Cryptocurrency Price Prediction With Multi-task Multi-step Sequence-to-Sequence Modeling

Jesper Kristensen*, Juan P. Madrigal-Cianci* †, Giorgos Felekis *, and Maria Liatsikou
* *Advanced Blockchain Research.*
{jesper,juanpablo,giorgos}@advancedblockchain.com.
liatsikou.maria@gmail.com.
†*SB-MATH-CSQI, École Polytechnique Fédérale de Lausanne, Switzerland.*

***Abstract*—In the scientific and commercial worlds, predicting cryptocurrency prices over time has gotten a lot of interest. Most related studies use variations of Recurrent Neural Networks to forecast the next value of a single coin due to the temporal nature of the challenge. As a result, determining how effectively such a model would perform across various tasks (cryptocurrencies), several future timesteps, and forecasting horizons will be difficult. This paper proposes a multi-task and multi-step sequence-to-sequence model that is trained jointly on 22 cryptocurrencies' time series. Our findings show the value of sequence-to-sequence modeling for future predictions, as well as the significant improvements in accuracy and training time that can be achieved by using a single multi-task model rather than numerous distinct models for each task.**

1. Introduction

The recent advances in financial technology, cryptography, and distributed ledger technology have led to the establishment of cryptocurrencies [1] as one of the pillars of Web3 [2], Decentralized Finance (DeFi) [3], decentralized applications (dApps) [4], and NFTs [5]. These cryptocurrencies are digital currencies used in decentralized and traditional financial systems. Over the last years, they have gained massive popularity as investment assets and as means of exchange. One of the great benefits of cryptocurrencies is that they are secured by cryptography which makes them non-forgeable [6].

Nowadays, billions worth of transactions are happening on cryptocurrencies, and as such, there is an increased interest in algorithmic trading and price forecasting. The problem of accurate cryptocurrency price predictions is challenging due to high non-Gaussian volatility. Most of the traditional variables that usually affect stocks have nothing to do with them. Compared to fiat, the assets of cryptocurrencies are extremely dynamic and present high volatility and fluctuation. The factors related to the price movement can be blurry, but they have to do with the blockchain network's mining cost, market trends, prices of other currencies, world events, and social media content. Sovbetov et al. [7] considered many technical factors that influence the prices and trading volume of Bitcoin, Ethereum, Dash, Litcoin, and Monero, concluding that *positivity is higher than negativity, and there exist relations between price changes and attitudes*. Furthermore, Narman et al. [8] described how positive and negative comments on social media affect the prices of cryptocurrencies.

Although the extreme fluctuation of cryptocurrency prices and their high cross-correlation creates fertile ground for significant profits (and great losses), it also makes it extremely challenging to develop intelligent low-risk strategies that will benefit investors. As expected, simple statistical methods for price forecasting fail to capture the complexity of the price movements, which are strongly affected by "whale moves" and external world events. Hence, to improve forecasting abilities, there is a clear motivation to consider, beyond any external information, the development of more advanced machine learning methods, especially *Deep Learning* (DL) algorithms suited to sequential multivariate multi-output problems.

Accurately forecasting prices and other related time-series data is also highly relevant to developing inter-blockchain bridging algorithms [9] where liquidity distributions need to be forecasted to ensure smooth user experiences. In the realm of cross-chain trading, decentralized exchange (DEX) aggregators [10] [11] need to account for delays in transfers, originating from having to reach consensus, can change the underlying state of the networks including the exchange rates, leading to unfavorable pricing conditions which can be mitigated through accurate forecasting.

Related Work

Many researchers have recently tried to perform forecasting on the crypto market using various statistical techniques and machine learning algorithms. A large body of research is around works using various traditional time series analysis techniques, such as *ARIMA* and *GARCH*, trying to predict the price of *bitcoin* as the ones in [12], [13], [14], and [15].

However, in recent years deep learning methods have become the dominant approach when it comes to forecasting the dynamics of financial markets [16], and the same applies to crypto markets. To this end, the work in this domain is

978-1-6654-6104-7/22/$31.00 ©2022 IEEE
DOI 10.1109/Blockchain55522.2022.00018

mainly focused on the implementation of *recurrent neural networks* (RNNs), which are most efficient across standard approaches when it comes to the price forecasting tasks [17], [18], [19], [20], [21], [22] and [23]. A big body of research also focuses on the study of DL models by specifically using *sentiment analysis* on social media content [24] [25].

Finally, and closer to our work are the studies in [26] and [27]. Here, both authors proposed DL-based methods, which include the implementation of GRU, RNN, and LSTM models for predicting the prices of different types of cryptocurrencies in isolation, such as bitcoin, litecoin, and ether in the first work and Litelcoin and zcash in the second. By examining their results, we notice that the *gated recurrent unit* (GRU) model predicted better than the long short-term memory (LSTM) ones. However, these papers consider cryptocurrencies in isolation when predicting their price, thus failing to borrow statistical strength from the time series of multiple cryptocurrencies, have limited evaluation on restricted forecasting step sizes and horizons, limited or no baseline comparisons, and do not consider sequence-to-sequence architectures.

Contributions

The primary goal of this paper is to introduce a state-of-the-art deep learning forecasting approach to the crypto price forecasting domain and provide a thorough comparison and analysis across multiple cryptocurrencies and forecasting horizons. The specific contributions of our work are the following:

1) A detailed study of fine-grained predictions every eight hours, compared to the typical one-day predictions found in prior art, in a multi-step sequence-to-sequence setting which can easily generalize further for future predictions.

