Journal of Emerging Information Systems and Business Intelligence
ISSN: 2774-3993
Journal homepage: https://ejournal.unesa.ac.id/index.php/JEISBI/

Bitcoin Transaction Multivariate Forecasting Analysis Deep Learning Model Walk Forward Validation

Muhammad Dafi Bagas Nugroho[1], Wiyli Yustanti[2]

[1,2]*Program Study of Information System, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia*

muhammaddafi.22139@mhs.unesa.ac.id, *wiyliyustanti@unesa.ac.id*

ABSTRACT

The volatile and non-linear movement of Bitcoin prices makes price prediction a complex problem in time series analysis. This study aims to compare the performance of several deep learning models, namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Transformer, and Temporal Fusion Transformer (TFT), in predicting Bitcoin closing prices based on multivariate data. The dataset consists of daily historical data from 2020 to 2025, including Open, High, Low, Close, and Volume features. Model evaluation was conducted using the Walk Forward Validation (WFV) approach with 5 folds and was compared with the Cross Validation (CV) method. Three data split scenarios were applied: 70:30, 80:20, and 90:10. Model performance was measured using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Symmetric MAPE (sMAPE), and the coefficient of determination (R^2). Furthermore, the Wilcoxon Signed-Rank Test was employed to analyze the statistical significance of performance differences between validation methods. The results indicate that the GRU model under the 90:10 data split scenario achieved the best performance, with a median MAE of 0.0116 and RMSE of 0.0179, along with an R^2 value of 0.8622. This model demonstrated lower prediction errors and greater stability compared to the other models. Meanwhile, the Wilcoxon test results showed no significant difference between Walk Forward Validation and Cross Validation (p-value > 0.05), indicating that both validation methods produce statistically equivalent performance. Based on these findings, the GRU model is recommended as the most optimal model for Bitcoin price prediction under the experimental configuration used in this study.

Keywords: Deep Learning, Time Series Forecasting, GRU, LSTM, Transformer, Walk Forward Validation, Wilcoxon Test.

Article Info:
Article history:
Received May 06, 2026
Revised May 20, 2026
Accepted June 02, 2026

Corresponding Author
Muhammad Dafi Bagas Nugroho
Universitas Negeri Surabaya, Surabaya, Indonesia
muhammaddafi.22139@mhs.unesa.ac.id

1. INTRODUCTION

The era of global financial digitalization has created a fundamental transformation in the investment and asset trading ecosystem. Technological developments *Blockchain* that began in 2009 has allowed the emergence of digital currencies or *Cryptocurrency* as a new

asset class that offers an alternative to the traditional financial system. This phenomenon reflects a paradigm shift from centralized financial systems to decentralized systems, which provide more inclusive and efficient financial access opportunities in the face of increasingly complex global market dynamics [1].

Bitcoin, as the first and largest cryptocurrency, has been a pioneer in this digital financial revolution. Since its inception, bitcoin has experienced significant adoption by various circles, ranging from retail investors to institutions, and even some countries have made it legal tender [2]. Bitcoin's market capitalization of hundreds of billions of dollars shows that this digital asset has evolved from a technological experiment to a serious investment instrument.

However, the unique characteristics of bitcoin as a digital asset create special challenges in terms of extreme price volatility. The cryptocurrency market , especially bitcoin, is characterized by price fluctuations that can reach tens of percent in a short period of time, far exceeding the volatility of traditional financial assets such as stocks or bonds. This high volatility is caused by a variety of complex factors, such as technical factors related to the mechanism of *the blockchain* itself [2]. This results in the price of bitcoin often moving over a wide range and exhibiting inconsistent patterns across different time periods.

The complexity of bitcoin price behavior is further strengthened by the characteristics of data that are *non-linear*, undergo *regime changes*, and have multivariate relationships with interrelated OHLCV (*open, high, low, close, volume*) features. Recent research shows that not only is closing *price* necessary for accurate predictions, but the combination of OHLCV features provides more comprehensive information for forecasting systems [3]. This *multivariate interconnectedness* adds a dimension of complexity to prediction modeling because it requires consideration of many *interrelated price-based* variables in capturing the complex dynamics of the bitcoin market.

The urgency of developing an accurate bitcoin prediction model is not only an academic challenge, but a practical necessity that has direct implications for risk management and the formulation of trading strategies. The accuracy of cryptocurrency price predictions is crucial for investors who *are informed*, manage risk exposure, and optimize returns. In addition, for financial institutions and companies that adopt bitcoin as part of their services, the ability to predict price movements is fundamental to sustainable business operations [4].

