IEEE ICBC'24 1570981883

Integrating Behavioral Finance Factors with Temporal Convolutional Networks for Enhanced Cryptocurrency Return Predictions

***Abstract*—This study empirically integrates behavioral finance theories with technical analysis to predict cryptocurrency returns. Utilizing Temporal Convolutional Networks (TCN), the study aims to improve the accuracy of these predictions. In a methodologically focused approach, the research selects Ethereum and utilizes granular, one-minute interval data sourced from the Binance exchange. The empirical findings underscore the superiority of the TCN model over conventional models, highlighting its greater precision and computational efficiency. This approach underlines the significance of incorporating behavioral insights into technical analysis, offering a more sophisticated understanding of the influence of behavioral finance factors on market efficiency.**

***Keywords*—Behavioral finance, Cryptocurrency, Predictions, Investor behavior, Technical analysis indicators**

I. Introduction

This research embarks on a critical examination of the influence of investor irrational behavior on the cryptocurrency market, a domain that notably deviates from the principles of weak-form market efficiency. We contend that this deviation provides a compelling case for the feasibility of market prediction through technical analysis. Central to our study is the exploration of the entire predictive process in cryptocurrency markets, with a specific emphasis on demystifying the role and selection of technical indicators within this context. The use of Temporal Convolutional Networks (TCN) is not merely a technological choice but is supported by robust theoretical considerations. We seek to elucidate the reasons behind the selection of specific technical indicators, ensuring that these choices are not arbitrary but are grounded in behavioral finance theories. This approach aims to blend analytical techniques with a theoretical understanding, thereby offering a nuanced and informed framework for predicting cryptocurrency market movements.

II. Literature Review

A. Key Theories and Research

This research is grounded in the Efficient Market Hypothesis (EMH) as conceptualized by Eugene Fama [1], particularly its application to the unique realm of cryptocurrency markets. We begin by examining the extent to which cryptocurrencies conform to or diverge from the principles of EMH, especially its weak-form efficiency. Notable in this context is a previous study that used a quantum harmonic oscillator (QHO) model to assess Bitcoin's market efficiency suggesting that cryptocurrency close to being weakly efficient [2].

Despite this indication of weak-form efficiency, several studies highlight behavioral anomalies that challenge the efficiency of cryptocurrency markets. Works such as Elie Bouri's research on imitation in investment decisions [3], Oliver Borgards et al.'s analysis of market overreactions [4], and Grobys' investigation into lottery-like investor behavior exemplify this trend [5]. Additionally, Zhilu Lin's exploration of herding and momentum phenomena, particularly in the context of social media and news influence [6], adds further depth to this discourse.

Crucially, studies like Kappal and Rastogi (2020) delve into the interplay between behavioral biases—such as the disposition effect, herding effect, and overconfidence bias—and market inefficiencies [7][11]. These works collectively underscore the significant impact of psychological factors on rational decision-making and, by extension, on the efficiency of investment decisions.

Our research acknowledges the dichotomy that exists within cryptocurrency markets: they display characteristics of weak-form efficiency, yet simultaneously, they are susceptible to inefficiencies brought about by behavioral biases. This paradox forms the basis of our study, which aims to empirically integrate behavioral finance theories into the analysis of cryptocurrency markets, thereby enhancing the understanding of their dynamics and predictive potential.

B. Model Design

We focused on the Temporal Convolutional Network (TCN) model, chosen for its advantages in handling time series momentum sequences. The TCN component employs dilated convolutions to extend its receptive field, enabling the model to process information from various time scales effectively. This feature is particularly beneficial for analyzing the volatile and complex patterns characteristic of cryptocurrency markets. TCN is able to model longer term dependencies in time series compared to traditional Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP), and the computational time and cost will be significantly reduced compared to models such as Transformer. The research incorporates behavioral finance factors into the TCN model, aiming to improve prediction accuracy by considering investor sentiment, herding behavior, and other psychological influences on cryptocurrency markets.

III. Methodology

The proposed model in this paper integrates a TCN with a fully connected layer, offering a robust architecture for time series analysis.