2) A novel approach through a multi-task setting that jointly predicts the price of 22 different cryptocurrencies, which also makes our study one of the most extensive ones regarding the number of the examined assets.

3) A comparison of our proposed models against simplistic and traditional baselines, such as naïve last day predictor, Random Forests, and LSTMs using evaluation matrices such as MAPE, MSE, and Coefficient of Determination (r^2).

2. Methodology

2.1. Long Short Term Memory: A Primer

LSTM is a type of RNN that mitigates the vanishing gradient problem of RNNs [29]. Each LSTM cell has two main types of states: the cell state c captures information from the entire sequence until the current timestep, whereas the hidden state h captures information from the previous timestep. To calculate these states, the following operations take place:

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f), \\ i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \\ \widetilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c), \end{aligned}$$

where f, i, and o are referred to as ‘forget’, ‘input’, and ‘output’ gates, respectively, whereas σ_g and σ_c denote the sigmoid and hyperbolic tangent function, respectively. The cell and hidden states are then calculated as:

$$c_t = f_t \circ c_{t-1} + i_t \circ \widetilde{c}_t,$$
$$h_t = o_t \circ \sigma_h c_t,$$

where σ_h is the hyperbolic tangent function and $\circ$ denotes the Hadamard product.

In our modelling, we use an LSTM operating on the past N historical values of a cryptocurrency and the output hidden state of the last timestep is passed to a second LSTM layer. The outputs of the second LSTM at each timestep are followed by a dense layer to make a timestep-level prediction $f^*_{t_k}$ (see Figure 1, with k=[N,$N+1$,$N+2$]). Although the input is processed sequentially, non sequence-to-sequence LSTMs lack of the ability to model the output in a sequential manner explicitly.

2.2. Sequence-to-Sequence

Sequence-to-sequence models follow an encoder-decoder architecture [30]. In such models, the ‘encoder’ models the input via some neural architecture (in our case, an LSTM), and the hidden and cell states of the last timestamp (T_{N-1} in Figure 1) are used to initialize the respective states of the decoder. The ‘decoder’ then starts operating in a sequential manner, where at inference time, the output on the k-th (future prediction) timestep is used as an input for the (k+1)-th timestep. During training time, the actual values of the target for timestep $k-1$ are used as an input to the k timestep (e.g., f_{t_N} – and not $f^*_{t_N}$ shown in Figure 1 – are used as an input to the second timestep of the decoder). $< START >$ in Figure 1 is a placeholder for the input to the very first timestep of the decoder (in our modeling, we use $x_{t_{N-1}}$).

2.3. Multi-Tasking with Sequence-to-Sequence

An extension of the encoder-decoder architecture described above, is that of *Multi-Task Learning* (MTL) [28]. Loosely speaking, MTL aims at improving the generalization performance of a task using other related tasks. MTL involves training a model to perform more than one prediction task from a given input. Its implementation could be based on, e.g., using a single encoder and then feeding the intermediate representation obtained by this encoder to several other separate decoders performing different output tasks. This, in turn, makes the encoder influenced by

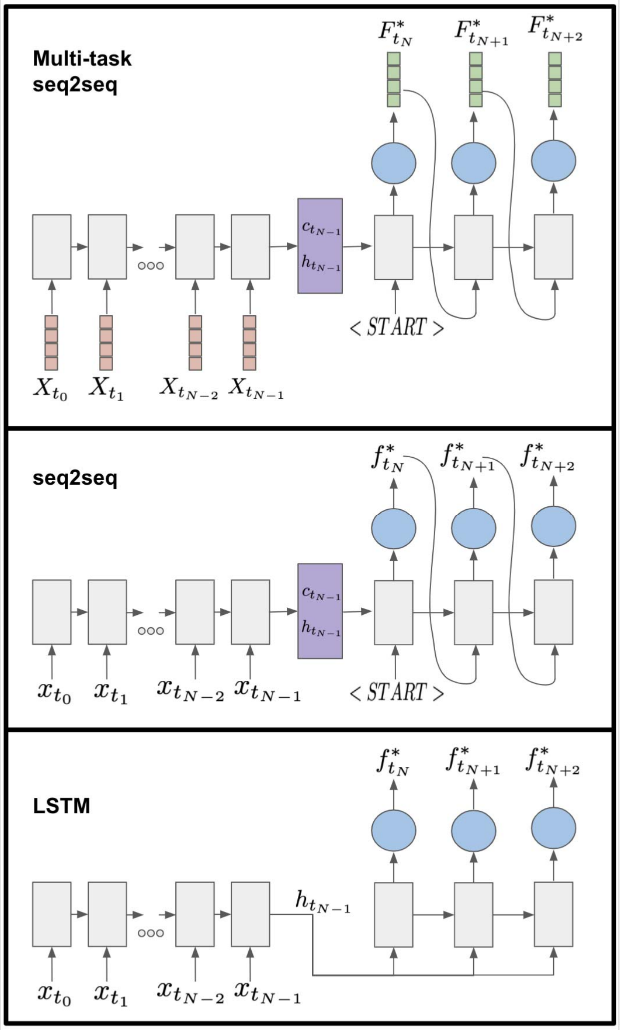


Figure 1: LSTM (below) vs sequence-to-sequence (middle) vs Multi-task sequence-to-sequence (above) models, for a prediction task of 3 future steps given the past N steps. In sequence-to-sequence, the states of the encoder are used to initialise the decoder, while during inference the prediction for the k-th timestep is used as an input to predict the (k+1)-th timestep.

both tasks during training. Sequence-to-sequence MTL has successfully been employed in multiple domains, including language and vision [33], [34].

