Short-term prediction for Ethereum with Deep Neural Networks

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. *The main contribution of this research is to investigate whether an Artificial Neural Network is an option to predict Ethereum cryptocurrency close price on a time constrained scenario. The ANN training time and time lagged data availability are considered as constraints on finding the fastest and the most accurate regression model using ARIMA results as a baseline. As part of the study, hourly aggregated data is processed to generate a step-ahead forecast and then processing time is compared for each architecture. Previous work related to cryptocurrency forecasting usually focus the analysis only on accuracy, and use coarser data granularity. Results have shown that convolutional neural networks over performed other architectures for accuracy and time objectives.*
Keywords: *Cryptocurrency. Ethereum. Neural networks. Deep Learning. Forecasting.*

1. Motivation

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's intrinsic to the human being to foresee future events analyzing past data. "The uncertainty that surrounds the future is both exciting and challenging, with individuals and organisations seeking to minimise risks and maximise utilities. The large number of forecasting applications calls for a diverse set of forecasting methods to tackle real-life challenges" [Petropoulos et al. 2022].

Part of these challenges is definitely 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 being used on 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. Clearly, there is an unbalance of time availability and processing power between these two sorts of investors. A motivation for this study is to provide the best possible resources to them when performing timely and accurately time series short-term forecasting activities.

Information availability is usually on the critical path to perform time sensitive tasks, but blockchain related metric/on-chain data providers usually deliver only near real-time information to the general public. It means that a metric which is aggregated by hour may not be available to everyone exactly at the top of the following hour. These time lags may vary from five to sixty minutes and it is usually linked with increased data granularity. An investor who needs to re-run its prediction models considering the newly available, but delayed, information may be on a disadvantage compared to premium users who have the very same data available in a shorter time. A competitive advantage also arises when an investor knows beforehand which prediction models fit best the data most accurately and on a time constrained scenario. In layman's terms, the sooner and faster an investor receives, and proccesses relevant data, the quicker an investor may be able to make profits.

Machine learning and deep learning algorithms have obtained a lot of 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. It was a matter of time to consider such a powerful technology on this new type of investment class.

This study aims to find an accurate deep learning regression model by comparing recurrent networks, convolutional networks and attention based architectures for a crypto-currency called Ethereum using an ARIMA model as a reference. It considers the prediction processing time together with an estimated time lagged data availability for each architecture, and compare them with each other and with baseline.

The rest of the paper is organized as follows. Section II briefly shows the theory supporting the work performed on this paper, and Section III illustrates detailed studies on the existing methodologies that predict crypto-currency prices and related work. Section IV explains tools and techniques considered as well as details about the data set considered. On Section V the performance evaluation of the proposed models are discussed and finally, Section VI concludes the paper and set the stage for future work.

2. Theory

Time series is a collection of observations indexed by time and some researchers, namely [Chung et al. 2014] among others divide problems and analysis into: forecasting, modelling and anomaly detection. As the name suggests, time series forecasting tries to find the most likely time series values in the future [Lim and Zohren 2021]. Unlike the more straightforward classification and regression problems, time series problems add the complexity of order dependence between observations. As stated by [Pintelas et al. 2020a]: "Cryptocurrency price prediction can be considered as a common type of time series problem, like the stock price prediction" .

2.1. Cryptocurrency

Cryptocurrency is a virtual or digital decentralized currency used in financial systems. It is secured by cryptography that makes it impossible to be counterfeited or double-spent and it is not issued from a central authority or central banks, making it distinguishable from traditional fiat money [Hamayel and Owda 2021]. Bitcoin and Ethereum are the biggest two cryptocurrencies in market capitalization as of May-2022. Ethereum is also an open source, globally decentralized computing infrastructure that executes programs called smart contracts. In simple terms, smart contract is a program stored on a blockchain that run when predetermined conditions are met, and typically used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. While providing high availability, auditability, transparency, and neutrality, cryptocurrencies also reduce or eliminate censorship, and reduce certain counterpart risks. [Antonopoulos and Wood 2018].