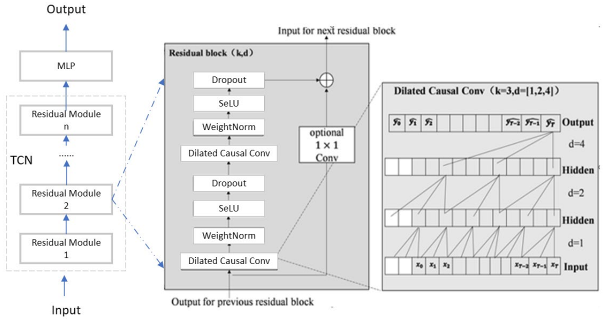


Fig. 1. Full model structure

In Fig. 1, the TCN architecture is depicted as a series of stacked residual blocks, each designed to process sequential data with multiple features. Unlike the traditional single-feature TCN, our model accommodates a multi-dimensional feature space. Each residual block is composed of multiple layers of dilated causal convolutions, which are crucial for ensuring that the model's predictions are based solely on historical data. These convolutional layers are augmented with the non-linear SeLU activation function, providing the necessary non-linearity to capture complex patterns. Regularization is introduced through dropout and weight normalization techniques to prevent overfitting. The output of each block feeds into the next and combines with its input in a residual connection to improve training efficiency and mitigate the vanishing gradient problem. Following the TCN's feature extraction process, the MLP comes into play. The fully connected layers of the MLP are tasked with analyzing these refined features. At this stage the model synthesizes the extracted features. The MLP's role is crucial in translating the intricate patterns identified by the TCN into final output predictions. By combining the detailed feature extraction of the TCN with the analytical prowess of the MLP, the model achieves a heightened capability to forecast cryptocurrency returns with greater accuracy.

This sequential process of feature refinement by the TCN and subsequent analysis by the MLP underlines the collaborative strength of the model. Each component plays a vital role, with the TCN providing a deep understanding of time-series dynamics and the MLP bringing its analytical acumen to bear on these insights, culminating in precise and reliable predictive outcomes. The following sections go into detail about the full structure of TCN and the meaning of the terms.

A. 1-D Convolutional Networks

TCN employs the structure of a 1-D Fully Convolutional Network (FCN), where the input and output time lengths of each hidden layer remain the same, maintaining consistent time steps. Specifically, for the first hidden layer, regardless of the kernel size and dilation, if the input has "n" time steps, the output also has "n" time steps. Similarly, for the second hidden layer, the third hidden layer, and so on, the input-output time step lengths are all "n". This aspect resembles RNNs, where every time step's input has a corresponding output at any layer.

For the initial time step, which lacks historical information, TCN assumes that its historical data is entirely composed of zeros (essentially due to the padding in convolution operations). Additionally, previous studies discovered that TCN exhibits a stronger ability to retain long-term historical information compared to LSTM [8][9].

B. Dilated Causal Convolutions

TCN harnesses the powerful characteristics of convolution directly to extract features across time steps. This is achieved through dilated causal convolutions, which can be broken down into three components: dilation, causality, and convolution.

- “Convolutions” refers to the convolution operation used in CNNs, which involves the sliding operation of a convolutional kernel over the data.
- "Dilation" refers to the allowance for gapped sampling of the input during convolution. As “Fig. 2” shares similarities with the "stride" used in Convolutional Neural Networks (CNNs), but also exhibits distinct differences. we adopted the dilated convolutions in our model instead, which is defined as：

$$F(x) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d\times i} \tag{1}$$

where x ∈ R T is a 1-D series input, and f : {0, · · · , k − 1} → N is a filter of size k, d is called the dilation rate, and (s − d × i) accounts for the direction of the past. In a dilated convolutional layer, filters are not convoluted with the inputs in a simple sequential manner, but instead skipping a fixed number (d) of inputs in between.

