Time-Series Transformer for Predicting Bitcoin Price

Longchen Zheng *

Jinan Xicheng Experimental High School, Jinan, 250118, China

* Corresponding Author Email: olongchenzheng@outlook.com

Abstract. Bitcoin's high volatility poses significant challenges for short-term price prediction, making it a critical area of study for financial forecasting. Traditional models such as Long Short-Term Memory (LSTM) networks often encounter difficulties in handling long-range dependencies and non-stationary data, limiting their predictive accuracy under volatile conditions. This study introduces the Time-Series Transformer (TST) as a novel approach to predict Bitcoin's short-term prices. By leveraging self-attention mechanisms, TST effectively captures complex temporal patterns in historical Bitcoin data, including prices and trading volume. The data was segmented into fixed-length windows to facilitate model training and testing. Evaluation metrics such as Mean Squared Error (MSE), Mean Absolute Scaled Error (MASE), and R-squared (R^2) demonstrated TST's superior performance over LSTM, particularly during periods of high market fluctuation. Furthermore, TST exhibited notable computational efficiency when working with large datasets, underscoring its scalability. These findings not only highlight TST's potential for enhancing cryptocurrency price prediction but also pave the way for future research integrating external data sources and exploring further model enhancements for more robust financial forecasting.

Keywords: Time-Series Transformer, Bitcoin, Cryptocurrency.

1. Introduction

Bitcoin has attracted a great deal of attention in recent years in several ways. Bitcoin payments are based on a new and interesting technological solution and function differently than traditional payments. In some payment situations, Bitcoin can bring lower costs, speed, the anonymity of traditional payment methods, and more [1]. Traditional statistical models, such as ARIMA, and machine learning techniques, including Long Short-Term Memory (LSTM) networks, have been widely applied to time-series forecasting. However, these methods often fall short in capturing complex temporal dependencies and adapting to non-stationary data inherent in cryptocurrency markets [2]. The emergence of Transformer-based architectures, initially developed for natural language processing tasks, has opened new avenues for time-series forecasting. The Time-Series Transformer (TST) leverages self-attention mechanisms to model long-range dependencies and complex temporal patterns efficiently. Unlike recurrent models, TST processes all time steps simultaneously, enabling it to handle large datasets with high computational efficiency [3]. This makes it particularly well-suited for financial data characterized by high dimensionality and frequent fluctuations.

This study focuses on evaluating the performance of the TST model in predicting Bitcoin's short-term price movements. By examining key metrics such as Mean Squared Error (MSE), Mean Absolute Scaled Error (MASE), and R-squared (R^2), this research aims to determine whether the TST model offers superior predictive accuracy and stability under volatile market conditions.

The findings of this research not only contribute to the growing body of literature on deep learning applications in financial forecasting but also highlight the potential of Transformer-based architectures in addressing the unique challenges of cryptocurrency markets. Additionally, this study lays the groundwork for future exploration into integrating external data sources and developing advanced enhancements to further improve prediction accuracy and applicability.

2. Related Works

2.1. Overview of Transformer and LSTM Models

Transformer: Transformers rely on a self-attention mechanism that evaluates the relationships between all elements in a sequence simultaneously. The key components include query Q, key K, value matrices V, the dimension of the key vector which is used for scaling to avoid excessive dot product values, and SoftMax used to normalize attention score. Calculate attention scores as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Additionally, to capture different subspace information, the multi-head attention mechanism parallel computing multiple attention heads:

$$Multihead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W^O \tag{2}$$

With each head:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

W_i^Q, W_i^K, W_i^V are linear transformation matrices for different attention heads, W^O is output linear transformation matrix.

In each Transformer layer, the output after multiple attentions is nonlinearly transformed through a fully connected network:

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2 \tag{4}$$

W_1, W_2 are weight matrix, $b_1$and b_2 are bias term.

LSTM: LSTM is a type of Recurrent Neural Network (RNN) designed to overcome the vanishing gradient problem often encountered in standard RNNs. It incorporates three gates—input, forget, and output—that regulate the flow of information:

Forget Gate which determines how much past memory the current time step retains:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \tag{5}$$

h_{t-1} is the hidden state from the previous time step.

x_t is the input at the current time step.