Financial markets, which cryptocurrencies are part of, usually use the concept of random walk hypothesis and efficient market to define market efficiency; these two concepts work together under risk neutrality circumstances [Lo and MacKinlay 2011]. There is still no consensus among economists, but some studies argued that efficient markets are an impossibility and the degree of inefficiency determines the effort people are willing to invest to gather and trade on information [Lo 2007]. Besides, different investors understand available information differently which may cause inefficiencies even tough no asymmetry is verified [Hughes-Morgan et al. 2018]. The fact that the random walk hypothesis can be rejected suggests the presence of predictable components on the markets. A study performed by [Palamalai et al. 2021] has shown that major crypto-currencies are not completely random after all, and predictable components do exist in Ethereum.

2.2. Neural Networks

Neural networks are inspired by the human brain [Negnevitsky and Intelligence 2005] and have been used to extract and process complex information, including non-linearity seen in financial time series. Some examples of ANNs are: Recurrent neural networks, Convolutional neural networks, Residual deep learning networks, and neural networks using attention mechanisms. This section covers a high level overview of the neural network main concepts, but also applications of such concepts on inovative network architectures considered on this work.

Recurrent Neural Networks are designed to learn temporal or sequential information by feeding the output from the previous step as an input to the current step and it can keep track of arbitrary long-term dependencies in the input sequences. The problem with standard RNNs lies in the back-propagation mechanism, whose gradients can vanish or explode. The most common types of these networks are long short-term memory (LSTM) and gated recurrent units (GRU), which were both developed to overcome this particular problem.

A LSTM network was also designed to model long range dependencies more accurately than conventional RNNs and contains special units called memory blocks in the recurrent hidden layer. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information. Each memory block in the original architecture contained an input gate and an output gate. The input gate controls the flow of input activation into the memory cell and the output gate controls the output flow of cell activation into the rest of the network. The forget gate scales the internal state of the cell before adding it as input to the cell through the self-recurrent connection of the cell. Gated Recurrent Units is like a long short-term memory but lacks an output gate.

Convolutional neural network is also a class of deep neural networks, most commonly applied to computer vision tasks. But time series can also be handled like a one-dimensional image that a CNN model can analyze. 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 non-parametric and parametric approaches to model the probabilistic distribution based on neural networks. Results from both industrial datasets and public datasets show that the framework yields superior performance compared to other state-of-the-art methods in both point and probabilistic forecasting.

NBEATS is a deep neural architecture based on backward and forward residual links and a very deep stack of fully-connected layers [Oreshkin et al. 2019]. It focuses on solving the univariate times series point forecasting problem using deep learning, and its architecture design methodology relies on a few key principles: the base architecture should be simple and generic, and the architecture should not rely on time-series-specific feature engineering or input scaling. Another goal for this architecture is to explore the potential of pure deep learning architecture in time series forecasting.

Recently deep networks are employing attention mechanisms [Vaswani et al. 2017] on Natural Language Processing (NLP) and and CV (Computer Vision) tasks. Transformer architectures, relying on attention mechanisms, are designed to handle sequential data but not necessarily in order. Rather, the attention mechanism provides context for any position in the input sequence. An example of such architecture is Temporal Fusion Transformer [Lim et al. 2019] which follows an encoder-decoder structure and adds a time encoder to account for sequential data, but 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. 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.

3. Related work

Multiple machine learning methods to predict the ETH price change direction have been studied in the recent years [Chen et al. 2019]. Traditional classification models such as Logistic Regression, Naive Bayes, Support Vector Machine and Random Forest were evaluated. Amongst all models, the ARIMA model performed better. Another work also considered it for predicting ETH price direction for different time aggregation scenarios [Alahmari 2019]. Pre-processing work and re-sampling techniques were required due to series seasonality and trend characteristics and therefore generated results that were time granularity dependant.

Deep learning networks are used extensively to forecast the movement of the crypto market and "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]. Albeit modest, DNNs did deliver an improvement over traditional machine learning methods. 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].

Prediction for Dogecoin crypto-currency described in [Sridhar and Sanagavarapu 2021] performs an hour-by-hour closing prices model calculation using Temporal Fusion Transformer architecture, and present better performance metrics compared to other deep learning methods applied to other crypto-currencies. Online Advertising Revenue Forecasting has also benefited from Temporal Fusion Transformer model predictions [Würfel et al. 2021]. This study was able to deliver better performance metrics compared to a LSTM architecture to longer forecast periods.

