A Data-driven Deep Learning Approach for Bitcoin Price Forecasting

Parth Daxesh Modi*, Kamyar Arshi† , Pertami J. Kunz‡, Abdelhak M. Zoubir§
‡*Grad. School of Comp. Eng.*, ‡§*Signal Processing Group*
Darmstadt University of Technology
*modiparth527@gmail.com, †kamyararshi@gmail.com,‡pertami.kunz@ieee.org, §zoubir@ieee.org

***Abstract*—Bitcoin as a cryptocurrency has been one of the most important digital coins and the first decentralized digital currency. We propose a shallow Bidirectional-LSTM (Bi-LSTM) model, fed with feature engineered data using our proposed method to forecast bitcoin closing prices in a daily time frame. We compare the performance with that of other forecasting methods, and show that with the help of the proposed feature engineering method, a shallow deep neural network out-performs other popular price forecasting models.**

***Index Terms*—Machine Learning, Deep Learning, Neural Networks, Feature Extraction, Cryptocurrency, Price prediction**

I. Introduction

Bitcoin is the first decentralized cryptocurrency that has become popular and widespread in the past years. It was introduced initially by an unknown identity under the pseudonym of Satoshi Nakamoto [13], and it was built without the need for any intermediate party in making transactions, thereby making it secure by verifying each transaction in a publicly distributed ledger called the blockchain [9]. Bitcoin's transactions run 24/7, and the currency is exchangeable in almost all cryptocurrency exchanges. Furthermore, Bitcoin allows traders and investors to benefit from better portfolio management [8]. Despite all the upsides, the price of bitcoin has experienced drastic rises and falls showing its high volatility and risk, hence bitcoin price prediction has always been an attractive topic among traders and the research community.

Thanks to the era of big data, deep learning algorithms have been showing their dominance in different fields such as logistics, computer vision, finance, and signal processing. There has been a lot of research in previous years using machine learning methods for crypto market forecasting, and deep learning methods play a big role in most of it [8]. One of the famous deep learning networks that are a state-of-the-art method in processing sequential data is the Long-Short-Term-Memory (LSTM) network [10], which is capable of finding long-term as well as short-term hidden dependency sequential structures in data such as natural language. Since the bitcoin price also follows a sequential structure, meaning the price of each time frame depends on previous prices in the order of time, LSTM networks can be exploited to predict bitcoin's price in a defined time proportion. There have been further studies around using time-series networks for cryptocurrency price forecasting such as [7], where Dutta *et al.* introduced a robust feature engineering with a simpler time-series network for prediction, or Wu *et al.* [16], where they have proposed two LSTM based models and compared the performance on the price prediction. In Jaquart *et al.* [11] various machine learning models are tested for price prediction in different time frames, ranging from one minute to 60 minutes, and it was concluded that recurrent neural networks and gradient-boosting classifiers are well-suited for such a task. In our proposed method, we have used a novel feature extraction and selection method, in which we use technical analysis indicators for the former and a Random Forest Regressor for the latter, to exploit the best possible features for bitcoin closing price forecasting in a daily time frame, and feed our features into a shallow Bi-LSTM network to not only decrease computational complexity but also having a promising performance.

Since deep learning models require a vast amount of data, one of the main challenges in bitcoin price prediction is that the available data is limited and none of the data augmentation tricks works. Therefore, we cannot simply use as many layers in our network as we want. As a result, we propose a method in that not only useful features are exploited with the help of feature engineering, but also our model is kept shallow and not computationally heavy.

Our main contribution is in the feature engineering and selection steps as well as the shallow architecture that completes the whole pipeline of Bitcoin price prediction. In Section II we elaborate the features used and the process of feature extraction and selection. We describe the various models we have exploited and compared them with our proposed method. Finally, we show our results and conclusion in Section III.

II. Methodologies

One of the most prominent figures used in price analysis in finance is the OCHL chart, which includes four prices for each defined time frame. **Open**, **Close**, **High**, and **Low** refer to, respectively, the opening price, closing price, highest price, and the lowest price of a transaction in the respective

*†These two authors contributed equally

The work of Pertami J. Kunz is supported by the Graduate School CE within the Centre for Computational Engineering at Technische Universität Darmstadt.

979-8-3503-3959-8/23/$31.00 ©2023 IEEE

time frame. We used the bitcoin's OCHL prices for each day from January 2013 until September 2021 while extracting and selecting some of the most important indicators for our task as our dataset for training and validation. We utilized InvestPy API [6] to scrap the historical bitcoin prices.

The raw transaction data show high correlations with one another. We aim to predict the closing price of the next day using this dataset. Using raw transactions may lead to overfitting of the machine learning (ML) models due to the aforementioned high correlation among the features. Therefore, we proposed a feature engineering method to extract and select the best features for training, with respect to our target task.