The development of *machine learning* and *deep learning technologies* has opened up new opportunities in overcoming the complexity of bitcoin price predictions. Various neural network architectures such as *Long Short Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), and *Transformer* have shown potential in capturing complex patterns and temporal dependencies contained in *time series* data [5]. The main advantage of the deep learning approach lies in automatically extracting features from OHLCV raw data and capturing *non-linear relationships* that are difficult to identify by statistical methods.

Previous studies that discussed bitcoin price prediction using *deep learning* have shown varied results, with each model having different characteristics and performance. Comparative studies show that GRU can achieve a MAPE of 0.38% for bitcoin data, while LSTM shows consistent performance [6]. However, the results of each study indicate the need for standardization in model evaluation and validation methodologies, especially in the optimal use *of OHLCV* multivariate features to improve prediction accuracy.

In the context of financial *time series prediction*, the right validation method is a critical aspect that determines the reliability of the model. *Walk Forward Validation* (WFV) has been recognized as the *gold standard for* time series *model evaluation* due to its ability to maintain the temporal sequence of data and prevent *look ahead bias* that can lead to *data leakage* [7]. This technique simulates *real-world trading* conditions where the model is trained

with historical data and tested in subsequent periods sequentially, providing more realistic performance estimates and providing more accurate results.

Based on the identification of *gaps* and limitations of previous research, there is an urgent need to conduct a comprehensive evaluation of various *deep learning* architectures for bitcoin prediction using strict validation protocols. This study aims to fill these gaps by conducting a systematic comparison between LSTM, GRU, *Transformer and Transformer temporal fusion models* in the context of *multivariate* bitcoin transaction forecasting using the OHLCV feature and applying *walk forward validation* as the main evaluation method. This approach is expected to make a significant contribution to the development of more accurate *cryptocurrency forecasting* methodologies for practical implementation in *strategic trading*.

2. METHODS

This study uses the CRISP-DM methodology which consists of six main stages, namely *business understanding, data understanding, data preparation, modeling, evaluation,* and *deployment,* which is carried out systematically to produce an optimal prediction model. At the stage *Business Understanding*, the research focuses on the problem of Bitcoin price volatility and the need for a predictive system based on *Deep Learning* accurate and easily accessible. Stages *Data Understanding* using Bitcoin historical data from Yahoo Finance for the period January 2020 to October 2025 with OHLCV's multivariate feature to capture market dynamics. Furthermore, the data *Preparation* Covering the process *Cleaning* (deletion *missing value* and format consistency), data aggregation *Time series*, manufacture *Sequence Windowing* (*Lookback* 64 days, 1 day horizon), as well as dataset sharing using *Time Series Cross Validation* and *Walk-Forward Validation* with 70:30, 80:20, and 90:10 split scenarios and the implementation of *Scaling* to prevent data *leakage.* At the modeling stage, four models were used *deep learning, namely LSTM, GRU, Transformer, and Temporal Fusion Transformer* (TFT) which is designed to handle multivariate time series data, then validated using CV and WFV. The best models are determined based on evaluation metrics such as MAE, RMSE, MAPE, sMAPE, and R^2 as well as performance stability between *fold.* Stages *Evaluation* ensure the reliability of the model in predicting new data, while the *deployment* Implement the best model into a web-based system using the *waterfall*, Includes needs analysis, system design *Client Server*, implementation, testing (units, integrations, and performance).

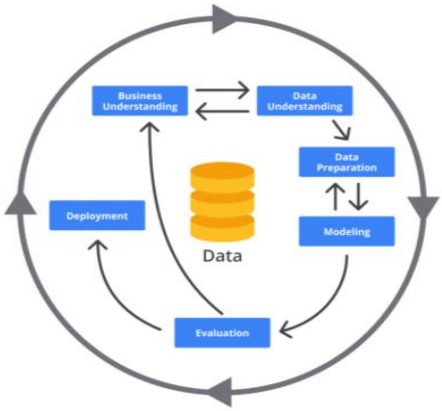

Figure 1. *CRISP-DM*

3. RESULTS AND DISCUSSION

3.1 Dataset and Variable Description (OHLCV)

This research dataset is in the form of historical Bitcoin data from the Yahoo Finance platform in the form of a daily time series for the period January 2020–October 2025 as many as 2,119 data. The data is stored in CSV format that contains dates, OHLC (*Open, High, Low, Close)* prices, and transaction volumes, where each line represents one daily observation chronologically. The study uses the OHLCV (Open, High, Low, Close, Volume) multivariate feature to comprehensively represent price movements and market activity as a basis for the forecasting process.