Pure deep learning models without a-priori hypothesis such as N-BEATS

[Oreshkin et al. 2019] added another option for time series related models. It treats forecasting as a non-linear regression problem and outperformed statistical, classical machine learning and hybrid methods for mid-term electricity load forecasting [Oreshkin et al. 2021].

The work performed on this paper extends further [Hamayel and Owda 2021] and [Agarwal et al. 2021] work by incorporating novel transformer based temporal architectures into the picture, a convolutional network specialized on time series, as well as NBEATS. Previous work for crypto-currency predictions usually calculate a one-step ahead forecast which may not fulfill the investor's needs to recalculate their models based on the latest data and have forecast information continuously.

4. Methods and Materials

To achieve the results of this work, we trained five distinct architectures using Ethereum data. Then, in order to evaluate the suggested schemes' performances, we compare the accuracy of an ARIMA model to our results. In summary, the proccess consists of five stages: (1) collecting historical cryptocurrency data; (2) data exploration and visualization; (3) training five types of models; (4) generate forecast data using the models; and (5) extracting and comparing the results.

4.1. Dataset

The data set was collected from glassnode.com, a blockchain data and intelligence provider that generates innovative on-chain metrics and tools for digital asset stakeholders. The feature considered is Ethereum (ETH) closing value with hourly granularity. The selected time span reflects the latest available series data at the time of conducting this work, and also the amount of data points needed to have a 70/15/15 data splitting ratios for the training, validation, and testing data set. But the full series is available from August-2015 until February-2022 and have very different behavior accross time. In order to help choosing the best data time span for the work, Table 1 displays the ETH series statistics for average, standard deviation, minimum, and maximum ETH closing values for two different periods. A visual inspection on these values can identify that these statistics differ on orders of magnitude, and the most recent data better reflects the latest asset behavior, further validating the time span selection.

Table 1. ETH Closing value (USD) time series details.

Series Datetime	Aug.15 to Dec.19	Jan.20 to Feb.22
Average	203.00	$1,602.00$
Std. Deviation	241.00	$1,439.00$
Minimum	0.42	106.00
Maximum	$1,416.00$	$4,844.00$

Figure 1 and Table 2 provide further details on the selected time frame. It's worth mentioning that an all-time-high Ethereum behavior can be found on the testing data set, bringing unseen data points to the prediction models.

Figure 1. Selected ETH time series.

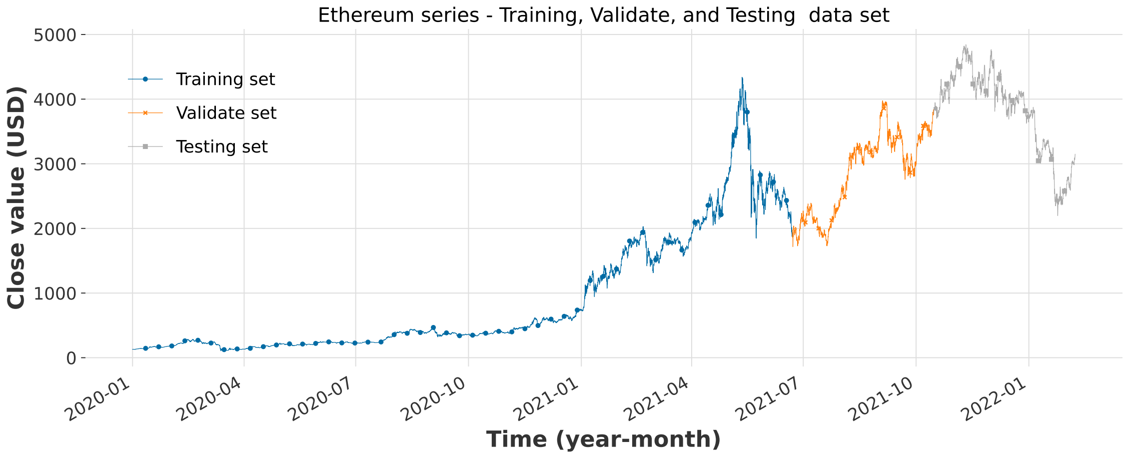


Table 2. Selected ETH time series details.