In our task, we utilize a multi-task setting, where each task is defined on a per-cryptocurrency basis. In particular, instead of using the past values of a single cryptocurrency to predict its next k values, we feed the past values of multiple cryptocurrencies, aiming to predict their next k values at the same time under the sequence-to-sequence model architecture described above (see Figure 1). The benefits of our proposed model are two-fold: (a) by modeling the input and output of multiple currencies at the same time, our model can capture cross-correlation signals and borrow strength from such highly related tasks; (b) the model requires less time for training compared to training C related tasks (C being the number of cryptocurrencies), since it converges much faster due to its ability to approximate the generative function of the time-series easier.

3. Experiments

3.1. Task Definition

Our goal is to predict the price of each cryptocurrency in the next $|f|$ time intervals based on its value in the previous $|h|$ time intervals (see section 3.2, Instance Generation for the exact definitions). We aim at demonstrating the gains in performance by utilizing (a) a sequence-to-sequence model and (b) integrating that into a multi-task approach for predicting the $|f|$ future prices of multiple cryptocurrencies.

3.2. Dataset

We employ a large dataset containing information about the prices of numerous cryptocurrencies at the minute level.[1] In particular, for each cryptocurrency, the data entails timestamped information about the high, low, opening and closing price as well as the volume of trade. In this paper, we focus on the closing price of the cryptocurrencies, translated into USD. However, our proposed models are easily extensible to work with multiple input features.

Preprocessing
We focus on the period between the 1st of January 2020 and the 31st of December 2021 (two years). This period is characterized by high volatility in the fluctuations of cryptocurrency prices during the COVID-19 pandemic (see Figure 2). Aiming at building models that assess the value of a given cryptocurrency at a fine-grained level, we firstly divide our input data into hourly intervals and define the closing price of a cryptocurrency during each interval as the average price observed within it. Then, we generate our time series based on 8-hour intervals (i.e., considering every 8^{th} hourly interval) so that we have three closing prices for each cryptocurrency on each day. Due to missing data, we only keep the 22 cryptocurrencies at least 99.5% complete and linearly interpolate any missing values in the remaining (max) 0.5% of their time series. Lastly, we scale the resulting time series [0,1]. In particular, given the time series $\{x_0, x_1, ..., x_N\}$ of a particular cryptocurrency, we transform it into:

$$\hat{x}_t = \frac{x_t - \min(x_t)}{\max(x_t) - \min(x_t)}, \tag{1}$$

where min and max correspond to the minimum and maximum value of that cryptocurrency over time, respectively.

1. https://www.kaggle.com/datasets/tencars/392-crypto-currency-pairs-at-minute-resolution

Instance Generation
Our goal is to predict the future values of any given cryptocurrency in $|f|$ future steps ahead, given its $|h|$ historical values. We thus conventionally set $h = 12$ (eight-hour intervals, based on the past four days) and $f = 3$ (future one day). Using a stride of one to create our instances, formally, given the normalised time-series that results from eq. 1 $\{\hat{x}_0, \hat{x}_1, \hat{x}_2, ..., \hat{x}_N\}$ for a particular cryptocurrency, our input:output pairs are defined as:

$$
\begin{aligned}
[\hat{x}_0, \hat{x}_1, \hat{x}_2, ..., \hat{x}_{11}] &: [\hat{x}_{12}, \hat{x}_{13}, \hat{x}_{14}] \\
[\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_{12}] &: [\hat{x}_{13}, \hat{x}_{14}, \hat{x}_{15}] \\
&... \\
[\hat{x}_{N-16}, \hat{x}_{N-15}, \hat{x}_{N-14}, ..., \hat{x}_{N-4}] &: [\hat{x}_{N-3}, \hat{x}_{N-2}, \hat{x}_{N-1}].
\end{aligned} \tag{2}
$$

In the following subsection, temporally informed models – such as those presented in section 2 – can sequentially model the task, thus exploiting the ordering of the input/output variables. In particular, sequence-to-sequence models can fully exploit the relationship in the outputs since the predictions made for one timestep are used to predict the output value of the next timestep. We will contrast the performance we obtain through sequence-to-sequence-based models against the performance of feature-based approaches, treating the element of the input/output as independent variables.

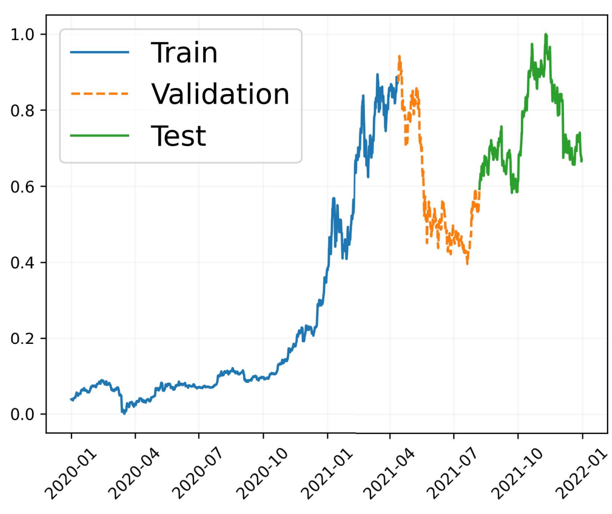


Figure 2: Normalised USD price of BTC over our entire dataset, split into train, validation, and test set.