Table 1. Table of definitions of research variables

Variable	Description	Data Type
Open	Bitcoin's daily opening price	Numerical (continuous)
High	Daily high price of Bitcoin	Numerical (continuous)
Low	Bitcoin's daily lowest price	Numerical (continuous)
Close	Bitcoin's daily closing price	Numerical (continuous)
Volume	Bitcoin transaction volume per period	Numerical (continuous)

3.2 Normalization/*Scaling* data

After the cleanup and *sequencing* process, normalization is performed using *MinMaxScaler* to equalize the scale of the OHLCV feature to the range of 0–1 to support model optimization. *Scaling* is applied to the input (X) and target (y/*Close*) features with two *separate scalers* to maintain consistency. To prevent data *leakage*, *the scaler* is only fitted to the training data in each *fold*, while the validation/test data is only transformed using the *scaler*. Implementation is carried out through the *scale_fold_minmax*() function which handles the normalization of sequence data with *a temporary reshape* process.

```
Def scale_fold_minmax(X_train_raw, X_test_raw, y_train_raw, y_test_raw):
    """
    Fit scaler hanya pada train fold (anti leakage)
    """
    n_features = X_train_raw.shape[-1]

    X_scaler = MinMaxScaler()
    X_tr_sc = X_scaler.fit_transform(X_train_raw.reshape(-1, n_features)).reshape(X_train_raw.shape)
    X_te_sc = X_scaler.transform(X_test_raw.reshape(-1, n_features)).reshape(X_test_raw.shape)

    y_scaler = MinMaxScaler()
    y_tr_sc = y_scaler.fit_transform(y_train_raw.reshape(-1, 1)).reshape(-1)
    y_te_sc = y_scaler.transform(y_test_raw.reshape(-1, 1)).reshape(-1)

    return X_tr_sc, X_te_sc, y_tr_sc, y_te_sc, X_scaler, y_scaler
```

Figure 2. *Code Scalling*

3.3 *Windowing (Lookback)*

After the cleaning process, the data is formed into *a sequence* (*windowing*) because the LSTM/GRU model requires input in the form of a time sequence. This study uses *the 64-day lookback parameter and a 1-day horizon with the OHLCV feature as the input and* Close *as the target. Each* sequence *consists of data from the previous 64 days to predict the closing price of the next day, with the number of* sequences of n−lookback−horizon+1. This process is implemented through the *create_sequences_from_df*() function that forms pairs (X, y) while also storing the target index (*y_idx*) to maintain the time sequence. The result is a *dimensional X_raw* (*n_seq*, 64, 5), *y_raw* as a target, and *y_idx* as a label index. Data segmentation is done after *windowing* by *y_idx* to keep it chronological, with split scenarios of 70:30, 80:20, and 90:10.

3.4 WFV Scheme

Model evaluation was carried out in three data sharing scenarios (70:30, 80:20, 90:10) to test performance consistency, with time-based distribution to prevent data *leakage.* The *data train* was taken from the initial period, while the data *was tested* from the final period after *the windowing* process. Furthermore, the *Walk Forward Validation* (WFV) method was used on the training data with an *expanding window scheme* to measure the stability of the model between periods. The configuration used includes 5 *folds,* an initial window of 1000 data, and a step of 100. Each fold expands the training data and tests the next segment. The results of the WFV evaluation are in the form of metrics for each *fold* (*Folds* 1–5) and average values as a summary of performance, which are used as the basis for compiling the results of the analysis in the study.

3.5 LSTM

The LSTM model is used to capture temporal patterns in Bitcoin price data with *a dimensional (64.5) (OHLCV) input* sequence and *a Close target (*horizon 1). The architecture uses two LSTM (*stacked*) layers with *Dropout* to reduce *overfitting*, as well as *Dense(1) output for regression. The main hyperparameters* included 64 units, 0.2 dropouts, Adam optimizer (*learning rate* 0.001), and MSE loss. The training was conducted with *10% split validation, 32 batch sizes,* and EarlyStopping *and ReduceLROnPlateau callbacks to maintain stability and prevent overfitting. The model output* is in the form of a prediction of *the Close* value one step forward which is evaluated using MAE, RMSE, MAPE, sMAPE, and R^2 in the WFV and CV schemes.

3.6 GRU

The GRU model is used to study temporal patterns in Bitcoin price data with dimensional *input sequences (64.5) (OHLCV) and* Close *targets (horizon 1). The architecture uses two layers of GRU (*stacked*) with* Dropout*, as well as* Dense(1) output for regression. The hyperparameters used include 64 units, *0.2* dropouts, *Adam optimizer* (*learning rate* 0.001), and MSE loss. The training process used 10% validation split, *batch size* 32, and *EarlyStopping* and *ReduceLROnPlateau* callbacks to maintain stability and prevent *overfitting. The model output* was a one-step *forward Close* value prediction evaluated using MAE, RMSE, MAPE, sMAPE, and R^2 in the WFV and CV schemes.

