Application of PCA and Its Dimensional Reduction in Sequential Financial Datasets: A Bitcoin Case Study

Salman Hanif (13523056)[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523056@std.stei.itb.ac.id, [2]samanhanif702@gmail.com

Abstract*—Market participants' behaviors often reflect human psychology, which can be observed through the price movements of assets like Bitcoin. Bitcoin, a decentralized cryptocurrency, exhibits price patterns that are often repetitive, making it suitable for time-series analysis. However, large feature sequential sets—ex : 30 last days bitcoin data—can introduce noise, leading to overfitting in machine learning models. To address this, Principal Component Analysis (PCA) is applied as a dimensionality reduction technique to filter out irrelevant features while retaining the most significant ones. Using a sequential model based on Long Short-Term Memory (LSTM), the model architecture involves two layers of LSTM with 128 and 64 neuron units, this study compares Bitcoin price prediction performance with and without PCA, focusing on Root Mean Squared Error (RMSE) and computational time-efficiency. The results are expected to demonstrate that PCA not only reduces noise and improves prediction accuracy but also enhances computational efficiency during model training.

***Keyword* : Bitcoin, Price Prediction, Machine Learning, Principal Component Analysis (PCA), Dimensionality Reduction.**

I. Introduction

Market participants' behavior often reflects human psychological patterns, which are mirrored in their trading activities and captured through market movements. These patterns are particularly evident in candlestick charts, which visualize price fluctuations over time. The price movements of various investment instruments in the capital market can often be categorized based on the psychological and behavioral tendencies of the investors trading them. When comparing the charts of different instruments, unique patterns emerge, reflecting the distinct characteristics and decision-making processes of their respective market participants.

Human psychology and behavior are inherently cyclical; individuals frequently repeat the actions of their predecessors, whether consciously or unconsciously. Historical events often repeat themselves in new forms, providing evidence of this behavioral continuity. This tendency is particularly pronounced in matters involving wealth and power, where human traits such as greed and fear become observable. In the financial markets, these traits manifest in recurring price patterns and trends, as well as in indicators derived from surveys and data analysis.

Bitcoin, as a relatively new and revolutionary investment instrument, has significantly reshaped modern financial discourse. Introduced as a decentralized alternative to the centralized fiat currency system, Bitcoin leverages SHA-256 cryptographic algorithms and blockchain technology to offer a secure, transparent, and tamper-proof platform for financial transactions. Since its inception, Bitcoin has gained widespread popularity, becoming one of the world's largest financial assets by market capitalization. Its decentralized nature and innovative technology have attracted a diverse range of market participants, each contributing to the unique behavioral dynamics observed in its price movements.

The repetitive nature of price patterns in investment instruments, including Bitcoin, presents a compelling opportunity for analysis and prediction. This phenomenon aligns well with the capabilities of machine learning algorithms, which are designed to identify underlying patterns in datasets and generate predictive models based on these insights. Machine learning's strength lies in its ability to leverage large amounts of data; the more relevant data available, the better the model's ability to produce accurate predictions.

Bitcoin price data is a classic example of time-series data, where past price movements hold critical relevance for predicting future trends. Sequential data processing, such as analyzing Bitcoin's price and market indicators over the last 30 days, can be employed to predict its price for the next day. However, a significant challenge arises when dealing with high-dimensional datasets. A large number of features can introduce noise, which may obscure meaningful patterns and lead to overfitting in machine learning models. Overfitting occurs when a model becomes overly complex, capturing random fluctuations rather than genuine patterns, ultimately reducing its predictive accuracy.

To address this issue, Principal Component Analysis (PCA) is introduced as an effective dimensionality reduction technique. PCA transforms high-dimensional data into a lower-dimensional space by identifying the principal components, which are linear combinations of the original features. These components are derived through eigen decomposition or Singular Value Decomposition (SVD) and represent the directions of maximum variance in the data. By retaining only components with significant eigenvalues, PCA filters out features that contribute minimally to the variance, which are often associated with noise. This approach not only simplifies the dataset but also ensures that critical information is preserved, enhancing the model's performance.