Train/Validation/Test Split
We use the first 80% of the instances in eq. 2 for training purposes and the remaining (last) 20% for our test set, where we only evaluate our models on. We further break our training set into a purely training and a validation set (using the last 20% of the training data).

3.3. Models

We contrast the performance of our proposed model against various practices commonly found in related work, as well as simplistic, yet highly competitive, baselines from the real world, namely:

- ***Last Values Predictor* (LV)**. LV is a crude baseline, predicting the values of our target's next k timesteps to be equal to the last k timesteps seen in the input space in a one-by-one mapping. Since we model the task so that we aim at predicting the following three values of eight-hour intervals, LV assumes that the prices of any cryptocurrency on the following day will be identical to the ones seen on the last day at the time of making the prediction. Although simplistic, we show that such an approach is highly competitive against state-of-the-art methods, primarily overlooked in related work. Outperforming such baselines in practice is essential for assessing any model's effectiveness.
- **Random Forest (RF)**. RF [31] is an ensemble model, constructing multiple decision trees during training time, the outputs of which is later combined in order to make the final prediction on a given test set. We set the number of trees equal to 200 and train RF to independently predict each cryptocurrency's following three values. We note that autoregressive features can be constructed and incorporated inside the trees. As opposed to our proposed methodology, RF does not naturally consider the temporal dependence between the input values, thus treating the points in the time series as independent features.
- **Feed Forward Neural Network (FF)**. We train a fully connected FF neural network with two intermediate layers (each with 32 units, linear activation) and an output layer (3 units, one per future timestep to predict). As in the case of Random Forest, FF treats the input and output as independent features, failing to consider their temporal dependence.
- **Long Short-Term Memory (LSTM)**. We employ a two-layer LSTM. The final output of the first (16-unit) layer is retrieved as an input to the second (3-unit) layer, followed by a Dense layer of one unit (per timestep) for the final prediction, as described in section 2.1. We experimented with increasing the number of units and adding more layers, but these harmed the performance of the validation set.
- **Sequence to Sequence (seq2seq/s2s)**. Our first proposed model is a sequence-to-sequence approach based on LSTMs, as described in section 2.2. We use 16 units for the LSTMs in the encoder and decoder. In the decoder's final stage, we add a Dense layer, making one prediction at a time (i.e., one per timestep).

- **Multi-Task Sequence to Sequence (MTs2s)**. Finally, our proposed multi-task sequence-to-sequence model (see section 2.3) follows the same setting with s2s, albeit operating on all of the available cryptocurrencies simultaneously – yet, in a sequential manner across timesteps. Due to the price ranges across the different cryptocurrencies, we transform each cryptocurrency's time series on its first-order residuals and convert it to the original values after the prediction step.

All of our DL models have been trained by minimizing the Mean Squared Error (MSE) loss using Adam [32] as our optimizer for 1k epochs, with an early stopping criterion being triggered after 100 epochs without improvement on the validation loss.

3.4. Evaluation Metrics

We assess and contrast the performance of MTs2s against our baselines using different evaluation metrics, each capturing a different aspect of the task. We denote a model's predictions on the test set as the mapping $x^{(c)}_{i,t} \mapsto f((x^{(c)}_{i,t}))$. Here, c indexes the the cryptocurrency under consideration and $i \in \mathbb{N}$, $t \in [1, 2, 3]$ indicate the instance and (future/prediction) time-step index, respectively. We compare the predicted values of our model against the real values given by $y^{(c)}_{i,t} \in \mathbb{R}$. To do so, we consider several standard measures of error, namely:

Mean Absolute Percentage Error (MAPE)
MAPE measures the average of the absolute value of the difference between the predicted data $f(x^{(c)}_{i,t})$ and the measured data $y^{(c)}_{i,t}$, written as a percentage of the actual value. More precisely, provided $y^{(c)}_{i,t} \neq 0$, we have that for any (future/prediction) timestep t:

$$\text{MAPE}^{(c)}_t = \frac{1}{N} \sum_{i}^{N} \left| \frac{f(x^{(c)}_{i,t}) - y^{(c)}_{i,t}}{y^{(c)}_{i,t}} \right|,$$

where N stands for the total number of instances in the test set.

Mean Squared Error (MSE)
MSE assesses model performance on the basis of the errors made in the predictions, by averaging over the squares of the errors, i.e., this is the square of the ℓ_2-norm and can be written as:

$$\text{MSE}^{(c)}_t = \frac{1}{N} \sum_{i} (f(x^{(c)}_{i,t}) - y^{(c)}_{i,t})^2.$$

Coefficient of Determination (r^2)
The r^2 measures how well a model can predict the different values of the target – in particular, how better our model's predictions are compared to the average (target) predictor:

$$r^{2\,(c)}_t = 1 - \frac{\sum_i (f(x^{(c)}_{i,t}) - y^{(c)}_{i,t})^2}{\sum_i (\hat{y}^{(c)} - y^{(c)}_{i,t})^2},$$

where $\hat{y}^{(c)}$ is the average of the actual values seen on the test set. As opposed to MAPE and MSE, higher r^2 values indicate a better model.