3.7 Temporal Fusion Transformer

The *Temporal Fusion Transformer* (TFT) model is used as an attention-based approach to Bitcoin time series forecasting, which combines variable selection, LSTM *encoder–decoder,* and *Multi-Head Attention* to capture temporal dependencies in a complex manner. The architecture includes *a Variable Selection Network* for feature weighting, latent representation (*hidden size* 256), LSTM *encoder–decoder* (2 layers), *Multi-Head Attention* (8 *heads*), as well as *gating mechanism* and skip *connections* for improved stability. *The output* uses *quantile regression* (*QuantileLoss*) so that it is able to predict the distribution of values. *The main hyperparameters* include *hidden_size* 256, *hidden_continuous_size* 128, *lstm_layers 2, attention_head_size* 8, *dropout* 0.1, and *a learning rate* of about 1e-3/1e-4. The training was conducted with normalized data (*GroupNormalizer log transform*), using *an encoder–decoder*

scheme based on *TimeSeriesDataSet*, and following *a batch size* and *epoch configuration* according to the evaluation scheme (WFV/CV).

3.8 Transform

The Transformer *model* is used to study temporal dependencies on Bitcoin price data using a *self-attention mechanism*, so that it is able to capture short- and long-term patterns efficiently. The input is in the form of *sequence* (64, 5) (OHLCV) with a Close prediction *output* (horizon 1). The architecture includes input projection to the representation dimension (d_model), Multi-Head *Self-Attention*, residual *connection* and *layer normalization, feed-forward network,* and *pooling* and *Dense*(1) for regression. *The main hyperparameters* include d_*model 64, num_heads 4, key_dim 32, ff_dim 128*, *dropout* 0.1, *optimizer* Adam (lr 0.001), and MSE loss. The training followed a common configuration (*MinMaxScaler normalization, batch size, epoch* according to the scheme, *0.1 split validation*, and EarlyStopping and ReduceLROnPlateau). The model was then evaluated using MAE, RMSE, MAPE, sMAPE, and R^2 on the WFV and CV schemes.

3.9 Evaluation results (MAE, RMSE)

MAE WFV Results per Fold (Fold 1–5)

Model GRU – MAE WFV per Fold

Table 2. GRU Table by MAE

Scenario	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Median	Mean
70:30	0.0137	0.0112	0.0065	0.0137	0.0212	0.0137	0.01326
80:20	0.0124	0.0107	0.0076	0.0145	0.0303	0.0124	0.01510
90:10	0.0116	0.0112	0.0080	0.0168	0.0331	0.0116	0.01614

Summary: The GRU tends to be stable in Folds 1–4, but there is an increase in errors in Fold-5 in all scenarios. The median MAE GRU is best at 90:10 (0.0116), followed by 80:20 and 70:30 (narrow margin).

Model LSTM – MAE WFV per Fold

Table 3. LSTM Table by MAE

Scenario	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Median	Mean
70:30	0.0159	0.0168	0.0093	0.0193	0.0335	0.0168	0.01896
80:20	0.0153	0.0167	0.0098	0.0198	0.0451	0.0167	0.02134
90:10	0.0158	0.0165	0.0115	0.0194	0.0415	0.0165	0.02094

Summary: LSTM is higher than GRU on most folds. Fold-5 shows significantly increased errors (especially 80:20 and 90:10 scenarios). LSTM's best median MAE is at 90:10 (0.0165)

Model TFT – MAE WFV per Fold

Table 4. TFT Table by MAE

Scenario	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Median	Mean
70:30	0.0153	0.0165	0.0143	0.0094	0.0364	0.0153	0.01382
80:20	0.0223	0.0189	0.0193	0.0432	0.0234	0.0223	**0.02542**
90:10	0.0082	0.0104	0.0116	0.0201	0.0287	0.0116	0.01580

Summary: TFT has a fairly high variation between folds (specifically Fold-4 in the 80:20 scenario and Fold-5 in the 70:30 scenario). However, in the 90:10 scenario the TFT is able to achieve competitive performance. TFT's best median MAE is in the 90:10 scenario (0.0116), equal to the GRU and better than the LSTM in the same scenario.