A. Proposed Feature Extraction and Selection

Other than collecting OCHL daily bitcoin transaction prices (4 features), we utilized Bitinfocharts[1] to extract 19 raw features: transactions in blockchain, average block size, sent by address, average mining difficulty, average hashrate, mining profitability, sent coins in usd, average transaction fees, median transaction fees, average block time, average transaction value, median transaction value, tweets, google trends, active addresses, top 100 to total percentage, average fee to reward, number of coins in circulation, and miner revenue.

For each of these (4+19) features, 3 windows (7 days, 30 days, 90 days) of 12 technical indicators were derived: the moving average (MA), weighted MA, Exponential MA, double exponential MA, triple exponential MA, standard deviation, variance, relative strength index, rate of change, upper and lower Bolliger bands [2], and MA convergence divergence.

In total, we derived 23×3×12=828 new features. Including the raw features, we fed in total 828+23=851 features to a robust scaler, that scales the data according to the interquartile range (IQR), to make the scales of all the features the same, and also that our ML models be less affected by outliers. Subsequently, we used a Random Forest (RF) Regressor [4] to evaluate the importance of each feature given our regression task, which is predicting the closing price of the next day. From the results, we only used the top 10 most important features ranked by the RF Regressor (Fig. 1).

B. Train/Test Split

As the bitcoin price is highly volatile, from 100 USD in 2013 to 63K USD in 2021, it is hard to train a model that generalizes well on such a huge dynamic range. Thus, we train multiple models by data splitting so as to include different time frames with different price ranges and thus various seasonality and trends in the sequence. Each training batch split consists of 500 data points, and the next 100 data points in the sequence are used as validation (testing) batch. This process is applied for all the available data points and is illustrated in Fig. 2.

Next we explain the methods of each model we exploited, followed by a description of our proposed model's building blocks.

[1]https://bitinfocharts.com/

C. Support Vector Regressor (SVR)

Support Vector Machines [3] are one of the most powerful supervised learning algorithms. They are versatile and able to perform nonlinear and linear classification and regression. SVR works in the same way as an SVM Classifier works, but instead of finding the hyperplane that maximizes the distance of the closest data points of two different classes, it tries to fit as many data points as possible on the hyperplane while limiting margin violations [5]. We have implemented this algorithm using the kernel Radial Basis Function (RBF), that has the benefit of being stationary and isotropic.

Another main reason of using SVR is that it works well on small datasets, and since it is indeed our case due to the splitting approach, as explained in the previous section.

D. LSTM

While using all the above models, the sequential relationship in the time series is not taken into consideration. The statistical models ARMA, and GARCH did so, but they lack in capturing the non-linearity in the time series. Furthermore, in a time-series dataset, both the long-term and short-term dependencies may be important. As a result, using simple RNN blocks might lead to gradient vanishing problems and will not consider long-term relations in the data. At this point, using Long-Short-Term-Memory neural networks will solve the aforementioned dilemma. The structure of one LSTM cell is shown in Fig. 3 and the output is calculated as follows [14]:

$$
\begin{aligned}
\tilde{c}^{<t>} &= \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \\
\Gamma_u &= \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u) \\
\Gamma_f &= \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f) \\
\Gamma_o &= \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o) \\
c^{<t>} &= \Gamma_u \odot \tilde{c^{<t>}} + \Gamma_f \odot c^{<t-1>} \\
a^{<t>} &= \Gamma_o \odot \tanh(c^{<t>}),
\end{aligned} \tag{1}
$$

where $\tilde{c}$ is the cell input activation vector, Γ_u, Γ_f, and Γ_o are the update, forget, and output gates activation vectors, respectively, c and a are the cell the hidden state vectors, σ is the sigmoid function, W and b refer to the weight matrices and bias vector parameters, $\odot$ sign is element-wise multiplication, and $< t >$ means at time step t. The architecture of the LSTM neural network we have used is exactly the same as our proposed Bi-LSTM model, and instead of the Bi-LSTM cells we have LSTM cells.

E. Bidirectional-LSTM

LSTM neural networks are fed with a sequence in the dataset in order from the beginning of the series at time 0 until the end of the sequence. However, sometimes there are hidden relations in a sequence when looking at it from the other way, meaning in the reverse descending order. To exploit this, we can use Bi-LSTM networks [15], which not only do the same thing as LSTM does, but also they take input from the last element in a sequence and continue going back to the start of

