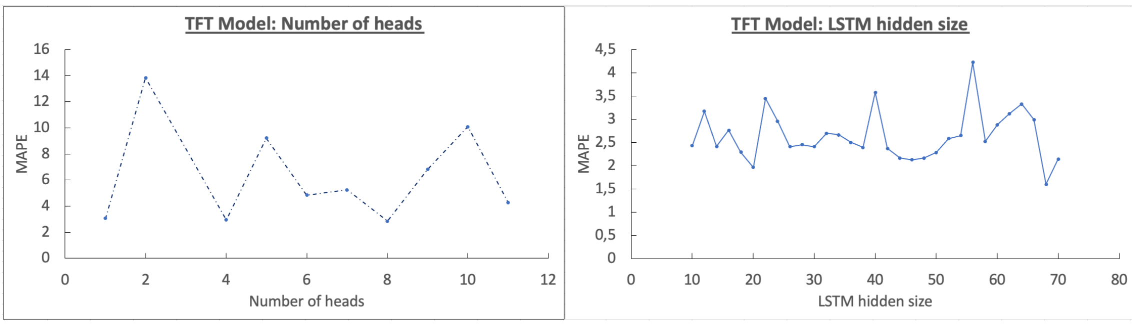

Figura 5. TFT hyper-parameter tuning results.

Mean Average Percentage Error (MAPE) is used in the hyper-parameter selection process. It represents how far the model's predictions are off from their corresponding outputs on average.

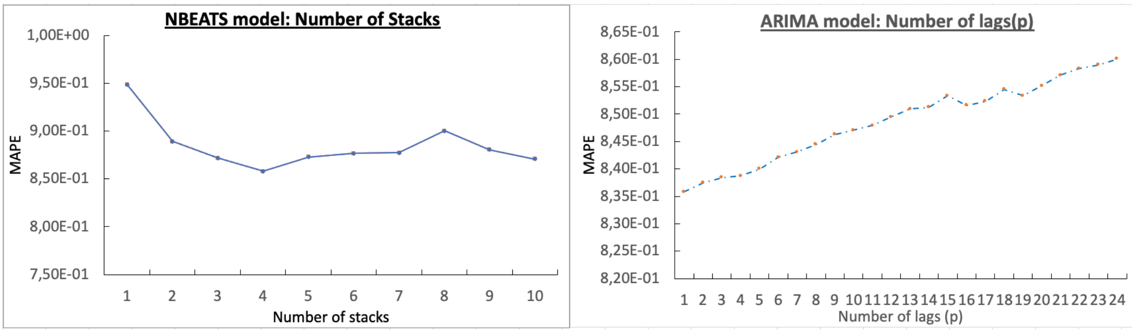

Figura 6. NBEATS - ARIMA hyper-parameter tuning results.