Model Transformer – MAE WFV per Fold

Table 5. Table of Transformers by MAE

Scenario	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Median	Mean
70:30	0.0115	0.0140	0.0087	0.1128	0.0173	0.0140	0.03286
80:20	0.0262	0.0366	0.0169	0.0165	0.0564	0.0262	0.03052
90:10	0.0159	0.0157	0.0099	0.0257	0.0464	0.0159	0.02272

Summary: The Transformer shows cases of instability in the 70:30 scenario, especially the Fold-4 (MAE = 0.1128) which makes the mean value increase sharply. Therefore, in this model the median is more representative than the mean. The best median of the Transformer is at 70:30 (0.0140), but it is worth considering the stability between the folds.

Provisional Conclusions (based on MAE WFV) Based on MAE per fold:

1. GRU provides the lowest performance and is relatively stable in most folds.
2. LSTM is below the GRU (larger MAE) and tends to increase in the final Fold.
3. TFT shows the highest error.
4. *Transformers* can be excellent on some folds, but risk being unstable (e.g. spikes on certain Folds).

Performance Recap per Split Scenario (70:30, 80:20, 90:10)

After obtaining the results of the evaluation per fold in the Walk Forward Validation (WFV) scheme, the next stage is to conduct a recapitulation to facilitate the comparison of performance between models in each split scenario. The recapitulation is carried out by calculating the summary size (e.g. **median** and **mean**) of the MAE value per fold (Fold 1–5). The use of **the median** is important because it is more robust to the extreme values of a particular fold, especially when there is a fold that produces a much larger error than other folds.

Basis of recapitulation

1. Metrics recapped: MAE_scaled
2. Data source: WFV evaluation results per fold (Fold 1–5)
3. Recap shown:
   a. Median MAE_scaled per model scenario
   b. Mean MAE_scaled per scenario–model (for comparison)

MAE WFV Recap by Split Scenario

Scenario 70:30 (WFV)

Table 6. Scenario Recapitulation Table 70:30

Models	Median MAE	Mean MAE	Ratings
GRU	0.0137	0.01326	1
Transform	0.0140	0.03286	2
TFT	0.0153	0.01382	3
LSTM	0.0168	0.01896	4

Analysis: In the 70:30 scenario, the GRU has the lowest median MAE (0.0137). The transformer has a competitive median (0.0140), but the mean value is much larger due to the

error spike in the Fold-4, so the stability is lower than that of the GRU. TFT is ranked third, while LSTM has the highest median MAE in this scenario.

Scenario 80:20 (WFV)

Table 7. Table Scenario 80:20

Models	Median MAE	Mean MAE	Ratings
GRU	0.0124	0.01510	1
LSTM	0.0167	0.02134	2
TFT	0.0223	0.02542	3
Transform	0.0262	0.03052	4

Analysis: In the 80:20 scenario, GRU still performs best. LSTM is in second place, while *Transformer* and TFT have higher MAE.

Scenario 90:10 (WFV)

Table 8. Scenario Table 90:10

Models	Median MAE	Mean MAE	Ratings
TFT	0.0116	0.01580	1
GRU	0.0116	0.01614	2
Transform	0.0159	0.02272	3
LSTM	0.0165	0.02094	4

Analysis: In the 90:10 scenario, TFT appears as the best model with the lowest median MAE (0.0116), equaling the GRU but with a lower mean MAE (0.01580 vs 0.01614) so it ranks first. GRU ranks second with a very narrow margin. Transformer ranks third, while LSTM records the highest median MAE (0.0165) in this scenario.

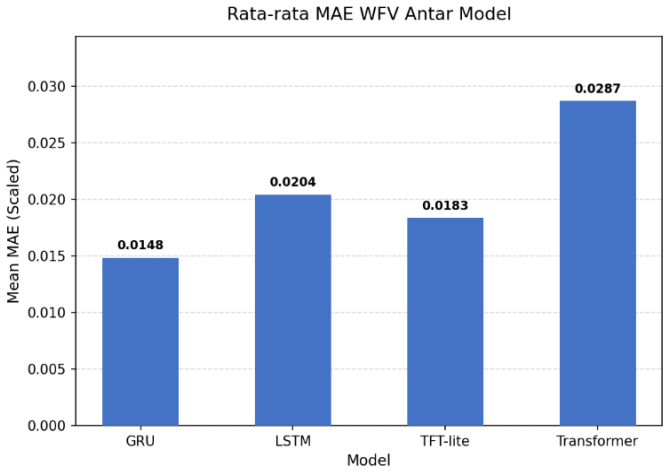

Figure 3. MAE WFV Average Barchart Images

Conclusion of recapitulation (based on MAE) Based on MAE per fold:

1. GRU provides the best and relatively stable performance in most folds, especially 70:30 and 80:20 scenarios.
2. LSTM is below the GRU (MAE is larger) and tends to increase in the final fold.
3. TFT shows high error in 70:30 and 80:20 scenarios, but performs competitively in 90:10 scenarios.
4. Transformers can be excellent on multiple folds, but risk being unstable (e.g. Fold-4 spike in a 70:30 scenario).