Working on each cryptocurrency independently, all of our metrics are calculated on each future timestep individually and are then averaged across the three timesteps we aim at predicting.

4. Results

Tables 1, 3, and 4 show the results based on the three metrics for each cryptocurrency (averaged across the three timesteps) as well as for the 'average' cryptocurrency ('Median', at the bottom row of each table). MTs2s dominate in all metrics, offering a clear-cut relative improvement of 23%, 4%, and 35% in MAPE, r^2, and MSE compared to the LV baseline. The effect of multi-tasking across the different cryptocurrencies is more clearly highlighted when comparing MTs2s to the single-tasked s2s models. Depending on the metric used for assessment in 17-19 (out of 22) cases, the performance increases, often rather rapidly as shown in Table 2 and Figure 3, with the average relative gain in performance against the single-tasked s2s being 18%.

	LV	RF	FF	LSTM	s2s	MTs2s
BAT	.054	.075	.043	.045	.042	**.041**
BSV	.083	.215	.139	.083	.082	**.066**
BTC	.026	.034	.020	.020	**.019**	.020
DSH	.049	.128	.046	.043	**.039**	.039
EOS	.066	.067	.064	.056	.058	**.052**
ETC	.035	.657	.031	.095	.047	**.027**
ETH	.030	.425	.024	.029	.029	**.022**
ETP	.080	.069	.081	.077	.091	**.068**
IOT	.048	.068	.045	.038	**.037**	.038
LEO	.026	.449	.022	.143	.043	**.021**
LTC	.042	.042	.036	**.032**	.035	.033
NEO	.042	.065	.044	**.032**	.035	.033
OMG	.055	.151	.047	.045	.050	**.042**
TRX	.034	.245	.039	.075	.056	**.027**
UOS	.060	.303	.053	.346	.091	**.046**
VSY	.092	.099	.113	**.091**	.130	.092
XLM	.040	.047	.039	**.031**	.033	.033
XMR	.034	.052	.032	.036	.032	**.027**
XRP	.038	.303	.048	.113	.047	**.030**
XTZ	.067	.106	.053	.057	.052	**.052**
ZEC	.051	.050	.043	**.038**	.042	.039
ZRX	.048	.401	.041	.037	.038	**.037**
Med.	.048	.103	.043	.045	.043	**.037**

TABLE 1: Average MAPE per cryprocurrency across all future timesteps, for each model. Best results appear in bold.

As expected, the comparison between sequential vs. non-sequential models indicates that the former outperforms the latter in almost all metrics/cryptocurrencies. Nevertheless,

Crypto	Gain (%, MAPE)
TRX	52.4
LEO	50.9
UOS	49.7
ETC	42.6
XRP	35.1
VSY	29.0
ETP	24.8
ETH	21.8

TABLE 2: Most important gains in performance (%) were obtained by multi-tasking across different cryptocurrencies (MTs2s) in comparison with single-task sequence-to-sequence models (s2s).

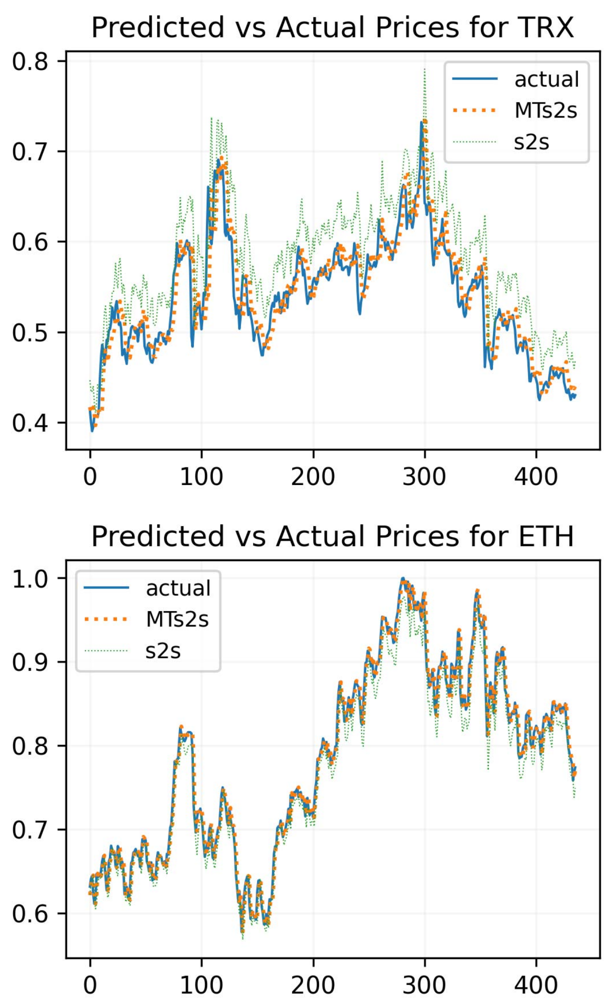


Figure 3: Comparison of predictions made by sequence-to-sequence (s2s) and Multitask sequence-to-sequence (MTs2s) models against the actual prices (a) for TRX, based on the third (last) forecasting timestep (corresponding MAPEs: .088/.035) and (b) for ETH, on the basis of the first forecasting timestep (corresponding MAPEs: .022/.014).