W_f, b_f are forget gate weights and bias.

σ is the sigmoid activation function.

Input Gate which determines the degree to which new information is written in the current time step:

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{6}$$

Output Gate determines the hidden state output for the current time step:

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o \tag{7}$$

$$h_t = o_t * tanh\ (c_t) \tag{8}$$

2.2. Characteristics and Advantages of Transformer Models

Transformers offer several key advantages over traditional models like LSTM. First, their parallelization capability allows them to handle all time steps simultaneously, significantly reducing training time compared to LSTM, which processes data sequentially. Second, the self-attention mechanism enables Transformers to scale effectively with large datasets, making them particularly suitable for high-frequency financial data [4]. Third, Transformers excel at modeling relationships between distant time steps, addressing a critical limitation of LSTM by leveraging attention scores to capture long-range dependencies. Finally, Transformers' flexibility allows for seamless integration

of additional features, such as trading volume or external sentiment data, enhancing their predictive capabilities.

These strengths position Transformers as a powerful tool for tackling the unique challenges of cryptocurrency price prediction, where data complexity and market dynamics demand robust and adaptable models [5].

3. Methodology

3.1. Data Preprocessing

The dataset utilized in this study originates from the Kaggle repository (https://www.kaggle.com/aipeli/btcusdt) and comprises one year of minute-level trading data for the BTC-USDT pair. This dataset contains 35 columns, including a timestamp and 34 features derived from price and volume data. The core features include open price, high price, low price, close price, and trading volume. Additionally, 29 technical indicators were calculated using the Finta library, encompassing metrics such as TRIX, VWAP, MACD, ROC, MOM, RSI, and more. These features collectively provide a comprehensive representation of Bitcoin's trading behavior.

To ensure data quality, all rows preceding the last missing value in any feature sequence (approximately 131 rows) were removed. This preprocessing step guarantees a complete dataset devoid of missing values, facilitating accurate model training and evaluation.

Use the plot_correlation function to plot correlation matrix (see Figure 1) to explore the relationships between all 34 features, including the newly calculated indicators. The correlation matrix was plotted with minute-level granularity, offering insights into inter-feature dependencies. Based on the correlation analysis, the features were reranked by importance that optimizes model performance.

Bayesian hyperparameter optimization, conducted using the Optuna framework, identified min-max normalization as the optimal scaling technique. This method scales each feature to a range of [0,1], enhancing the model's ability to learn from data with varying magnitudes. The input sequence length (source sequence) was determined to be 10 time steps (10 minutes), while the prediction sequence length (target sequence) was set to 2 time steps (2 minutes). The model will delineate a source sequence on the dataset and slide it in steps of 1 (Overlap = 1).

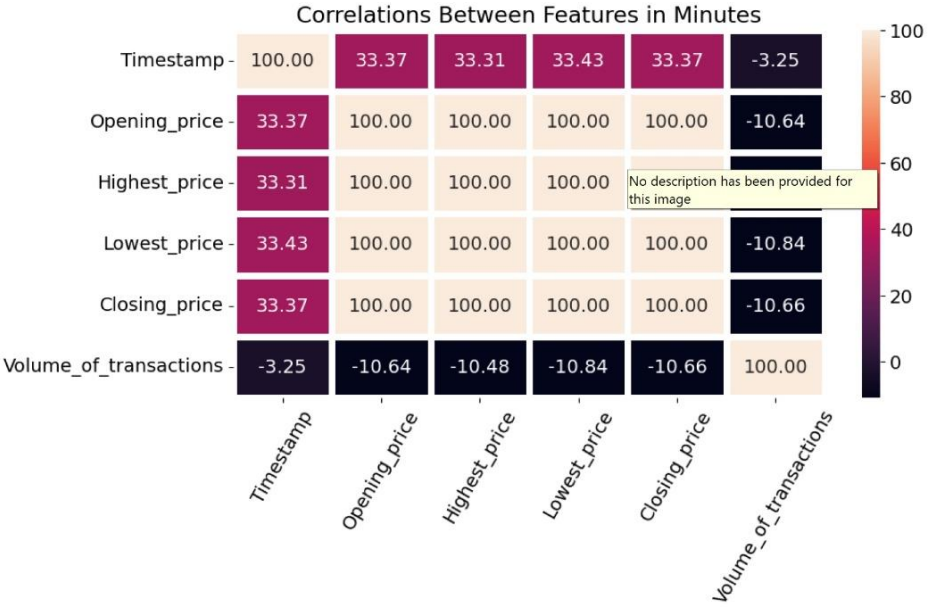

Figure 1. Part of Correlation Matrix.