3.10 Split scenario comparison

Model Comparison Results (Median WFV) per Split Scenario

The comparison results are presented in Table 4.x–4.z. The numbers shown are the median WFV (Fold 1–5).

WFV Metric Median Recap – 70:30 Scenario

Table 9. Median Metric Table 70:30

Models	MAE	RMSE	MAPE	sMAPE	R^2
GRU	**0.0137**	**0.0180**	**2.5135**	**2.5159**	**0.8612**
Transform	0.0140	0.0191	3.1378	3.1527	0.1503
TFT	0.0153	0.0199	3.3477	3.5420	0.0526
LSTM	0.0168	0.0226	3.7674	3.7857	0.7805

Interpretation: In a 70:30 scenario, the GRU model has the smallest error in all error metrics (MAE/RMSE/MAPE/sMAPE) and the highest R^2, making it the best model in this scenario.

WFV Metric Median Recap – 80:20 Scenario

Table 10. Median Metric Table 80:20

Models	MAE	RMSE	MAPE	sMAPE	R^2
GRU	**0.0124**	**0.0182**	**2.3993**	**2.4432**	**0.8716**
LSTM	0.0167	0.0223	3.7101	3.7102	0.7647
TFT	0.0223	0.0280	4.1651	4.1751	0.2062
Transform	0.0262	0.0330	6.5477	6.5645	-0.2504

Interpretation: In the 80:20 scenario, GRU again excels with the lowest median MAE and RMSE, lowest MAPE/sMAPE, and highest R^2.

WFV Metric Median Recap – 90:10 Scenario

Table 11. Median Metric Table 90:10

Models	MAE	RMSE	MAPE	sMAPE	R^2
TFT	**0.0116**	**0.0119**	**1.1131**	**1.1609**	**3.3315**
GRU	0.0116	0.0179	2.7668	2.8296	0.8622
Transform	0.0159	0.0204	3.1532	3.1814	0.7842
LSTM	0.0165	0.0215	3.6864	3.7642	0.7808

Interpretation: In a 90:10 scenario, TFT is the best model in all error metrics and has the highest R^2.

Figure 4. WFV median heatmap image

3.11 Best Model Selection

Based on the results of the comparison on all split scenarios, it can be concluded that:

1. GRU always produces the lowest error values (MAE, RMSE, MAPE, sMAPE) and the highest R^2 values in scenarios of 70 : 30, and 80 : 20.
2. This consistency occurs across three split scenarios (70:30, 80:20, 90:10), so the election results don't depend on one specific data-sharing configuration.

Thus, the GRU model was selected as the best model candidate based on the study's main evaluation metrics.

Table 12. Best Model Performance Summary Table (GRU – 90:10)

Models	Split Scenario	MAE	RMSE	MAPE(%)	sMAPE (%)	R^2
GRU	90:10	0.0116	0.0179	2.7668	2.8296	0.8622

3.12 Wilcoxon Signed-Rank Test Results (Two-Sided, n = 5)

Testing Procedure

After the normality test showed that most of the difference distributions did not meet the normality assumptions, the difference test was performed using the Wilcoxon Signed-Rank Test with a two-sided approach. The test was performed separately for each combination:

1. Model (GRU, LSTM, TFT, *Transformer*)
2. Scenario (70:30, 80:20, 90:10)
3. Metrics (MAE, RMSE, MAPE, smape, R^2)

In each combination there were five observation pairs (n = 5) derived from the five validation folds.

Hypotheses tested:

$$H_0 \text{Median}(\quad - \quad) = 0$$
$$_1 \text{:Median}(\quad - \quad) \neq 0$$