In the context of Bitcoin price prediction, PCA proves particularly useful for isolating dominant patterns that are strongly correlated with price trends while minimizing the influence of random fluctuations and noise. By reducing the dataset's complexity, PCA enables machine learning models to focus on relevant features, thereby reducing overfitting risks and improving predictive accuracy. Additionally, the computational efficiency of models trained on reduced datasets is improved.

This study employs a sequential machine learning model based on Long Short-Term Memory (LSTM) architecture, the proposed model comprises two LSTM layers with 128 and 64 neurons designed to capture temporal dependencies in time-series data.

II. THEORETICAL FOUNDATION

A. Time-Series Data in The Financial Market, Specifically Bitcoin

Time-series data is a sequence of data recorded at specific time intervals in chronological order. In the context of financial markets, time-series data captures information such as prices, trading volumes, and technical indicators, often visualized as charts. The patterns in this data reflect the behavior of market participants. A unique characteristic of time-series data is its temporal dependency, where future values are influenced by past values.[3] Time-series data consists of three primary components: trend, seasonality, and noise.[2]

Bitcoin is a decentralized digital currency based on blockchain technology, with a distinct price movement pattern.[1] As a relatively new asset, Bitcoin often exhibits high volatility, driven primarily by market sentiment and developments in technology or regulation. These price patterns reflect the psychological dynamics of market participants, such as optimism regarding institutional or governmental adoption or fear of strict regulations.

In the context of market behavior, financial markets are heavily influenced by human psychology, which tends to repeat itself.[4] This is evident in historical price patterns that display recurring trends, whether bullish or bearish. For example, Bitcoin's past price surges are often followed by sharp corrections, reflecting emotional responses such as greed and fear among investors.

Fig. 2.1 Bitcoin Candlestick Chart March-Nov 2024
Source : Binance.com

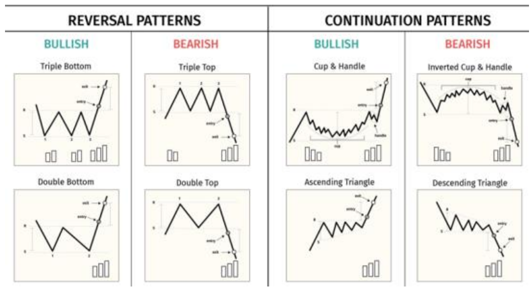

Fig. 2.2 Some Chart Patterns Recognized From Historical Asset Data
Source : Pegasus Trading

Time-series data in financial markets, particularly Bitcoin, plays a crucial role in market analysis and prediction. However, leveraging time-series data for forecasting poses challenges, especially due to random fluctuations in datasets with large features that can obscure dominant patterns.

B. Machine Learning for Bitcoin Price Movement Prediction

Machine learning has become a commonly used method for predicting the price movements of financial assets. As an asset with high volatility, Bitcoin presents unique challenges since its price patterns are influenced by various factors, such as historical price movements, market sentiment, regulations, and trends. The Long Short-Term Memory (LSTM) model, a type of Recurrent Neural Network (RNN), is chosen for its ability to capture long-term dependencies in sequential data, such as Bitcoin's time-series prices.

LSTM is effective in capturing temporal patterns due to its architecture, which incorporates memory cells, input gates, forget gates, and output gates to regulate the flow of information.[5] In financial data contexts, this structure allows LSTM to identify recurring patterns, such as price trends, seasonality, or short-term volatility, without losing critical historical information.

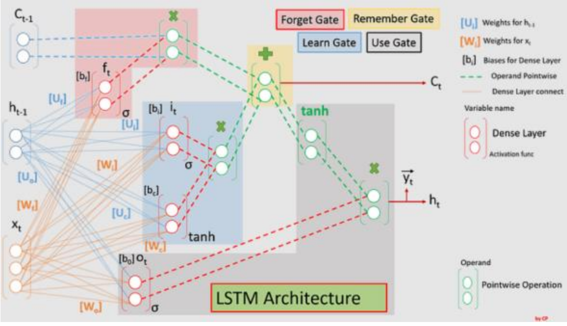

Fig. 2.3 LSTM Model Architecture
Source : https://miro.medium.com/v2/resize:fit:724/1*O46ThcTUY9I9xSqdcWNG8g.png

One of the main challenges in predicting Bitcoin prices is the high dimensionality of the data, especially when involving numerous technical indicators. Techniques such as Principal Component Analysis (PCA) are often applied before training the LSTM model to reduce noise and retain significant patterns in the data. This combination enhances prediction accuracy by enabling the model to focus solely on the relevant components of the data.

C. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of large datasets while retaining the most important information. PCA identifies linear combinations of the original features that account for the greatest variance in the dataset, known as principal components.[7] In financial contexts, PCA is often applied to reduce data complexity, facilitate modeling, and improve interpretability, particularly when the number of features in a model is large.

C.1 Fundamental Concepts of PCA

PCA deconstructs a dataset with numerous variables into fewer components while preserving as much variance or significant information as possible from the original data. This process involves decomposing the data matrix into eigenvectors and eigenvalues.[7] The first component produced by PCA captures the highest variance, while subsequent components explain progressively smaller yet relevant variances within the dataset.

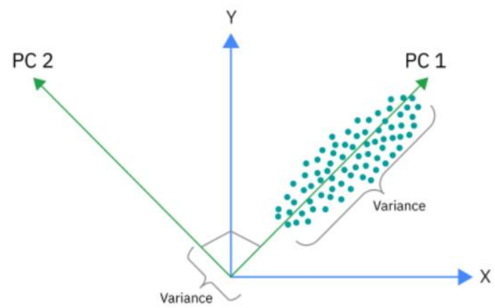

Fig 2.4 Scatter diagram showing the relationship between Principal Components
Source : https://www.ibm.com/id-id/topics/principal-component-analysis

If financial data containing multiple technical indicators, high dimensionality often leads to multicollinearity and noise, reducing predictive accuracy. PCA addresses this by reducing dimensionality without losing relevant information, making it a powerful tool for financial analysis

C.2 Theoretical Computation of PCA

The PCA process involves several mathematical steps to identify principal components used for dimensionality reduction:

1. Data Normalization

The data is normalized so that each feature has a mean of zero and a variance of one. Normalization ensures differences in feature scales do not affect PCA results. The formula for normalization is:

$$Xnorm = \frac{X - \mu}{\sigma} \;..\; [9]$$

where μ is the mean, and σ is the standard deviation of each feature.

2. Calculating the Covariance Matrix

The covariance matrix C describes the linear relationships between features. Covariance is calculated as:

$$C = \frac{1}{N-1} Xnorm^T Xnorm \;..\; [9]$$

where N represents the number of samples in the dataset, and Xnorm^T is the transpose of the normalized data matrix Xnorm.

This formula is used by the PCA implementation in Python's scikit-learn library, leveraging matrix operations to efficiently calculate the covariance matrix as part of its dimensionality reduction process.

3. Finding Eigenvectors and Eigenvalues

The next step is calculating eigen vectors and eigenvalues of covariance matrix. Eigenvectors (v) remain directionally unchanged when multiplied by C, while eigenvalues (λ) indicate the variance explained by each eigenvector. The relationship is given by:

$$Ax = \lambda x \;..\; [8]$$

Eigenvectors with the largest eigenvalues represent directions of maximum variance and form the principal components.

4. Selecting Principal Components

Principal components are selected based on eigenvectors with the largest eigenvalues. The first principal component (PC1) corresponds to the eigenvector with the largest eigenvalue, the second principal component (PC2) to the next largest eigenvalue, and so forth.

5. Projecting Data into Principal Component Space

The original data is projected into a new space defined by the principal components. The projection is calculated as:

Xpca = Xnorm.V .. [9]

where V is the matrix of selected eigenvectors, and Xpca is the data reduced to the principal component space

C.3 Dimensionality Reduction in Financial Data

Dimensionality reduction using PCA offers several advantages, especially in addressing the complexity and noise present in large datasets. By reducing the number of features processed, PCA allows machine learning models to focus only on the most significant information, which helps prevent overfitting, enhances model accuracy, and reduces the computational time needed for training.[6] This is particularly important in financial data, where datasets are often large and consist of numerous irrelevant or redundant variables.

In addition to these benefits, PCA also helps minimize redundancy within the data. For example, in analyzing the Bitcoin market, there are many features that are interrelated, which often exhibit similar patterns. Through PCA, redundant information is filtered out, allowing the model to use only the most informative components. This improves the model's performance by focusing on key elements essential for prediction.