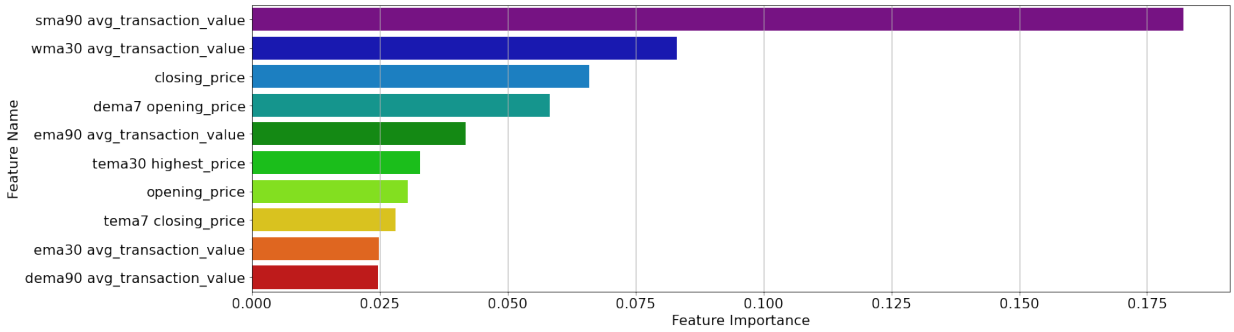

Fig. 1: Top 10 most important features after feature extraction using the technical indicators and ranking them using a Random Forest Regressor, where ema=Exponential Moving Average, wma=Weighted Exponential Moving Average, dema=Double Exponential Moving Average, tema=Triple Exponential Moving Average, avg=Average, and the numbers (7, 30, 90) refer to the window sizes.

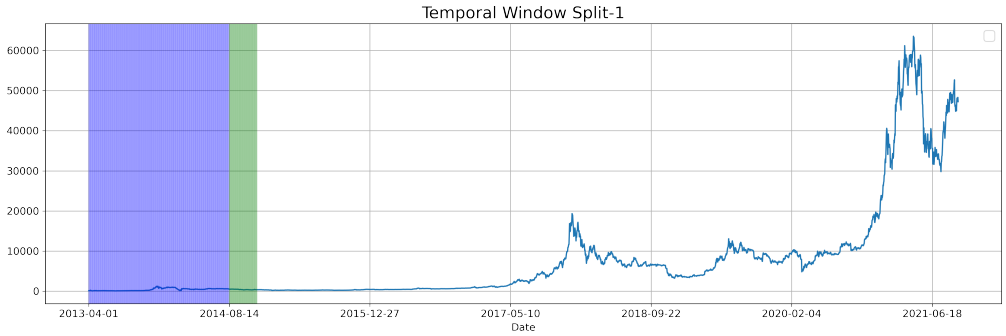

Fig. 2: Train set and Test set splitting in the first batch. The blue box is the first training batch and the green box is the first test batch. This is done on sequentially on the dataset shown in this figure.

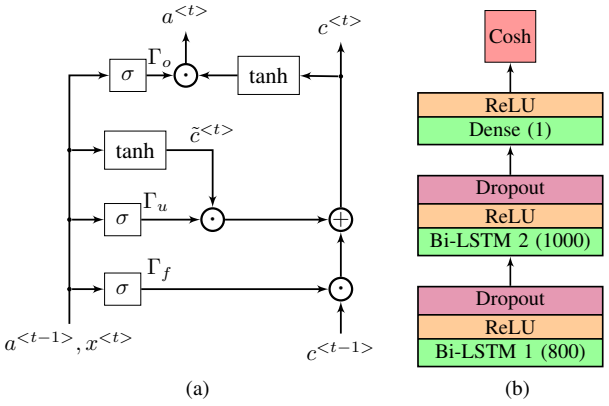

Fig. 3: (a) The individual LSTM cell and (b) the proposed Bi-LSTM neural network. Cosh is the Cosine Hyperbolic loss.

it. This makes the neural network capable of finding hidden sequential relations in both ways.

F. Proposed architecture

Our proposed architecture consists of three layers. The first and the second layer are Bi-LSTM cells and each is

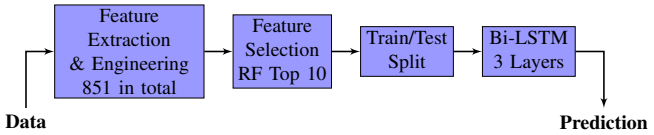

Fig. 4: Whole proposed pipeline. The final block, Bi-LSTM 3 Layers, contains the cells illustrated in Fig. 3(b).

followed by dropout layers in training to avoid overfitting. At the end, we have a single neuron fully-connected layer to output the prediction. Each layer's output goes through a ReLU [1] activation function since the price cannot be a negative number and to avoid the vanishing gradient problem. We used a hyperbolic cosine loss as our loss function due to two main reasons: (a) It behaves stable during the gradient descent search and (b) is also not affected by sudden disparate predictions [12]. The architecture of the proposed Bi-LSTM structure is shown in Fig. 3, and the whole proposed pipeline is illustrated in Fig. 4.