By using Optuna, it can find the best data preprocessing hyperparameters, as shown in Table 1 below.

Table 1. The Best Data Hyperparameters.

Hyperparameters	Value
Features	34
Scaler	Minmax
Train batch size	32
Bptt source steps	10
Bptt target steps	2
Overlap	1

3.2. Time-Series Transformer (TST) Model Architecture and Hyperparameters

The Time-Series Transformer (TST) model is built on the Transformer architecture, originally designed for natural language processing tasks. It consists of an Encoder-Decoder framework, as illustrated in the figure 2. This structure enables the TST model to efficiently handle sequential data and capture both short- and long-term dependencies in time-series datasets.

1. Encoder: The encoder comprises multiple identical layers ('Nx') that sequentially process the input data. Each layer has the following components:

(a) Multi-Head Attention: This mechanism evaluates the relationships between all time steps in the input, enabling the model to focus on significant temporal dependencies.

(b) Feed-Forward Network (FFN): A fully connected network applied independently to each time step, enabling nonlinear transformations.

(c) Add & Norm: Residual connections and layer normalization improve gradient flow and stabilize training.

2. Decoder: The decoder, similar to the encoder, also comprises multiple identical layers ('Nx'). However, it includes an additional Masked Multi-Head Attention layer, which ensures that the predictions for each time step only depend on previous time steps, preserving the autoregressive property required for sequence prediction.

3. Time Embedding: Both the encoder and decoder utilize time embeddings to represent temporal features, integrating positional information into the model.

4. Output Layer: A linear layer maps the decoder's output to the desired prediction target.

This architecture enables parallel processing and captures complex temporal relationships, making it particularly suitable for financial time-series data with high dimensionality.

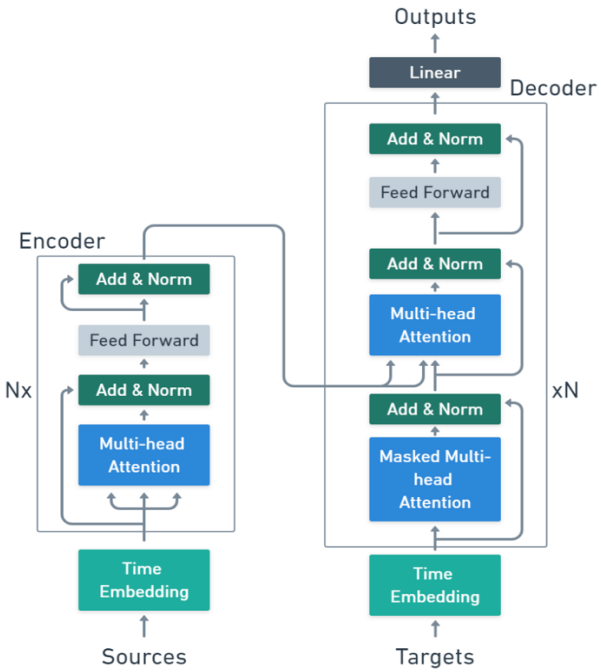

Figure 2. TST Architecture.

The hyperparameters used for the TST model are carefully selected to optimize its performance for predicting Bitcoin price changes (see Figure 2). These parameters govern the architecture's complexity, data processing, and training strategy.

By using Optuna, it can find out the best model architecture and training hyperparameters, as shown in Table 2 below.

Table 2. The Best Model Architecture and Training Hyperparameters.