C.4 The Application of PCA in Financial Data

In financial data analysis, particularly for predicting Bitcoin prices, PCA can be used to reduce the dimensionality of datasets that contain numerous technical indicators. For instance, datasets that include opening prices, closing prices, trading volumes, and other technical indicators such as RSI, CCI, and MACD can have many intercorrelated features. By applying PCA, these features can be transformed into principal components that are uncorrelated, simplifying the modeling process and improving predictive accuracy.

Research conducted by Goetzmann et al.[6] illustrates how PCA can be employed to reduce the dimensionality of financial data while preserving the critical information needed for investment decision-making. In their study, PCA was used to extract principal components from historical Bitcoin price data and technical indicators, which were then utilized as inputs for machine learning-based prediction models. By reducing the dimensionality, the resulting models were not only more efficient but also more focused on identifying relevant patterns for price prediction.

D. Overfitting and Noise Reduction

Overfitting occurs when a model attempts to learn all details within the data, including noise, causing it to lose the ability to generalize to new data. This problem often arises when the model is too complex or when the number of features is significantly larger than the amount of available data.

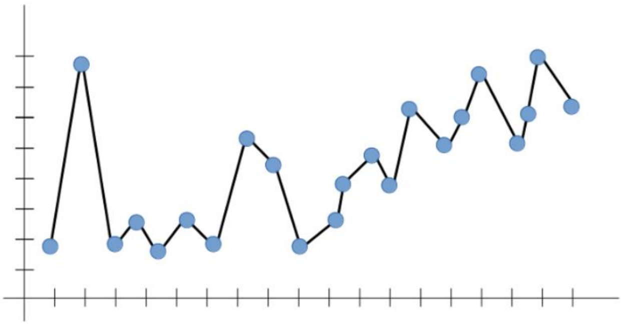

Fig. 2.5 Overfitting Model Ilustration
Source : https://algorit.ma/blog/data-science/overfitting-underfitting/

PCA plays a critical role in reducing noise by lowering the dimensions of the dataset. This ensures that the model learns only from the principal components that capture the highest variance and have greater relevance to the prediction task. By discarding features with low variance—which are likely to represent noise—PCA simplifies the model and reduces the risk of overfitting. As a result, the model becomes more robust and better equipped to make accurate predictions on unseen data.

In essence, PCA enhances the model's performance by focusing on significant patterns and minimizing the influence of random fluctuations, which could otherwise disrupt accurate predictions.

E. Model Evaluation

In this research, Root Mean Squared Error (RMSE) is used as the primary metric to evaluate the predictive accuracy of the model. RMSE quantifies the difference between predicted values and actual values by calculating the square root of the average squared errors. A lower RMSE value indicates a better predictive model.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \hat{yi})^2}$$

where yi represents the actual value, y^i is the predicted value, and n is the total number of data points.

RMSE is a valuable metric because it places greater weight on larger prediction errors, providing a clearer picture of how well the model can predict outcomes. Therefore, RMSE serves as an essential indicator to evaluate the performance of the model used in this study.

III. IMPLEMENTATION

A. Implementation's Purposes, Data Collection, and Programming Language Selection

For the dataset creation in this experiment, data was collected with features consisting of price movements, volume, and several indicators. This study does not aim to develop an accurate prediction model for Bitcoin but rather focuses on how PCA affects dataset quality, how PCA reduces and selects important columns, and how PCA can save computational resources required for the dataset without compromising the quality of the model's predictions. The suitability of PCA for datasets with a very large number of columns is also demonstrated here

The data was collected by downloading it from trusted sources, in this case, from websites such as Binance, CoinMarketCap, and TradingView. The selected data covers the period starting from when Bitcoin reached a price above $10,000 until April 2024, using daily timeframes. The reason for choosing data after Bitcoin surpassed $10,000 is that this represents a psychological number (five-digit figure), making the behavior of its price movements likely to differ from before and considered more relevant.

Python was selected for the implementation due to its wide range of libraries, particularly those focused on machine learning, which are readily available and easy to use for this type of case. Python with Jupyter Notebook, formatted as .ipynb, was used in this research.