Time series	Start datetime	End datetime	Data points	Split (%)
Full set	01/01/20 00:00	07/02/22 18:00	18,451	100
Training set	01/01/20 00:00	22/06/21 02:00	12,915	70
Validation set	22/06/21 03:00	15/10/21 09:00	2,767	15
Testing set	15/10/21 10:00	07/02/22 18:00	2,769	15

4.2. Tools and Techniques

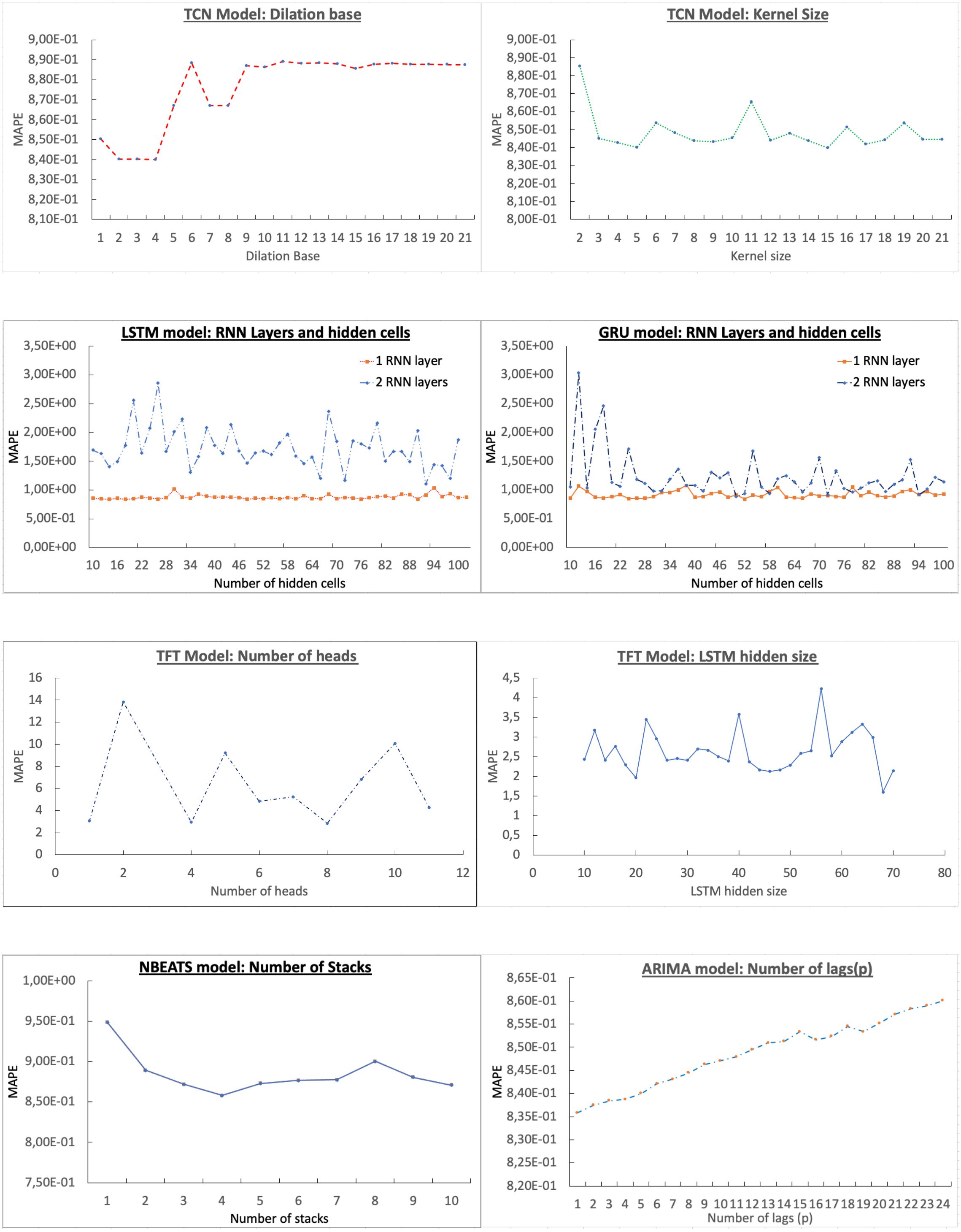
Amongst all the deep network python packages evaluated for the work performed on this paper, darts was selected [Herzen et al. 2021]. It combines the forecast-related classes of PyTorch with several other packages, and facilitates switching between forecast methods, pre-processing, and evaluation tasks. Other packages considered were pytorch-forecasting, pytorch-lightning, TensorFlow, and Flow Forecast. The selection was based on the following criteria: ease of use, deep learning model availability, time series handling features, and documentation availability and accuracy.