Hyperparameter	Value
Encoder/Decoder layers	4
Output features	64
Attention heads	16
Dimension of the feedforward network	384
Activation	GeLU
Gradient Clip	0.75
Learning rate	0.5
Learning rate decay factor	0.95
Step size	1.0

These hyperparameters are tailored for the BTC-USDT dataset, balancing computational efficiency and prediction accuracy. The use of min-max normalization ensures all features are scaled consistently, while the 10-to-2 time-step ratio allows the model to focus on short-term price trends. Additionally, the chosen architecture with 4 encoder and decoder layers, 16 attention heads, and the GELU activation provides sufficient complexity to capture the intricate patterns in the financial data.

3.3. Dataset Splitting and Evaluation Metrics

The BTC-USDT dataset was divided into three subsets to ensure robust model evaluation. The training set, comprising 80% of the data, was used to optimize the model parameters during training. The validation set, accounting for 10% of the data, was employed for hyperparameter tuning and model selection, ensuring the model's ability to generalize well to unseen data. Finally, the test set, also comprising 10% of the data, was held out for final evaluation. This division balances the need for sufficient data in training while reserving adequate portions for validation and testing (see Figure 3), effectively preventing issues like overfitting or underfitting during the modeling process.

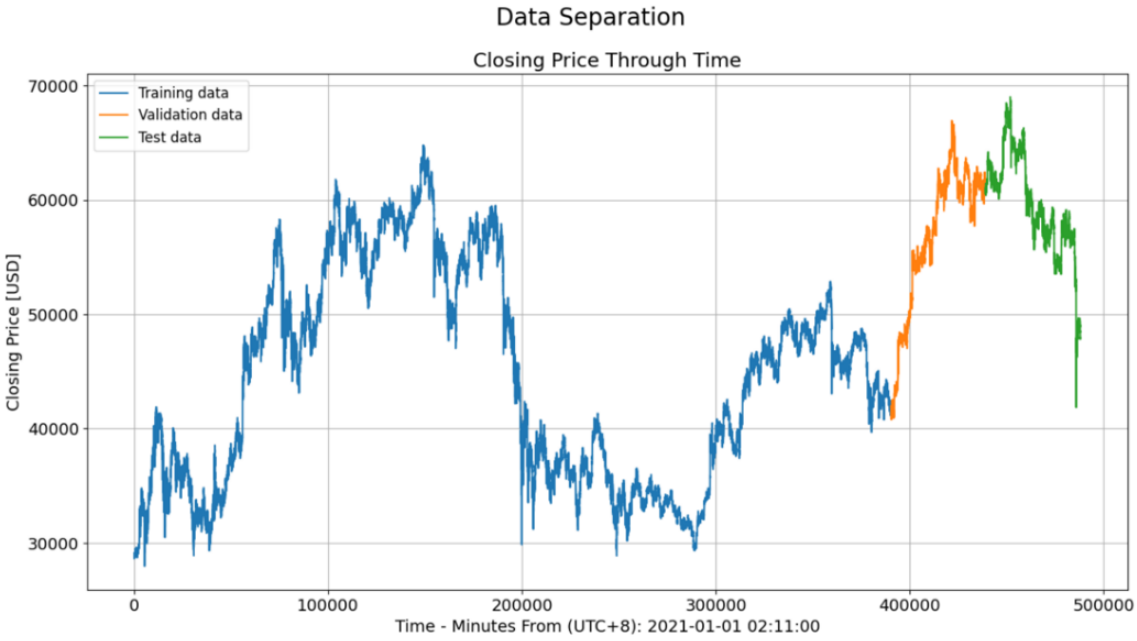

Figure 3. Data Separation.

The model's performance was evaluated on the test set using three key metrics. The Mean Squared Error (MSE) measures the average squared difference between predicted and actual values, penalizing larger errors more heavily and providing a clear indication of the model's accuracy. The Mean Absolute Scaled Error (MASE) offers a normalized measure of prediction quality by scaling absolute errors relative to a naive benchmark model, ensuring that the model's performance is assessed in a context-sensitive manner. Additionally, the Coefficient of Determination (R^2) evaluates how well the model explains the variance in the actual data, with values closer to 1 indicating better explanatory power. Together, these metrics provide a comprehensive understanding of the model's predictive accuracy, robustness, and generalization capabilities, offering critical insights into its overall performance.