	LV	RF	FF	LSTM	s2s	MTs2s
BAT	.907	.870	**.940**	.935	.938	**.940**
BSV	.846	.434	.781	.876	.863	**.900**
BTC	.942	.890	**.964**	.963	**.964**	.962
DSH	.907	.501	.928	.931	**.939**	.937
EOS	.891	.868	.914	**.926**	.922	.925
ETC	.933	-11.179	.950	.653	.895	**.959**
ETH	.925	-9.654	.947	.928	.930	**.952**
ETP	.843	**.893**	.884	.883	.868	.892
IOT	.799	.681	.847	.866	**.872**	**.872**
LEO	.900	-15.898	.929	-1.129	.751	**.930**
LTC	.876	.882	.911	**.923**	.912	.920
NEO	.940	.884	.945	.960	.955	**.961**
OMG	.952	.380	.965	.963	.955	**.968**
TRX	.824	-3.572	.791	.498	.641	**.883**
UOS	.965	-0.714	.970	-2.234	.833	**.979**
VSY	.787	.784	.789	**.815**	.725	.808
XLM	.844	.787	.865	**.900**	.889	.882
XMR	.898	.815	.914	.896	.914	**.932**
XRP	.863	-3.517	.816	.144	.813	**.907**
XTZ	.900	.715	**.936**	.929	**.936**	**.936**
ZEC	.899	.876	.928	.935	.929	**.932**
ZRX	.827	-5.809	.870	**.887**	.880	**.887**
Med.	.899	.698	.921	.898	.903	**.931**

TABLE 3: Average r^2 metric per cryptocurrency and across all future timesteps for each model (LV, RF, FF, LSTM, s2s, MTs2s). Best, statistically significant results appear in bold font.

FF still achieves highly competitive performance in median terms against our more advanced LSTM/s2s models. This is attributed to the short temporal horizon we set in our experiments. I.e., the prediction of the next day (three 8-hour intervals) is an easier task than setting a wider prediction horizon (e.g., a week). Therefore the sequential predictions are not fully exploited. We plan to investigate this effect further in our future work. Finally, LV proves to be a rather competitive baseline, which is often ignored in related work. Indeed, on the average cryptocurrency ('Median' row on tables 1, 3 and 4), LV performs on par with LSTM, showcasing the challenging nature of the task. Setting up such naïve baselines is vital to offer insights into the effectiveness of models in future work.

Next, we further break down the performance of LSTM, s2s, and MTs2s on the timestep level. The charts shown in Figure 5 present the average MAPE and r^2 of the three models. As expected, the performance decreases as time goes by in our future horizon. However, MTs2s consistently outperforms the rest, whereas its performance on the third (i.e., most challenging) future timestep is still better than the performance of LSTM in the first (easier to predict) timestep, demonstrating the importance of (a) the sequential predictions and (b) multitasking. Finally, this performance improvement is further accompanied by much lower training times (see Figure 4) due to the ability of MTs2s to approximate the generative function of our time series much easier than the single-task models.

Limitations

Aiming further to exploit the performance of our best per-

	LV	RF	FF	LSTM	s2s	MTs2s
BAT	0.228	0.319	0.148	0.159	0.151	**0.146**
BSV	0.043	0.159	0.061	0.035	0.039	**0.028**
BTC	0.068	0.130	**0.043**	0.044	**0.043**	0.045
DSH	0.062	0.336	0.048	0.047	**0.041**	0.042
EOS	0.038	0.046	0.030	**0.026**	0.027	0.026
ETC	0.036	6.569	0.027	0.187	0.057	**0.022**
ETH	0.092	12.997	0.064	0.088	0.085	**0.058**
ETP	0.235	**0.161**	0.174	0.175	0.197	0.162
IOT	0.115	0.182	0.088	0.076	0.073	**0.073**
LEO	0.077	12.946	0.055	1.628	0.190	**0.054**
LTC	0.071	0.068	0.051	**0.044**	0.051	0.046
NEO	0.034	0.065	0.031	0.023	0.026	**0.022**
OMG	0.187	2.391	0.134	0.143	0.172	**0.123**
TRX	0.070	1.821	0.083	0.200	0.143	**0.047**
UOS	0.145	7.136	0.126	13.465	0.697	**0.087**
VSY	0.043	0.044	0.043	**0.038**	0.056	0.039
XLM	0.048	0.066	0.042	**0.031**	0.034	0.037
XMR	0.051	0.093	0.043	0.053	0.043	**0.034**
XRP	0.085	2.799	0.114	0.530	0.116	**0.057**
XTZ	0.275	0.785	0.175	0.196	0.177	**0.175**
ZEC	0.104	0.129	0.074	**0.067**	0.073	0.070
ZRX	0.072	2.837	0.054	**0.047**	0.050	0.047
Med.	0.072	0.251	0.058	0.072	0.065	**0.047**

TABLE 4: Average MSE per cryptocurrency across all future timesteps for each model (LV, RF, FF, LSTM, s2s, MTs2s). All results have been multiplied by 10^2 to ease readability. The best, statistically significant results appear in bold font.