Neural network hyper-parameters tuning was performed with the training set using manual search technique, and Figures 2 and 3 show the accuracy results obtained:

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

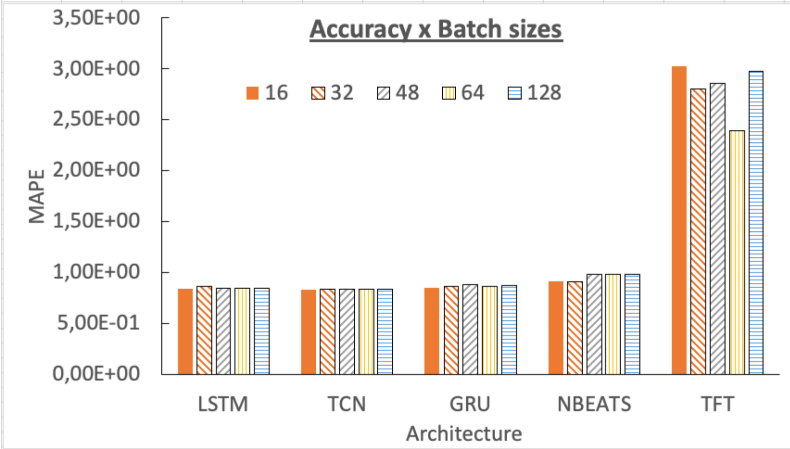
Mean Average Percentage Error (MAPE) is used on the hyper-parameter selection proccess. It represents how far the model's predictions are off from their corresponding outputs on average. Also a lookback sliding window of length 48 which represents 48 hours worth of historical data was chosen to generate the input sequences from the data set for the neural networks. The input feature has been normalized by removing the mean and scaling it to unit variance. Accuracy metric results did not change significantly among

Figure 2. Hyper-parameter tuning results.



100, 200, and 500 epochs during the hyper-parameter tuning tasks so the smallest number of epochs will be considered given the time constraint objective included on this work.

Figure 3. Tuning proccess results for batch sizes and architecture models.



4.2.1. Forecast window size

An important part of the work considers time related information and commercial off-the-self computer hardware to create neural network models and generate forecasts. These two components introduces variability on the processing time results which may impact on forecast data availability. It can be noted on figure 4 that depending on the combined time lagged data duration and model processing time, the forecast window size can be either one or two hours without adding any forecast data gaps. The former will be considered for all models for analysis simplification purposes.

Figure 4. Forecast window size selection.