III. Results and Conclusion

Each model is trained on the transformed dataset with the selected features, and divided into training and testing batches,

as discussed in Section II. The training is performed to predict the next closing price of bitcoin. As shown in Table I, the mean value of the three error types, RMSE, MAE, and MAPE, is measured for the predictions on both the training sets and test sets. Since there might be outliers while taking the mean, we also have provided the median performance metric values of each training and test batch in Table I. The results of the ML models of Section II are presented in the Table I along with the results of a linear regression (LR) model to compare each model performance with a baseline.

We can observe in Table I that our proposed Bi-LSTM is performing more consistently and better compared to other models. Furthermore, it is clearly presented in the table that the performance of the proposed model on the test batches has the fewest outliers, since the median and the mean MAPE are the same, 3.16%. Please note that we have not included the comparison with the ARMA, ARIMA and the GARCH models since they are trained on the entire sequence of data set (without splitting) and thus it is not a fair comparison to describe.

As a summary, this study has proposed a data-driven approach for predicting Bitcoin's closing price, using various methods of feature extraction, selection, and data splitting, alongside a proposed Bi-LSTM neural network architecture to tackle the high volatility and time series dependencies in bitcoin price. We have also compared and explained various time-series and ML methods with their pros and cons and clarified the reason of using neural networks and in specific Bi-LSTM networks. Eventually, we have compared the results and showed that the proposed shallow Bi-LSTM architecture performs the best and the most consistently on average. Other than being computationally optimized, this forecasting model may aid traders working with cryptocurrency, especially since the crypto market is 24/7, and with the high volatility bitcoin price has, it could be a good metric for AI-assisted trading for professional traders.

TABLE I: Mean and Median Performance Comparison

Methods	**Mean of Metrics**					
	RMSE		***MAE***		***MAPE***	
	train	test	train	test	train	test
LR	378.9091	674.032	246.3711	546.109	0.1945	0.22664
SVR	380.7813	898.2263	239.8385	738.6972	0.1370	0.1850
LSTM	**262.8562**	455.5994	**149.1471**	377.3157	**0.0297**	0.0337
Proposed	268.3314	**450.3816**	152.8135	**334.6625**	0.0312	**0.0316**
	Median of Metrics					
LR	298.8742	373.9383	216.6750	312.1933	0.0554	.0687
SVR	299.1291	403.4483	201.6113	340.4691	0.05493	0.0856
LSTM	**211.8146**	215.4055	**122.4450**	154.995	**0.0258**	**0.03073**
Proposed[1]	215.9530	**197.4914**	125.0576	**135.7671**	0.02647	0.0316

[1]The Proposed shallow Bi-LSTM model has the same MAPE for the mean and median on the test set

ACKNOWLEDGMENT

We thank Abhishek Deshmukh and Ekican Cetin for their involvement in the earlier stage of this project.

REFERENCES

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.

[2] John Bollinger. Using bollinger bands. *Stocks & Commodities*, 10(2):47–51, 1992.

[3] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[4] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[5] Stella M. Clarke, Jan H. Griebsch, and Timothy W. Simpson. Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses. *Journal of Mechanical Design*, 127(6):1077–1087, 08 2004.

[6] Alvaro Bartolome del Canto. investpy - financial data extraction from investing.com with python. https://github.com/alvarobartt/investpy, 2018-2021.

[7] Aniruddha Dutta, Saket Kumar, and Meheli Basu. A gated recurrent unit approach to bitcoin price prediction. *Journal of risk and financial management*, 13(2):23, 2020.

[8] Qiutong Guo, Shun Lei, Qing Ye, and Zhiyang Fang. Mrc-lstm: A hybrid approach of multi-scale residual cnn and lstm to predict bitcoin price. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[9] Tian Guo, Albert Bifet, and Nino Antulov-Fantulin. Bitcoin volatility forecasting with a glimpse into buy and sell orders. In *2018 IEEE international conference on data mining (ICDM)*, pages 989–994. IEEE, 2018.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Patrick Jaquart, David Dann, and Christof Weinhardt. Short-term bitcoin market prediction via machine learning. *The journal of finance and data science*, 7:45–66, 2021.

[12] Thilo Moshagen, Nihal Acharya Adde, and Ajay Navilarekal Rajgopal. Finding hidden-feature depending laws inside a data set and classifying it using neural network, 2021.

[13] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[14] Andrew Ng. *Machine Learning Yearning*. Online Draft, 2017.

[15] Mike Schuster and Kuldip Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681, 12 1997.

[16] Chih-Hung Wu, Chih-Chiang Lu, Yu-Feng Ma, and Ruei-Shan Lu. A new forecasting framework for bitcoin price with lstm. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 168–175, 2018.