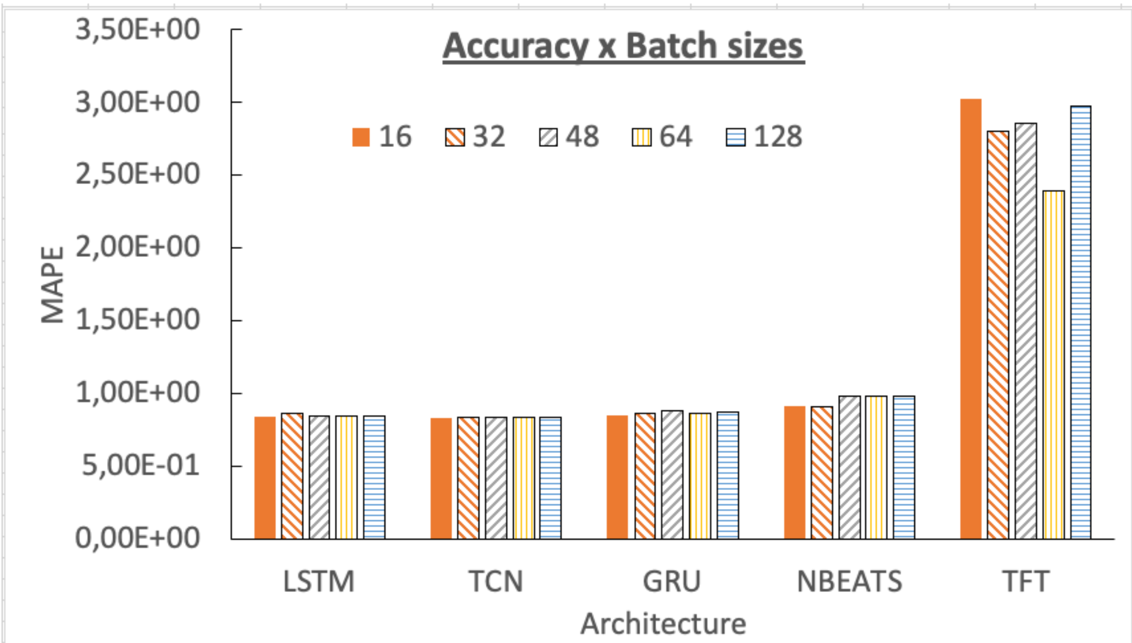

Figura 7. Batch Size hyper-parameter tuning results.

The list of hyperparameter values selected in the tuning process can be found in Figure 8. It is based on the best values obtained on the tests shown in Figures 3 to 7. Some others not considered in the process are empirical values.

A look-back sliding window of length 48 was chosen to generate the input sequences from the data set for the neural networks. The input feature has been normalized by removing and scaling the mean to unit variance. Accuracy metric results did not change significantly among 100, 200, and 500 epochs during the hyper-parameter tuning tasks, so the smallest number of epochs will be considered.

Normalization aims to change the values of numeric columns in the set input data to a standard scale without distorting differences in data intervals. Given that the ETH time series has a considerable variation in the scale of values in different periods, the input variable was normalized by removing the mean and dimensioning it to the unit variation, a procedure recommended by [Hastie et al. 2009]. Only the training dataset was considered for the average calculation, thus avoiding contamination of the validation and test set with data from training.

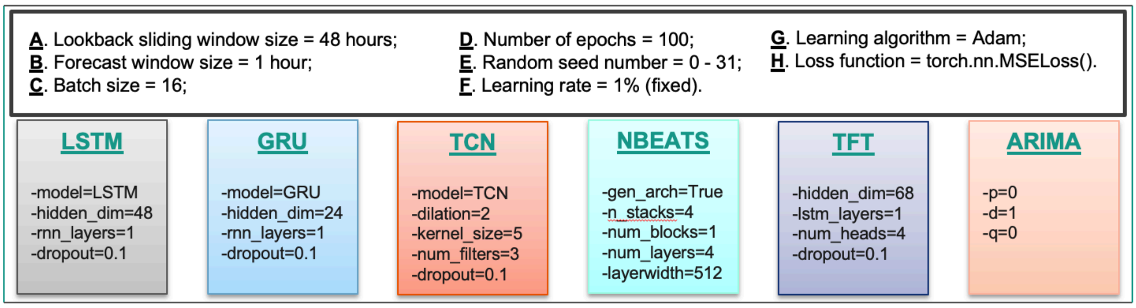

Figura 8. Summary of hyper-parameters considered for models.

Deep neural networks with many parameters are robust machine learning systems. However, overfitting is a severe problem in such networks, and this process occurs when the model has adapted very well to the data it is being trained on. Dropout is a technique to solve this problem [Srivastava et al. 2014], where the main idea is to drop units randomly (along with their connections) from the neural network during training, preventing them from adapting too much. Also, according to this study, dropout values (greater than 0.1) have already shown results for reducing this undesired behavior, and this value was adopted for the deep learning models herein considered.

4. Results and Discussion

Raw results and details about implementation for repeatability purposes are on [Lopes 2022].

Based on Kolmogorov-Smirnov's two-sided tests performed on the complete set of results composed of 32 samples calculated with thirty-two different seed values (0–31), Table 3 shows that none have a normal distribution with 1 percent significance by rejecting the null test hypothesis, justifying the need for nonparametric tests.

Tabela 2. Kolmogorov-Smirnov test results.