The level of significance used is:

$$\alpha=0.05$$

Test Implementation

Here are the code snippets used in testing:

```
# ============================
# 3. WILCOXON PER KOMBINASI
# ============================
for (model, skenario, metric), group in df.groupby(["model", "skenario", "metric"]):

    group = group.sort_values("fold")

    wfv = group["wfv"].values
    cv  = group["cv"].values

    # Selisih berpasangan
    diff = wfv - cv

    # Pastikan ada data non-zero
    if np.sum(diff != 0) > 0:

        # Wilcoxon two-sided
        stat, p_value = wilcoxon(wfv, cv, alternative="two-sided")

        # Median difference
        median_diff = np.median(diff)

        # Interpretasi arah performa
        if metric.lower() in ["mae", "rmse", "mape", "smape"]:
            # Error metric → lebih kecil lebih baik
            arah = "WFV lebih baik" if median_diff < 0 else "CV lebih baik"
        else:
            # R2 → lebih besar lebih baik
            arah = "WFV lebih baik" if median_diff > 0 else "CV lebih baik"

        # Kesimpulan signifikansi
        if p_value < alpha:
            kesimpulan = "Signifikan (p < 0.05)"
        else:
            kesimpulan = "Tidak Signifikan (p ≥ 0.05)"

        results.append({
            "Model": model,
            "Skenario": skenario,
            "Metric": metric,
            "n_fold": len(diff),  # harus 5
            "W_statistic": round(stat, 6),
            "p_value_two_sided": round(p_value, 6),
            "Median_Difference": round(median_diff, 6),
            "Arah_Performa": arah,
            "Kesimpulan": kesimpulan
        })
```

Figure 5. Test Code

Summary of Test Results

Based on the results of the Wilcoxon Signed-Rank Test (two-sided, n = 5) stored in *the hasil_wilcoxon_two_sided_n5.xlsx* file, the following summary was obtained:

Table 13. Test Results Summary Table

Categories	Number of Combinations
Significant ($p < 0.05$)	0
Insignificant ($p \geq 0.05$)	60
Total Combinations	**60**

Tests were carried out on 60 combinations consisting of:

1. 4 models
2. 3 Data sharing scenarios
3. 5 evaluation metrics

In each combination, five pairs of values (n = 5) derived from five validation pits were tested.

The results showed that all combinations resulted in a p-value ≥ 0.05. Thus, at a significance level of 5%, there was not enough statistical evidence to state a significant difference between the results of the Walk Forward Validation (WFV) and Cross Validation (CV) evaluations.

Analysis of Insignificant Causes

The insignificance of the test results can be explained by the following statistical factors:

1. Small Sample Size (n = 5)
   Each test was conducted with five pairs of observations. The small sample size caused the statistical power to be low. With limited n, the Wilcoxon test required a large and consistent difference in order to produce a p-value < 0.05. In the context of this study, the difference between WFV and CV was not consistently large enough in all five folds to produce statistical significance.
2. Inter-Fold Variability
   The model performance on each fold shows natural variations due to the characteristics of the time series data. The variation between these folds can be greater than the differences between the validation methods themselves. Statistically, if the internal variability is high, it is difficult to detect small effects caused by differences in evaluation methods.
3. Relatively Small Numerical Differences
   Although there is a difference in evaluation values between WFV and CV in some combinations, they tend to:
   a. Inconsistent direction across the fold.
   b. Not large enough compared to the variation in model performance.

Because Wilcoxon tests the median difference to zero, if the median difference is close to zero or unstable, the test results are likely to be insignificant.

Interpretation of Test Results

Interpretation Based on Statistical Significance

Based on the results of the Wilcoxon Signed-Rank test (two-sided, n = 5), all 60 model–scenario–metric combinations resulted in a p≥ value of 0.05. This shows that at a significance level of 5%, there was no statistically significant difference between the performance of Walk Forward Validation (WFV) and Cross Validation (CV). Formally, the decisions taken were:

Failed to reject H0

Which means the median difference between WFV and CV is not significantly different from zero.

Thus, statistically no evidence was found strong enough to state that one validation method is consistently superior to the other at the validation fold level.

Interpretation Based on the Direction of Difference

Although not statistically significant, the interpretation is also carried out based on the median direction of the difference:

$$\text{Median Difference} = \text{Median}(WFV - CV)$$

The interpretation of the direction is carried out as follows:

1. For MAE, RMSE, MAPE, and sMAPE:
   a. The median difference < 0 → WFV results in smaller errors
2. For R^2:
   b. The median difference of > 0 → WFV results in a higher determination coefficient

Here is the table of average and median difference (WFV-CV):

1. Globa Summary

Table 13. Global Summary Table of Differences

Average Difference (WFV-CV)	Median Difference (WFV-CV)
-25.178	-0.05515

Interpretation:

a. The median is close to zero → the general difference is relatively small
b. Mean is greater in absolute terms because it is influenced by outliers (especially MAPE & sMAPE)

2. Summary Per Model

Table 14. Summary Table by model

Models	Average Difference	Median Difference
GRU	-11.590	-0.0241
LSTM	-20.591	-0.0297
TFT	-46.392	-0.1529
Transformer	-22.139	-0.0288