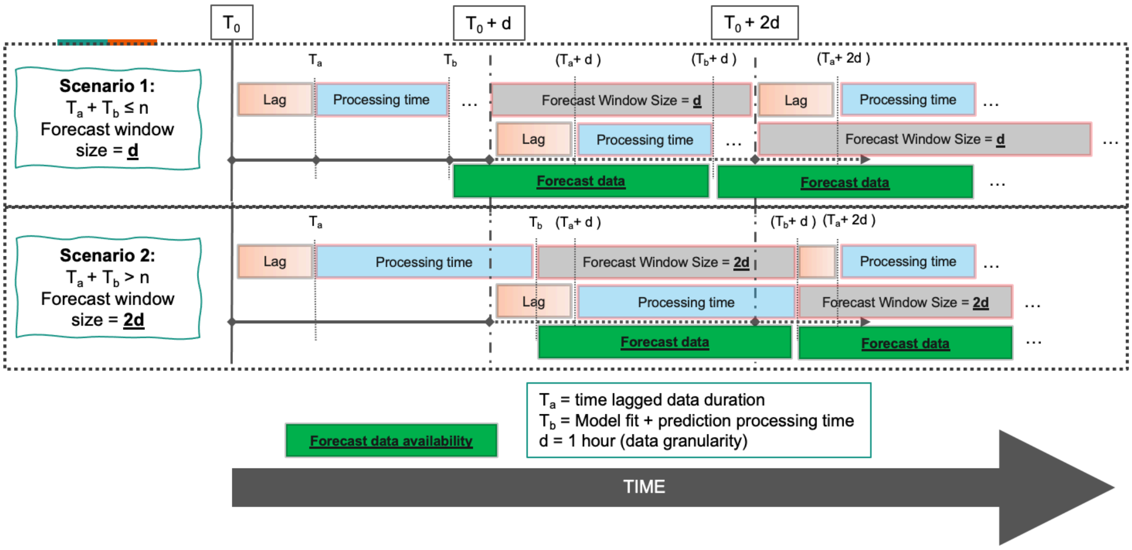
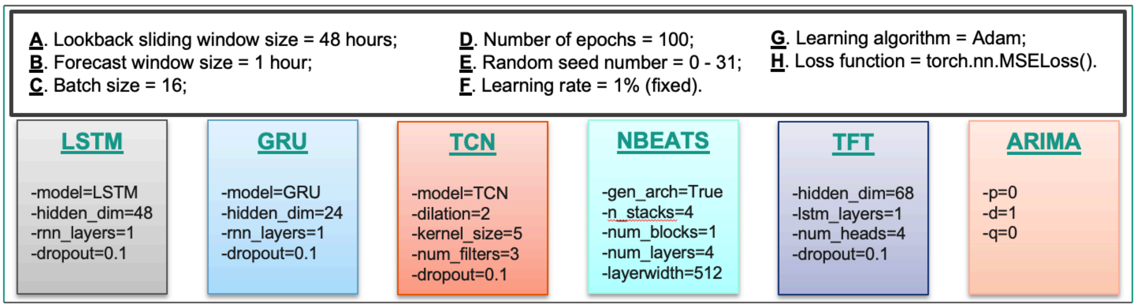


Figure 5 shows a summary of the hyper-parameters list chosen from the tuning proccess for different architectures, and also some parameters which are based on empirical values. Items (A) through (H) compose a common set of values used accross different models, and a different list of specific parameters for each architecture can also be found on this illustration.

Figure 5. List of hyper-parameters.



All the work (data set and Jupyter notebooks) will be provided after revision on the author's Github addresses.

4.3. Evaluation Criteria

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 the measures the average of the residuals in the data set while RMSE measures the standard deviation of residuals.

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

$$RMSE = \sqrt{\tfrac{1}{n} \textstyle\sum_{i=1}^{n} \left(\frac{d_i - f_i}{\sigma_i}\right)^2}$$

Since there is no one-size-fits-all performance indicator to evaluate cryptocurrency regression models, mean average percentage error (MAPE), mean squared error (MSE) and R^2 are also considered on the analysis.

$$MSE = \textstyle\sum_{i=1}^{D} (x_i - y_i)^2$$

$$R^2 = \frac{n \sum xy - \sum x . \sum y}{\sqrt{(n \sum x^2 - (\sum x)^2).(n \sum y^2 - (\sum y)^2)}}$$

MSE measures the variance of the residuals and the coefficient of determination, and R^2 represents the proportion of the variance in the dependent variable and always will be less than one. MAPE is used on the hyper-parameter tuning proccess and has a clear interpretation since percentages are easier for people to understand and visualize. It also does not present its shortcomings for our study because target variable is always a positive value.

$$MAPE = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \left(\frac{A_t - F_t}{A_t}\right)$$

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.