2*Model	p-value			
	MSE	MAE	RMSE	MAPE
LSTM	$6{,}37\times10^{-8}$	$5{,}51\times10^{-8}$	$6{,}99\times10^{-26}$	$2,84\times10^{-17}$
TCN	$6{,}36\times10^{-8}$	$5{,}51\times10^{-8}$	$6{,}98\times10^{-26}$	$2,95\times10^{-17}$
GRU	$6{,}36\times10^{-8}$	$5{,}49\times10^{-8}$	$7{,}38\times10^{-26}$	$1,53\times10^{-17}$
NBEATS	$6{,}36\times10^{-8}$	$5{,}50\times10^{-8}$	$7{,}29\times10^{-26}$	$2,19\times10^{-17}$
TFT	$6{,}26\times10^{-8}$	$4{,}11\times10^{-8}$	$1{,}23\times10^{-22}$	$2,18\times10^{-40}$

Tables 3 and 4 show accuracy metric results for Ethereum, median values for model calculations with 31 different seed values used to initialize the proposed deep learning models. The TCN deep learning model delivers more accurate results than the baseline ARIMA model.

The Wilcoxon tests in Table 6 confirm with 1 percent significance that the set of TCN model metrics delivers more accurate results than ARIMA and other deep learning models for ETH. The p-value for each test set between TCN metrics and other models is displayed, and the null hypothesis is rejected (p-values are lesser than significance level $\alpha = 0.01$) for all of them.

Tabela 3. Accuracy metric results (MSE - MAE).

Model-Metric	MSE(↓)	MAE(↓)
ARIMA*	$1,05 \times 10^{-4}$	$7,12 \times 10^{-3}$
TCN	$(5,03 \pm 0,09) \times 10^{-5}$	$(4,96 \pm 0,06) \times 10^{-3}$
LSTM	$(5,23 \pm 0,13) \times 10^{-5}$	$(5,06 \pm 0,10) \times 10^{-3}$
GRU	$(6,03 \pm 0,68) \times 10^{-5}$	$(5,60 \pm 0,41) \times 10^{-3}$
NBEATS	$(5,66 \pm 0,45) \times 10^{-5}$	$(5,24 \pm 0,21) \times 10^{-3}$
TFT	$(1,43 \pm 0,44) \times 10^{-4}$	$(7,02 \pm 1,26) \times 10^{-3}$

Tabela 4. Accuracy metric results (RMSE - MAPE).

Model-Metric	RMSE(↓)	MAPE(%)(↓)
ARIMA*	$1,02 \times 10^{-2}$	$0,835$
TCN	$(7,08 \pm 0,06) \times 10^{-3}$	$(0,579 \pm 0,006)$
LSTM	$(7,22 \pm 0,09) \times 10^{-3}$	$(0,591 \pm 0,010)$
GRU	$(7,75 \pm 0,43) \times 10^{-3}$	$(0,646 \pm 0,042)$
NBEATS	$(7,52 \pm 0,29) \times 10^{-3}$	$(0,613 \pm 0,025)$
TFT	$(1,18 \pm 0,17) \times 10^{-2}$	$(0,852 \pm 0,141)$

The processing time results obtained for the TCN model and displayed in Table 6 validate the work performed by [Bai et al. 2018], showing that a higher level of parallelism is expected compared to other deep learning models.

Wilcoxon tests confirmed that the Temporal Convolutional Network (TCN) deep learning model is the fastest and the most accurate model compared to the other models included in this study. Previous studies have yet to compare the accuracy of each model and prediction time and have yet to demonstrate that results were statistically valid to determine whether a deep learning model performs better than others, given the possible variance.

Tabela 5. Wilcoxon test results.

2***TCN**	**p-value**			
	MSE	**MAE**	**RMSE**	**MAPE**
LSTM	2,76 $\times 10^{-6}$	4,68 $\times 10^{-6}$	2,76 $\times 10^{-6}$	$4,68 \times 10^{-6}$
GRU	3,97 $\times 10^{-7}$	3,97 $\times 10^{-7}$	3,97 $\times 10^{-7}$	$3,97 \times 10^{-7}$
NBEATS	3,97 $\times 10^{-7}$	8,47 $\times 10^{-7}$	3,97 $\times 10^{-7}$	$4,81 \times 10^{-7}$
TFT	3,97 $\times 10^{-7}$	3,97 $\times 10^{-7}$	3,97 $\times 10^{-7}$	$3,97 \times 10^{-7}$

5. Conclusion and Future work

We proposed and validated statistically that one deep neural network produces more accurate results for Ethereum than the ARIMA model used as a reference. Future work may be related to adding more model capabilities and on-chain related data into the analysis, using past and future covariates to achieve even better metric accuracy than univariate models without compromising processing time severely.