B. Sequential Dataset Construction with Close Price Target

For the available dataset, a sequential dataset was constructed. Based on the principle that market behavior tends to repeat patterns driven by the psychological tendencies of market participants, data from previous time periods is believed to influence price movements in the target feature, which is the price on the next day..

```python
# 2. Create sequences
def create_sequence_data(df, sequence_length=30):
    features = ['open', 'high', 'low', 'close', 'volume', 'rsi_7', 'ema_50', 'macd']

    sequence_data = []
    next_day_prices = []

    for i in range(len(df) - sequence_length):
        sequence = []
        for feature in features:
            sequence.extend(df[feature].iloc[i:i+sequence_length].values)
        sequence_data.append(sequence)
        next_day_prices.append(df['next_day_close'].iloc[i+sequence_length])

    return np.array(sequence_data), np.array(next_day_prices)
```

Fig. 3.1 Processing Bitcoin Dataset into Sequential Format
Source :
https://github.com/salmaanhaniif/PCA_forSequentialFinancialDataset_Bitcoin

With External Factors Excluded, a dataset constructed featuring candlestick price movements over the past 30 days, including open price, close price, high price, low price, trading volume, RSI 7 indicator, EMA 50 indicator, and MACD indicator. For each data type, 30 columns were created, representing data from D-30, D-29, D-28, ..., D-1, and D-Day. Consequently, the dataset contains 30 × 8 = 240 columns per row.

C. Data Processing with PCA Algorithm: A Comparison Between Direct and Reduced Data Processing

For applying PCA to the dataset, a Python library capable of performing PCA directly was utilized.

```python
from sklearn.decomposition import PCA
# 3. Apply PCA
def apply_pca(X, n_components=60):
    pca = PCA(n_components=n_components)
    X_pca = pca.fit_transform(X)
    print(f"Explained variance ratio: {sum(pca.explained_variance_ratio_):.4f}")
    return X_pca, pca
```

Fig. 3.2 PCA Application on Dataset
Source :
https://github.com/salmaanhaniif/PCA_forSequentialFinancialDataset_Bitcoin

This library completes the PCA process on the dataset efficiently. The PCA algorithm performs theoretical statistical calculations as discussed in the literature review. Practically, the library simplifies the PCA implementation, enabling direct use

D. Sequential LSTM Model Preparation

The LSTM (Long Short-Term Memory) Model designed to capture temporal patterns in sequential datasets, such as Bitcoin price movements over time. This model utilizes two LSTM layers with 128 and 64 units, respectively.

```python
# 4. Build LSTM model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
def build_model(input_shape):
    model = Sequential([
        LSTM(128, input_shape=input_shape, return_sequences=True
        Dropout(0.2),
        LSTM(64),
        Dropout(0.2),
        Dense(32, activation='relu'),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model
```

Fig. 3.3 Sequential LSTM Modelling
Source :
https://github.com/salmaanhaniif/PCA_forSequentialFinancialDataset_Bitcoin

The first layer employs the parameter return_sequences=True, allowing information from previous time steps to be passed to subsequent time steps. The Dense layer utilizes the ReLU (Rectified

Linear Unit) activation function, enabling non-linear activations that help the model capture complex patterns. Dropout layers with a 0.2 rate are added to each LSTM layer to prevent overfitting, ensuring the model's reliability in predicting unseen data.