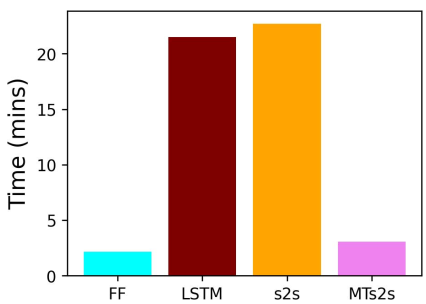
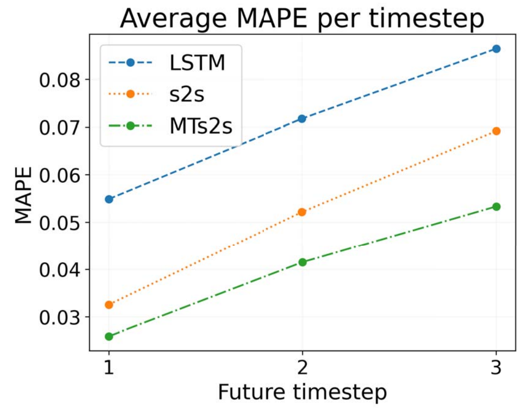


Figure 4: Comparison of training time needed (average, per cryptocurrency) for our deep learning models. Notice that the training time for the multitask MTs2s model is the *overall time* needed for training on all 22 cryptocurrencies. The experimental runs were performed on a 2.4 GHz Intel Core i5 CPU.

forming system and guide future work toward challenging directions, we performed a final experiment. Here, we increase the prediction horizon to one week (21 intervals) and contrast the performance (MAPE) of MTs2s against that of LV. The results are shown in Figure 6. Though MTs2s achieve better results than LV in the first timesteps (e.g., 2.6% vs. 5% in the first timestep), its accuracy degrades with time, and from the third day onwards, LV achieves better performance. Arguably, training with historical data from only the past four days might limit the capabilities of MTs2s to exploit long-range dependencies; however, this performance degradation over time also points to the need for incorporating online training in our future work.



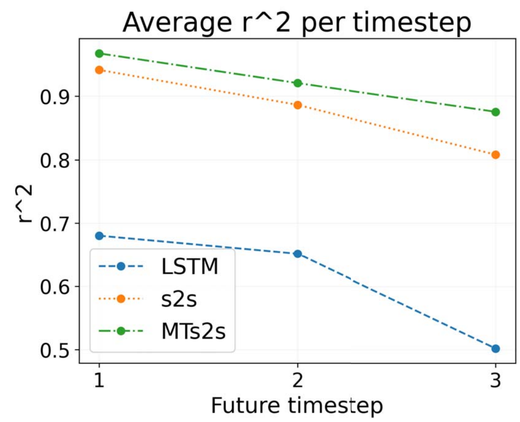


Figure 5: The evaluation scores (MAPE and r^2) of the three temporal models per timestep averaged across all cryptocurrencies.

5. Conclusions and Future work

This work explored and introduced sequence-to-sequence modeling of cryptocurrency fluctuations in time in a multi-task setting. We worked with univariate time series of multiple cryptocurrencies over two years and explored many evaluation metrics and baseline models. Our proposed model offers significant performance gains against common practices in related work and competitive baselines from the real world, highlighting the benefit of joint learning in a sequence-to-sequence manner while also offering essential improvements in training time.

In our future work, we plan to explore several other features that vary over time, such as trade volume, and incorporate external sources of information, such as news articles and textual data from social media, using natural language processing techniques and sentiment analysis. Finally, it would be of great interest to see how methodologies as the one developed in this paper can be used as price oracles for decentralized exchanges such as Automated Market Makers, where one of the main difficulties is the ability to query data external to a blockchain (*oracle problem*). Specifically, reliable forecasting models will release us from the need of

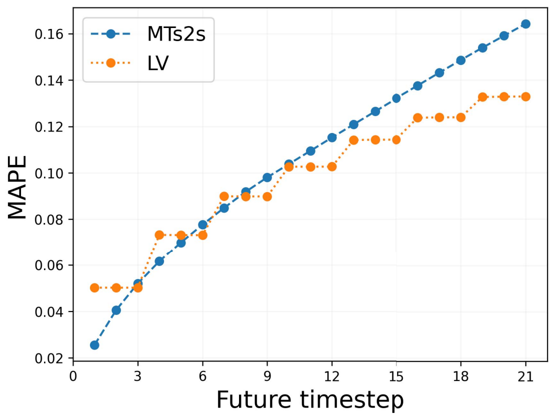


Figure 6: Comparison of (average across all cryptocurrencies) performance between MTs2s and LV per timestep, when working with a prediction horizon of seven days.

having constant access to oracles and, in the same time, will tackle one of their main issues of them regarding the track of the market price of assets.

References

[1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." Decentralized Business Review (2008): 21260.

[2] Voshmgir, Shermin. Token Economy: How the Web3 reinvents the Internet. Vol. 2. Token Kitchen, 2020.

[3] Werner, Sam M. and Perez, Daniel and Gudgeon, Lewis and Klages-Mundt, Ariah and Harz, Dominik and Knottenbelt, William J. SoK: Decentralized Finance (DeFi). ArXiv 2021.

[4] Buterin, Vitalik. "A next-generation smart contract and decentralized application platform." white paper 3.37 (2014).