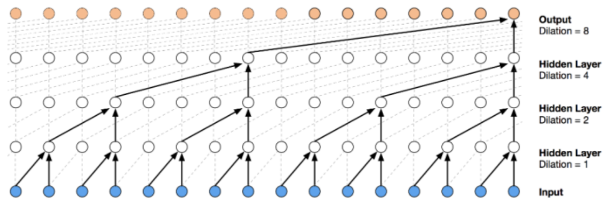
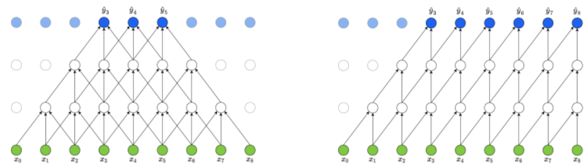


Fig. 3. Dilated convolution

- Causality refers to the fact that in the "i-th" layer, the data at time "t" depends only on the influence of the values at time "t" and earlier in the "(i-1)-th" layer. Causal convolution ensures that during training, future data is disregarded, making it a strictly time-constrained model.



(a) standard (non-causal) (b) causal

Fig. 4. Standard convolution and Causal convolution

C. Residual Module

Moreover, even when employing dilated causal convolutions, the model might still be quite deep. Deeper network structures can sometimes trigger issues like gradient vanishing or gradient explosion. To address this situation, TCN adopts a structure akin to the residual blocks in ResNet, allowing for the replacement of convolutional layers. This enables the training of even deeper networks.

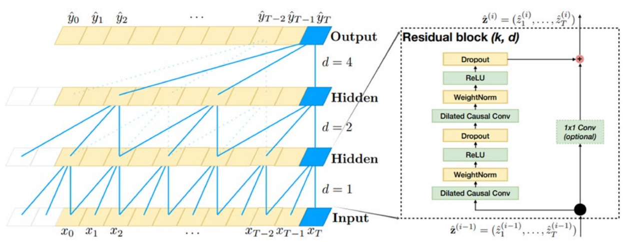


Fig. 2. Residual module structure

Within TCN's residual modules, there are two layers of dilated causal convolutions alongside SeLU non-linear functions. Additionally, the weights of the convolutional kernels undergo weight normalization (Weight-Norm). Furthermore, TCN introduces dropout after each dilated convolution within the residual module for regularization, as illustrated in the “Fig. 4”. This design of the TCN structure enhances its generalizability.

As shown in the “Fig. 4”, there are two layers of dilated convolutions, two layers of weight normalization, two layers of ReLU activation functions, and two layers of Dropout. Unlike stander-ResNet, in which the input is directly added to the output, in TCN, due to the potential mismatch in input and output dimensions, a 1x1 convolution is introduced to modify the input feature dimensions before adding it to the output. The core idea of ResNet is the usage of shortcut connection which skips one or more layers and directly connects to later layers (which is the so-called identity mapping), in addition to the standard layer stacking connection F. A residual block consists of the abovementioned two branches, and its output is then g(F(x) + x), where x denotes the input to the residual block, and g is the activation function.

1) Weight-normalization

Weight Normalization is a neural network optimization technique used to expedite the training process of neural networks and enhance the convergence speed and stability of models.

$$w = \frac{g}{||v||} v \tag{2}$$

"W" represents the weights linked to neurons, "g" is a scalar, with its magnitude being equal to the magnitude of "w". $\frac{v}{||v||}$ denotes a unit vector in the same direction as "w". Therefore, we can optimize the loss function "L" by performing scaling of $\frac{g}{||v||}$ and projecting it onto M_w.

$$\nabla_v L = \frac{g}{||v||} * M_w * \nabla_w L \tag{3}$$

Where $\nabla_v L$ represents the gradient of the loss function "L" with respect to "v", $M_w = I - \frac{w * w^T}{||v||^2}$, and the dot product with vectors can project any vector onto the orthogonal complement of "w".

2) Activation Function

The SeLU activation function (Scaled Exponential Linear Unit) originated from Neural Information Processing systems at 8 June 2017 [10].

$$SeLU(x) = \lambda \begin{cases} x & if\ x > 0 \\ \alpha(e^x - 1) & if\ x < 0 \end{cases} \tag{5}$$

Where α=1.6732..., λ=1.0507..., SELU can perform subunit normalization on neural networks, resulting in an output mean of 0 and a standard deviation of 1. Internal normalization is faster than external normalization, indicating that the network can converge more swiftly. Issues of gradient explosion or vanishing become implausible.