3.4. Training

The model was trained over 60 epochs to optimize its parameters. During the first 10 epochs, the validation loss was relatively high, reflecting the model's initial adjustments to the dataset. Between epochs 11 and 20, the loss showed a slower but steady decrease. From epoch 20 onward, the loss values stabilized, converging at approximately 0.000040. Figure 4 visualizes the loss evolution during training, highlighting the progressive improvements in the model's predictive performance over time. This trend demonstrates the model's ability to efficiently learn and generalize from the training data.

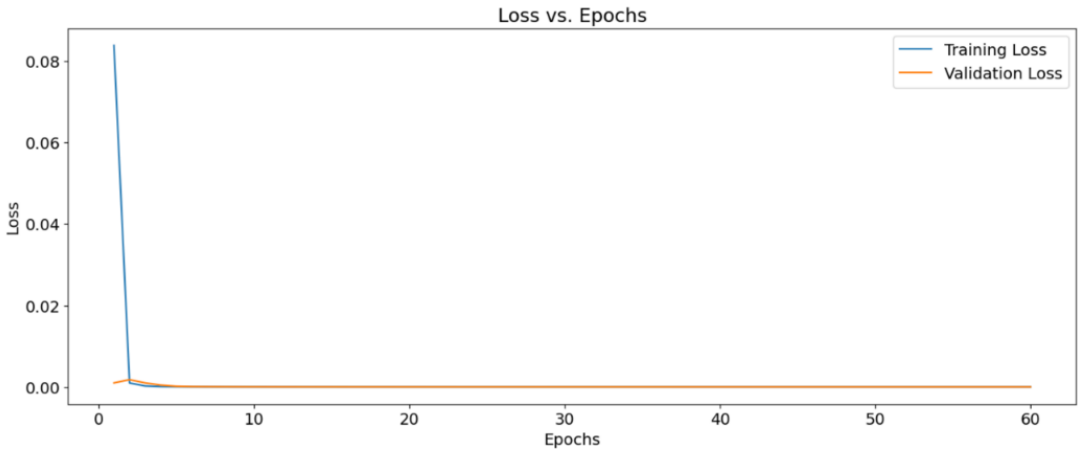

Figure 4. Training and Validation Loss vs. Epochs.

4. Experiments and Results

4.1. Time-Series Transformer vs. LSTM

The experimental results evaluate the performance of the Time-Series Transformer (TST) model against the baseline Long Short-Term Memory (LSTM) model on the BTC-USDT dataset. Table 3 presents the results for key evaluation metrics, including Mean Squared Error (MSE), Mean Absolute Scaled Error (MASE), and Coefficient of Determination (R^2). The TST model demonstrates a clear advantage, achieving lower MSE and MASE values compared to the LSTM, while also exhibiting higher R^2 values, which indicates better explanatory power of the variance in the test data. These results confirm the effectiveness of the TST model in handling short-term cryptocurrency price prediction tasks, particularly in high-dimensional and volatile datasets.

Table 3. Evaluation Metrics of TST and Baseline LSTM.

	TST	LSTM
R^2	0.996774	0.899122
MSE	62001.403974	293071.468541
MASE	4.747746	5.297903

4.2. Visualization of Predictions

To further analyze the prediction capabilities of the TST model, Figure 5 shows the model's fit on the entire dataset, illustrating the alignment between predicted and actual closing prices. While this provides an overview of the model's performance, the detailed trends may be difficult to observe.

Figure 5. Actual vs. Predicted Prices on the Entire Dataset.

Therefore, Figure 6 focuses on the test set portion of the dataset, zooming in to highlight the model's performance in capturing finer details of price movements. The zoomed-in visualization reveals that the TST model aligns closely with actual prices, even during periods of significant fluctuation. These visualizations provide clear evidence of the TST model's robustness and its ability to generalize well to unseen data, making it an effective tool for time-series forecasting in financial markets.

Figure 6. Actual vs. Predicted Prices on the Test Set – Zoom in.