5. Results and Analysis

It is important to point out that the complete and formal complexity analysis of the models is outside the scope of this paper. It can be noted that the TCN architecture delivers more accurate results compared to the baseline model and other architectures considered as well. Particularly for the TFT results, calculated model was not able to forecast an ETH close all-time-high (ATH) value that occurred during the testing data set, explaining higher accuracy metric figures.

Table 3. Results - ETH Testing data set accuracy results. *- Reference model.

Model - Metric	MSE	MAE	RMSE	R^2	MAPE (%)
ARIMA*	1.05×10^{-4}	7.13×10^{-3}	1.02×10^{-2}	9.96×10^{-1}	0.836
TCN	5.02×10^{-5}	4.95×10^{-3}	7.08×10^{-3}	9.98×10^{-1}	0.579
LSTM	5.08×10^{-5}	4.98×10^{-3}	7.13×10^{-3}	9.98×10^{-1}	0.583
GRU	5.35×10^{-5}	5.14×10^{-3}	7.31×10^{-3}	9.98×10^{-1}	0.599
NBEATS	5.50×10^{-5}	5.16×10^{-3}	7.41×10^{-3}	9.98×10^{-1}	0.602
TFT	1.26×10^{-3}	2.12×10^{-2}	3.55×10^{-2}	9.31×10^{-1}	2.26

Table 4. ETH processing time results. *- Reference model.

Specification	**Type 1**	**Type 2**
CPU	i9-3.8 Ghz 10-core	i9-2.4 GHz 8-Core
RAM	32 GB 2666MHz	32 GB 2400MHz
GPU	Nvidia RTX 3090	Not used
System	Win v. 10.0.19041	macOS v.12.2.1
Processing time (seconds)		
ARIMA*	150	179
TCN	457	636
GRU	617	1,270
LSTM	753	1,149
NBEATS	1,210	2,889
TFT	2,461	4,632

RMSE gives a relatively high weight to large errors, meaning that it should be more useful when large errors are particularly undesirable. Along with RMSE, MSE figures also showed different order of magnitude when evaluating architectures, proving to be useful on identifying accuracy differences for the models.

The processing time results obtained for the TCN architecture validates the work performed by [Bai et al. 2018], showing that a higher level of parallelism is expected compared to other architectures. According to the results displayed on Table 4, this model clearly benefits from GPU processing power.

Since the TCN architecture is the fastest and the most accurate model compared to the other models included, it may be considered the most suitable for the scenario proposed by this work. Previous studies have not identified the benefits of the architectures related to the processing and prediction time.

6. Conclusion and Future work

We proposed and validated that a deep neural network produces more accurate results for Ethereum than the ARIMA model used as a reference. A step-ahead forecast window size can ensure continuous forecast information in case the combined lagged data availability and model calculation time is smaller than 1 hour, and it can achieved with two different available off-the-self computer hardware.

Future work may be related to the random parameter initialization for the fit process as it may impact accuracy. Choosing the most convenient set of parameters, including random seeds, together with an statistical result validation analysis may add further value to the results obtained on this paper.

References

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.

Alahmari, S. A. (2019). Using machine learning arima to predict the price of cryptocurrencies. *The ISC International Journal of Information Security*, 11(3):139–144.

Antonopoulos, A. M. and Wood, G. (2018). *Mastering ethereum: building smart contracts and dapps*. O'reilly Media.

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.

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.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

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.

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.

Hughes-Morgan, M., Ferrier, W. J., and Morgan, F. W. (2018). Clear signals or ambiguity? how long-buyers and short-sellers react differently to competitive actions. *Journal of Managerial Issues*, pages 63–81.

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*.

Lim, B. and Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209.

Lo, A. W. (2007). Efficient markets hypothesis.

Lo, A. W. and MacKinlay, A. C. (2011). *A non-random walk down Wall Street*. Princeton University Press.

Negnevitsky, M. and Intelligence, A. (2005). A guide to intelligent systems. *Artificial Intelligence*.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. (2019). N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

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.

Palamalai, S., Kumar, K. K., and Maity, B. (2021). Testing the random walk hypothesis for leading cryptocurrencies. *Borsa Istanbul Review*, 21(3):256–268.

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.

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.

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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

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.