[5] Wang, Qin and Li, Rujia and Wang, Qi and Chen, Shiping. Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges. ArXiv 2021.

[6] Raikwar, Mayank, Danilo Gligoroski, and Katina Kralevska. "SoK of used cryptography in blockchain." IEEE Access 7 (2019): 148550-148575.

[7] Y. Sovbetov. Factors Influencing Cryptocurrency Prices: Evidence From Bitcoin, Ethereum, Dash, Litcoin, and Monero. Accessed: 2018. [Online]. Available: https://mpra.ub.uni-muenchen.de/85036/

[8] H. S. Narman and A. D. Uulu, "Impacts of positive and negative comments of social media users to cryptocurrency," in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), Feb. 2020, pp. 187–192.

[9] Jesper Kristensen, Panos T, 0xZoro. The Composable Finance Mosaic Bridge and Liquidity Forecasting. AIBC Blockchain Conference 2022.

[10] Kristensen, J., Steffensen, P., Ottina, M. Automated Market Makers. Book to be published 2022.

[11] Xu, Jiahua, et al. "Sok: Decentralized exchanges (dex) with automated market maker (AMM) protocols." arXiv preprint arXiv:2103.12732 (2021).

[12] A. Joshi. Machine Learning for Predictive Analysis. Accessed: 2020. [Online]. Available: https://link.springer.com/book/10.100715-7106-0about

[13] S. Garg, "Autoregressive integrated moving average model based prediction of bitcoin close price," in Proc. Int. Conf. Smart Syst. Inventive Technol. (ICSSIT), Dec. 2018, pp. 473–478.

[14] I. M. Wirawan, T. Widiyaningtyas, and M. M. Hasan, "Short term prediction on bitcoin price using ARIMA method," in Proc. Int. Seminar Appl. Technol. Inf. Commun. (iSemantic), Sep. 2019, pp. 260–265.

[15] A. Barrett. Forecasting the Prices of Cryptocurrencies Using a Novel Parameter Optimization of Varima Models. Accessed: 2021. [Online].

[16] Krollner, Bjoern, Bruce J. Vanstone, and Gavin R. Finnie. "Financial time series forecasting with machine learning techniques: a survey." ESANN. 2010.

[17] M. M. Patel, S. Tanwar, R. Gupta, and N. Kumar, "A deep learning-based cryptocurrency price prediction scheme for financial institutions," J. Inf. Secur. Appl., vol. 55, Dec. 2020, Art. no. 102583.

[18] D. Vanderbilt. An Applied Study of RNN Models for Predicting Cryptocurrency Prices. Accessed: 2020.

[19] T. Awoke, "Bitcoin price prediction and analysis using deep learning models," in Communication Software and Networks. Singapore: Springer, 2021,

[20] Q. Guo, S. Lei, Q. Ye, and Z. Fang, "MRC-LSTM: A hybrid approach of multi-scale residual CNN and LSTM to predict bitcoin price," 2021

[21] W. Yiying and Z. Yeze, "Cryptocurrency price analysis with artificial intelligence," in Proc. 5th Int. Conf. Inf. Manage. (ICIM), Mar. 2019,pp. 97–101.

[22] F. Ferdiansyah, S. H. Othman, R. Zahilah Raja Md Radzi, D. Stiawan, Y. Sazaki and U. Ependi, "A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market," 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), 2019, pp. 206-210, doi: 10.1109/ICECOS47637.2019.8984499.

[23] S. McNally, J. Roche and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning," 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018, pp. 339-343, doi: 10.1109/PDP2018.2018.00060.

[24] M. Ortu, N. Uras, C. Conversano, G. Destefanis, and S. Bartolucci, "On technical trading and social media indicators in cryptocurrencies' price classification through deep learning," 2021, arXiv:2102.08189. [Online]. Available: http://arxiv.org/abs/2102.08189

[25] X. Huang, W. Zhang, Y. Huang, X. Tang, M. Zhang, J. Surbiryala, V. Iosifidis, Z. Liu, and J. Zhang, "LSTM based sentiment analysis for cryptocurrency prediction," 2021, arXiv:2103.14804. [Online]. Available: http://arxiv.org/abs/2103.14804

[26] S. Tanwar, N. P. Patel, S. N. Patel, J. R. Patel, G. Sharma and I. E. Davidson, "Deep Learning-Based Cryptocurrency Price Prediction Scheme With Inter-Dependent Relations," in IEEE Access, vol. 9, pp. 138633-138646, 2021, doi: 10.1109/ACCESS.2021.3117848.

[27] Hamayel, Mohammad J., and Amani Y. Owda. 2021. "A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms" AI 2, no. 4: 477-496. https://doi.org/10.3390/ai2040030

[28] Caruana, R. "Multitask learning." Machine learning 28.1 (1997): 41-75.

[29] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[30] Sutskever, I., Vinyals, O., Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, 27.

[31] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

[32] Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. ArXiv Preprint ArXiv:1412.6980.

[33] Luong, M.-T., Le, Q. V., Sutskever, I., Vinyals, O., Kaiser, L. (2015). Multi-task sequence to sequence learning. ArXiv Preprint ArXiv:1511.06114.

[34] Pasunuru, R., Bansal, M. (2017). Multi-Task Video Captioning with Video and Entailment Generation. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1273–1283.