4.3. Error Analyzes

A scatter plot of prediction errors was generated to evaluate the TST model's accuracy across the dataset (see Figure 7). The majority of prediction errors are tightly clustered around zero, indicating that the model effectively minimizes large deviations from actual values. However, there are occasional spikes in error during periods of heightened market volatility, suggesting that extreme price movements pose a challenge for the model. Despite these outliers, the overall distribution shows a balanced error spread with no significant bias, highlighting the TST model's reliability in capturing general price trends.

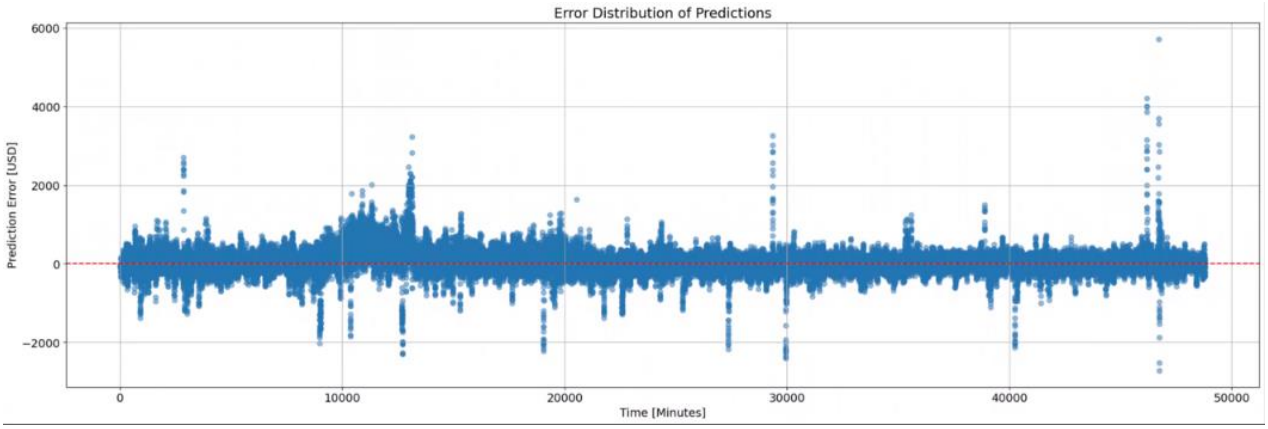

Figure 7. Error Distribution of Predictions.

Based on calculations, the standard deviation of prediction errors across the dataset is approximately $300.20. Given that the average price level of the dataset ranges between $50,000 and $60,000, this error standard deviation represents only about 0.5% to 0.6% of the price level. For high-frequency, minute-level predictions, such an error margin is relatively small and can be considered a precise performance benchmark.

5. Limitation and Reflection

Despite the promising performance of the Time-Series Transformer (TST) model, several limitations in the experimental design warrant attention. First, the dataset used in this study spans only one year of minute-level BTC-USDT trading data. While this provides a robust basis for evaluating short-term prediction capabilities, it limits the model's exposure to diverse market conditions, such as prolonged bull or bear trends, regulatory changes, or extreme volatility events. These factors could significantly impact the model's generalizability when applied to unseen data from other time periods or cryptocurrency assets [6]. Additionally, the study's reliance on a single dataset and a fixed set of technical indicators may overlook other influential variables, such as macroeconomic factors, blockchain-specific metrics, or social sentiment, which could enhance predictive accuracy if incorporated [7].

Another notable limitation is the computational complexity of the TST model. While Transformers are inherently efficient due to their parallel processing capabilities, their training and inference still require significant computational resources, particularly when applied to high-dimensional and large-scale financial data [8]. This resource-intensive nature may constrain the practical application of the model in environments with limited computational infrastructure. Moreover, the fixed hyperparameters, determined through Bayesian optimization for this specific dataset, may not be optimal for other datasets or market conditions. Future research should explore adaptive hyperparameter tuning and model architectures that balance computational efficiency with predictive power.