3) Dropout

Dropout is a neural network training technique that reduces overfitting by randomly deactivating a portion of hidden layer nodes in each batch. It disrupts interdependence among feature detectors, lessening reliance on others. Using a random variable "r" with a probability "p" in a Bernoulli distribution, this method sets some neuron values to 0 during training, promoting generalization. In this paper, with a 0.25 dropout rate, around 25% of neuron values are set to 0..

$$\begin{cases} r_j^{(l)} \sim \mathrm{Bernoulli}(p) \\ \hat{y}^{(l)} = r^{(l)} * y^{(l)} \\ z_i^{(l+1)} = w_i^{(l+1)} y^l + b_i^{(l+1)} \\ y_i^{(l+1)} = f\left(z_i^{(l+1)}\right) \end{cases} \tag{6}$$

IV. Experimental Testing and Model Evaluation

A. Experimental Method

We chose the highly representative cryptocurrency, Ethereum (ETH), as the subject of our study, with data sourced from the Binance exchange. For our experiments, we used one-minute data from Ethereum spanning from October 29th, 2022, at 00:42:00 to November 15th, 2022, at 23:59:00. as the subject of our study, comprising 200,000 consecutive minutes.

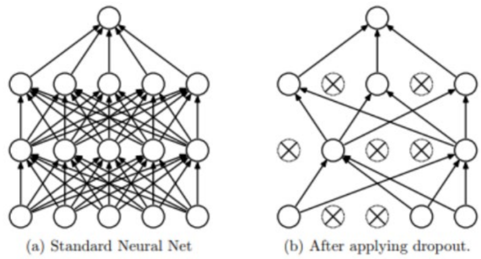


Fig. 5. Standard neural net VS Dropout neural net

The core of this study involves the utilization of technical indicators with one-minute interval data to forecast the future thirty-minute returns. The employed technical indicators, such as RSI, moving averages, volatility, skewness, and kurtosis as input, are all automatically derived from metrics like opening, closing, high, low, and trading volume.

Initially, we performed a temporal training-test split on each dataset. The first 75% of the data was allocated to the training set, while the remaining 25% was designated as the test set. To capture features from the sequences, we employed 16 filters with a kernel size of 2. We used dilations of [1, 2, 4, 8] to capture temporal dependencies effectively, after that we employ a MLP comprising four connected layers, with the neuron counts in these layers being 8, 16, 16, and 8, respectively. These layers are specifically designed to analyze the refined features processed by the TCN. Subsequently, we divided the training set into multiple batches, each containing 128 samples. These batches were fed into our model, undergoing a total of 75 training epochs to minimize the loss function.

B. Evaluation Method

To assess the quality of a model’s predictions, we computed the R-squared score ($R2$) as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_{t_i} - \hat{y}_{t_i}\right)^2}{\sum_{i=1}^{n} \left(y_{t_i} - \bar{y}\right)^2} \tag{7}$$

Where y_{t_i} is the true returns of the crypto after the normalization, $\hat{y}_{t_i}$ is the predicted value, $\bar{y}$ is the average of the predicted values, and n indicates their number, It highlights the model’s variance in relation to the total variance, the higher the R-squared value, the better the model's performance.

V. Results and Discussion

TABLE I. Report of TCN

Model	Value					
	Epoch	***CT***	***loss***	***R2***	***Val-loss***	***Val-R2***
TCN	75	45	0.0092	0.9537	0.0077	0.9510

[a] CT: computational time (min)

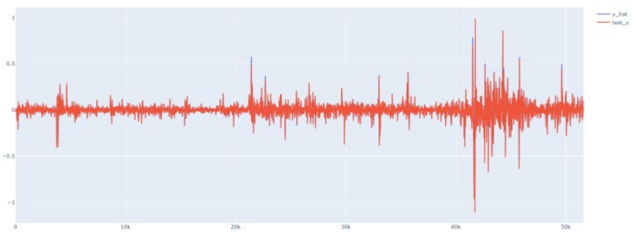


Fig. 6. Cross real-returns and prediction-returns

The result indicates that the TCN model has shown remarkable performance in predicting cryptocurrency price. The key metrics of loss value and R-squared value provide insights into how well the model is capturing the underlying patterns and relationships in the data. The low loss value of 0.0077 is a positive indicator. This means that the model is effectively minimizing the differences between its predictions and the true cryptocurrency returns; The high R-squared value of 0.9510 is equally notable.