Tabela 6. Processing time results in seconds.

Model	Model processing	Forecasting processing
TCN	367	87
GRU	523	96
LSTM	675	97
NBEATS	1,045	172
TFT	2,325	159

We proposed and validated statistically that one deep neural network produces more accurate results for Ethereum than the ARIMA model used as a reference. Future work may include adding more model capabilities and on-chain related data into the analysis, using past and future covariates without compromising processing time.

6. References

Referências

Agarwal, A., Keerthana, S., Reddy, R., and Moqueem, A. (2021). Prediction of bitcoin, litecoin and ethereum trends using state-of-art algorithms. In *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, pages 538–545. ieee. org.

Awoke, T., Rout, M., Mohanty, L., and Satapathy, S. C. (2021). Bitcoin price prediction and analysis using deep learning models. pages 631–640.

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Basu, T. (2022). Ukraine is turning to online crypto crowdfunding to fund its fight against russia.

Box, G., Jenkins, G., and Reisel, G. (2008). Time series analysis-wiley series in probability and statistics.

Chen, M., Narwal, N., and Schultz, M. (2019). Predicting price changes in ethereum. *International Journal on Computer Science and Engineering (IJCSE) ISSN*, pages 0975–3397.

Chen, Y., Kang, Y., Chen, Y., and Wang, Z. (2020). Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing*, 399:491–501.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Corder, G. W. and Foreman, D. I. (2014). *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons.

Derrick, B. and White, P. (2017). Comparing two samples from an individual likert question. *International Journal of Mathematics and Statistics*, 18(3).

Glassnode, W. (2022). On-chain data. `https://docs.glassnode.com/general-info/on-chain-data`. Accessed: 2022-06-09.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Hamayel, M. J. and Owda, A. Y. (2021). A novel cryptocurrency price prediction model using gru, lstm and bi-lstm machine learning algorithms. *AI*, 2(4):477–496.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., and Grosch, G. (2021). Darts: User-friendly modern machine learning for time series.

Lim, B., Arik, S. O., Loeff, N., and Pfister, T. (2019). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363*.

Lopes, E. (2022). Short-term prediction for ethereum with deep neural networks and statistical validation tests - github site.

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Oreshkin, B. N., Dudek, G., Pełka, P., and Turkina, E. (2021). N-beats neural network for mid-term electricity load forecasting. *Applied Energy*, 293:116918.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., et al. (2022). Forecasting: theory and practice. *International Journal of Forecasting*.

Pintelas, E., Livieris, I. E., Stavroyiannis, S., Kotsilieris, T., and Pintelas, P. (2020a). Investigating the problem of cryptocurrency price prediction: a deep learning approach. In *IFIP International conference on artificial intelligence applications and innovations*, pages 99–110. Springer, Springer.

Pintelas, P., Kotsilieris, T., Livieris, I., Pintelas, E., and Stavroyiannis, S. (2020b). Fundamental research questions and proposals on predicting cryptocurrency prices using dnns. Technical report.

Sridhar, S. and Sanagavarapu, S. (2021). Multi-head self-attention transformer for dogecoin price prediction. In *2021 14th International Conference on Human System Interaction (HSI)*, pages 1–6. ieee. org.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Student (1908). The probable error of a mean. *Biometrika*, pages 1–25.

Tanwar, S., Patel, N. P., Patel, S. N., Patel, J. R., Sharma, G., and Davidson, I. E. (2021). Deep learning-based cryptocurrency price prediction scheme with inter-dependent relations. *IEEE Access*, 9:138633–138646.

Würfel, M., Han, Q., and Kaiser, M. (2021). Online advertising revenue forecasting: An interpretable deep learning approach. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1980–1989. IEEE, ieee. org.