Reflecting on these limitations, the TST model's predictive capabilities are context-dependent, constrained by dataset scope, computational demands, and the absence of external factors like macroeconomic variables. Expanding datasets across diverse timeframes and integrating adaptive hyperparameter tuning could enhance robustness and accuracy. Incorporating multi-modal data sources, such as social sentiment or blockchain metrics, would further improve adaptability. Moreover, addressing the model's interpretability through explainability techniques is crucial to build trust in financial applications. These steps would expand the model's real-world applicability and address its current limitations effectively.

Building on the foundation of Transformer-based models, future research could explore advanced variants such as the Informer model. Informer introduces a self-attention mechanism optimized for long-sequence data by reducing computational overhead and addressing memory constraints [9]. This

makes it particularly suitable for high-frequency financial datasets, where both scalability and efficiency are critical. By leveraging Informer's ability to focus on sparse yet informative regions of the input sequence, researchers could enhance model performance in capturing intricate temporal dependencies while minimizing resource consumption [10]. Additionally, combining Informer with external factors, such as social sentiment or macroeconomic indicators, could unlock new possibilities for multi-modal time-series forecasting. This direction not only aligns with the ongoing evolution of Transformer architectures but also offers practical advancements for real-world applications in cryptocurrency markets and beyond.

6. Conclusion

This study explored the application of the Time-Series Transformer (TST) model for short-term price prediction in the cryptocurrency market, focusing on the BTC-USDT trading pair. The TST model outperformed traditional approaches like LSTM in predictive accuracy, scalability, and robustness, demonstrating its ability to capture complex temporal dependencies and handle volatile price movements. However, key limitations were identified, including the reliance on a single dataset with fixed technical indicators and the computational demands of the Transformer architecture. Additionally, the lack of external factors, such as macroeconomic variables or social sentiment, limits the model's broader applicability. Addressing these challenges will require integrating diverse data sources, expanding dataset scope, and exploring adaptive Transformer variants like Informer, which can optimize attention mechanisms for long-sequence data while reducing computational overhead. Despite these constraints, the TST model's capacity to generalize across high-dimensional data and adapt to dynamic market conditions positions it as a valuable tool for cryptocurrency analysis. Future efforts to incorporate multi-modal inputs and enhance model interpretability will unlock its full potential, enabling it to tackle the complexities of modern financial forecasting effectively.

References

[1] Sveriges Riksbank: Economic Commentaries: Insights on monetary policy frameworks. *Sveriges* Riksbank *Economic Review* 2014 (2), 73 (2014).

[2] Hua, Y.: Bitcoin price prediction using ARIMA and LSTM. *E3S Web of Conferences* 218, 01050 (2020).

[3] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L.: Transformers in Time Series: A Survey. *ArXiv.org* (2023).

[4] Ali, M., Manthouri, M.: Enhancing Price Prediction in Cryptocurrency Using Transformer Neural Network and Technical Indicators. *ArXiv.org* (2024).

[5] Penmetsa, S., Vemula, M.: Cryptocurrency Price Prediction with LSTM and Transformer Models Leveraging Momentum and Volatility Technical Indicators. (2023).

[6] Bisht, G.S., Monga, T., Goel, A., Sengar, N., Bahl, V.: Cryptocurrency Price Prediction Using LSTM, ARIMA, and Linear Regression. *International Research Journal of Modernization in Engineering Technology and Science* (2023).

[7] Jung, H.S., Kim, J.H., Lee, H.: Decoding Bitcoin: leveraging macro- and micro-factors in time series analysis for price prediction. *PeerJ Computer Science* 10, e2314 (2024).

[8] Amundi: Time Series Forecasting with Transformer Models and application for Asset Management. Amundi Research Center (2023). https://research-center.amundi.com/article/time-series-forecasting-transformer-models-and-application-asset-management, last accessed 2023/10/25.

[9] Yang, Y., Dong, Y.: Informer_Casual_LSTM: Causal Characteristics and Non-Stationary Time Series Prediction. 1 – 6 (2024).

[10] Zhu, Q., Han, J., Chai, K., Zhao, C.: Time Series Analysis Based on Informer Algorithms: A Survey. *Symmetry* 15 (4), 951 (2023).