This paper validates the efficacy of the Temporal Convolutional Network-Multi-Layer Perceptron (TCN-MLP) model in predicting 30-minute returns for Ethereum. Our approach not only aligns with the theoretical framework of predictive modeling but also integrates significant insights from the field of behavioral finance. The study underscores that investor behavior, characterized by biases such as herd mentality and the endowment effect, significantly deviates from the weak-form efficiency assumption of the Efficient Market Hypothesis. This deviation leads us to posit that technical analysis could possess predictive capabilities in the cryptocurrency market. Hence, we have incorporated a suite of technical analysis indicators into the TCN-MLP model. These indicators, including moving averages, volatility measures, and autocorrelation coefficients, were carefully selected to enhance the model's ability to accurately predict price trends.

The choice of specific indicators, such as the Exponential Moving Average (EMA) for capturing trend behaviors and the Relative Strength Index (RSI) for identifying mean-reversion characteristics, was strategic and based on their correlation with observed price behaviors in the cryptocurrency market. Additionally, trading volume was included as an indicator due to its proven correlation with price anomalies.

The robust performance of the TCN model, marked by its low loss, low training costs, and high R-squared values, confirms its effectiveness in capturing complex trends in the cryptocurrency market. However, the inherent volatility and complexity of these markets, influenced by factors beyond the predictive models, necessitate a cautious approach. While the TCN model is a powerful forecasting tool, it should be utilized as part of a diversified strategy, complemented with broader market analysis to mitigate the risks associated with relying on a single model.

In summary, this paper not only deepens the understanding of cryptocurrency market dynamics from a behavioral finance perspective but also showcases a novel application of combining behavioral finance theories with technical analysis.

References

[1] Fama, E.F. (1970). Efficient Capital Markets: A Review Of Theory And Empirical Work*. Journal Of Finance, 25, 383-417.

[2] Yi, E., Yang, B., Jeong, M., Sohn, S., & Ahn, K. (2023). Market efficiency of cryptocurrency: evidence from the Bitcoin market. Scientific Reports, 13.

[3] Bouri, E., Gupta, R., & Roubaud, D. (2019). Herding Behaviour In Cryptocurrencies. Finance Research Letters.

[4] Borgards, o., & czudaj, r.l. (2021). Features of overreactions in the cryptocurrency market. The quarterly review of economics and finance, 80, 31-48.

[5] Grobys, K., & Junttila, J. (2020). Speculation And Lottery-Like Demand In Cryptocurrency Markets. Ern: Speculation In Economic Markets (Topic).

[6] Lin, Z., Wu, W., & Zhang, H. (2021). Momentum, Information, And Herding. Journal Of Behavioral Finance, 24, 219 - 237.

[7] Kappal, J.M., & Rastogi, S. (2020). Investment behaviour of women entrepreneurs. Qualitative Research in Financial Markets, 12, 485-504.

[8] S. Gopali, F. Abri, S. Siami-Namini and A. S. Namin, "A Comparison of TCN and LSTM Models in Detecting Anomalies in Time Series Data," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 2415-2420, doi: 10.1109/BigData52589.2021.9671488.

[9] Bai, Shaojie, J. Zico Kolter and Vladlen Koltun. "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling." ArXiv abs/1803.01271 (2018): n. pag.

[10] Klambauer, Günter, Thomas Unterthiner, Andreas Mayr and Sepp Hochreiter. "Self-Normalizing Neural Networks." ArXiv abs/1706.02515 (2017): n. pag.

[11] Kristoufek, Ladislav. 'What Are The Main Drivers Of The Bitcoin Price? Evidence From Wavelet Coherence Analysis'. Plos One, Edited By Enrico Scalas, Vol. 10, No. 4, Apr. 2015, P. E0123923. Doi.Org (Crossref)