Interpretation:

a. The median of the entire model is close to zero
b. TFT has the largest median difference numerically
c. None of the models showed a consistent major shift

3. Summary Per Metric

Table 15. Summary Table Per Metric

Metrics	Average Difference	Median Difference
MAE	-0.2015	-0.00785
MAP	-52.716	-14.243
R2	-0.8453	-0.0785
RMSE	-0.2412	-0.0128
sMAPE	-60.295	-142.565

Interpretation:

a. The median MAE and RMSE are very small (≈ 0)
b. MAPE & sMAPE have a large mean because of the percentage scale
c. R^2 tends to be slightly smaller at WFV (negative median)

Based on the test results, some combinations showed a numerical tendency that WFV resulted in slightly smaller errors than CVs, but the difference was not large enough and not consistent enough to produce statistical significance.

This indicates that the differences that emerge are more numerical than statistically substantive.

CONCLUSION

Based on the results of the study, it can be concluded that the GRU model is the best *deep learning architecture* in the multivariate forecasting of Bitcoin transactions and prices, as it shows the most stable and consistent performance compared to LSTM, *Transformer,* and TFT based on MAE, RMSE, MAPE, sMAPE, and R^2 metrics, with the best results in the 90:10 scenario (MAE 0.0116; RMSE 0.0179; MAPE 2.76%; R^2 0.8622). The *Walk Forward Validation* (WFV) approach has been proven to provide a more realistic evaluation in the context of time series data because it is able to simulate real prediction conditions through an *expanding window scheme*, although statistically it does not always show significant differences compared to *Time Series Cross Validation* (CV), but is more stable in non-stationary data such as Bitcoin. The implementation of the GRU model into the prediction system also shows good generalization ability of new data without significant overfitting indications . In addition, the use of appropriate validation methods and the application of *per-fold scaling* to prevent data *leakage* play an important role in increasing the reliability and objectivity of model evaluation. Overall, the success of forecasting is greatly influenced by the selection of appropriate model architecture and validation methods, resulting in accurate, reliable, and applicable prediction systems.

ACKNOWLEDGEMENTS

Praise be to Allah SWT for all His graces, gifts, and guidance so that the author can complete the journal entitled "Analysis of Multivariate Forecasting of Bitcoin Transactions on *Deep Learning* Models with *Walk Forward Validation*" well. This journal was compiled as one of the requirements to obtain a Bachelor's degree in the S1 Information Systems Study Program, Faculty of Engineering, State University of Surabaya. This study discusses the application and comparison of several *deep learning* models in conducting multivariate forecasting on Bitcoin transaction data with *the Walk Forward Validation* approach. The author realizes that in the process of compiling this journal, it is inseparable from the support, guidance, and assistance from various parties. Therefore, on this occasion the author would like to express his deepest gratitude.

REFERENCES

[1] A. Kumar and T. Ji, “CryptoPulse: Short-Term Cryptocurrency Forecasting with Dual-Prediction and Cross-Correlated Market Indicators,” 2025. doi: 10.1109/BigData62323.2024.10982029.

[2] T. R. Noviandy, A. Maulana, G. M. Idroes, R. Suhendra, M. Adam, A. Rusyana, and H. Sofyan, “Deep Learning-Based Bitcoin Price Forecasting Using Neural Prophet,” *Ekonomikalia Journal of Economics*, vol. 1, no. 1, pp. 19–25, 2023. doi: 10.60084/eje.v1i1.51.

[3] D. O. Oyewola, E. G. Dada, and J. N. Ndunagu, “A novel hybrid walk-forward ensemble optimization for time series cryptocurrency prediction,” *Heliyon*, vol. 8, no. 11, 2022. doi: 10.1016/j.heliyon.2022.e11862.

[4] B. Deng, “A Comparative Analysis of Bitcoin Price Forecasting Approaches Using Machine Learning Techniques,” 2025. doi: 10.5220/0013214500004568.

[5] J. Lian, “Comparative Analysis of LSTM, GRU and Transformer Deep Learning Models for Cryptocurrency ZEC Price Prediction Performance,” pp. 396–405, 2024. doi: 10.2991/978-94-6463-408-2_45.

[6] R. S. Andromeda and N. A. S. Winarsih, “Perbandingan Kinerja Metode LSTM dan GRU dalam Prediksi Harga Close Cryptocurrency,” *Sistemasi: Jurnal Sistem Informasi*. [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[7] T. T. Ngoc, L. van Dai, and D. T. Phuc, “Grid search of multilayer perceptron based on the walk-forward validation methodology,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1742–1751, 2021. doi: 10.11591/ijece.v11i2.pp1742-1751.