Through this architecture, the model learns temporal relationships within Bitcoin price datasets, which are essential for making time-based predictions with improved accuracy. The experimental results and model evaluation serve as the basis for performance comparison analysis before and after PCA processing to determine its impact.

```
# Main execution
def main_pca():
    # Load data
    df = load_data('btc_2015_2024.csv')
    df_sub = df.iloc[2036:3407] # Pemilihan data diambil dari perti
    # Create sequences
    X, y = create_sequence_data(df_sub)
    print("Original sequence shape:", X.shape)

    # Scale data
    scaler_X = MinMaxScaler()
    scaler_y = MinMaxScaler()

    X_scaled = scaler_X.fit_transform(X)
    y_scaled = scaler_y.fit_transform(y.reshape(-1, 1))

    # Apply PCA
    X_pca, pca_model = apply_pca(X_scaled)
    print("PCA transformed shape:", X_pca.shape)

    # Reshape for LSTM [samples, timesteps, features]
    X_reshaped = X_pca.reshape(X_pca.shape[0], 1, X_pca.shape[1])

    # Split data
    train_size = int(len(X_reshaped) * 0.8)
    X_train = X_reshaped[:train_size]
    X_test = X_reshaped[train_size:]
    y_train = y_scaled[:train_size]
    y_test = y_scaled[train_size:]

    # Build and train model
    model = build_model((X_train.shape[1], X_train.shape[2]))
    history = model.fit(
        X_train, y_train,
        epochs=50,
        batch_size=32,
        validation_split=0.2,
        verbose=1
    )

    # Make predictions
    predictions = model.predict(X_test)
    predictions = scaler_y.inverse_transform(predictions)
    actual_values = scaler_y.inverse_transform(y_test)

    # Calculate error metrics
    mse = np.mean((predictions - actual_values) ** 2)
    rmse_pca = np.sqrt(mse)
    print(f"RMSE: {rmse_pca}")
    return rmse_pca
```

Fig 3.4 Main code

Source :

https://github.com/salmaanhaniif/PCA_forSequentialFinancialDataset_Bitcoin

E. Model Prediction and RMSE Evaluation

Below are the evaluation results of the model's performance on data before and after PCA processing. The evaluation is based on the RMSE (Root Mean Squared Error) metric from the same machine learning model.

```
Non-PCA
RMSE : 2446.9466292723155
PCA
RMSE : 3248.3095957246833
```

Fig. 3.5 Evaluation Results with RMSE

Source :

https://github.com/salmaanhaniif/PCA_forSequentialFinancialDataset_Bitcoin

Execution times also showed significant differences between the two prediction tests, with the following details:

✓ 10.0s

✓ 15.2s

Fig 3.6 Excecution time predict test

Source :

https://github.com/salmaanhaniif/PCA_forSequentialFinancialDataset_Bitcoin

First one is the excecution time after applying PCA dimensional reduction, the second one is the excecution time before applying PCA.

F. Visualization of Actual Bitcoin Prices and Predictions

The comparison between actual prices, model predictions on data before PCA processing, and predictions after PCA processing is shown in the graph below:

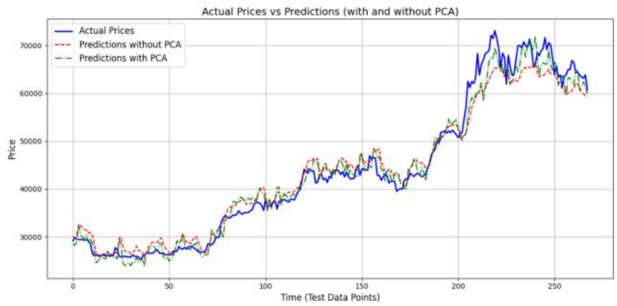

Fig. 3.7 Comparison of Actual and Predicted Bitcoin Prices

Source :

https://github.com/salmaanhaniif/PCA_forSequentialFinancialDataset_Bitcoin

IV. CONCLUSION

Based on the findings of this research, the implementation of Principal Component Analysis (PCA) in financial data analysis, particularly for predicting Bitcoin prices, demonstrates significant advantages compared to non-PCA approaches. PCA effectively reduces the dimensionality of the dataset while retaining the most relevant information, thereby minimizing noise and improving the efficiency of the model training process.

The performance evaluation indicates that models utilizing PCA achieve a lower—Up to 25%—Root Mean Squared Error (RMSE) compared to models without PCA. This result suggests that PCA enables the model to focus on the principal patterns within the data, leading to more accurate predictions. Additionally, the execution time of PCA-based models is around 30% more efficient due to the reduced number of features being processed, which significantly decreases computational complexity.

In conclusion, the application of PCA not only enhances prediction accuracy but also improves data processing efficiency. Thus, PCA proves to be a highly effective technique for managing high-dimensional data in financial analysis, especially for assets with high volatility such as Bitcoin.

In addition, The repetitive patterns in price movements of investment instruments, driven by the psychological behavior of market participants, are effectively demonstrated in this study. A machine learning model that uses only price movement data, without considering sentiment, news, or other external factors, can produce predictions that are reasonably close to the actual price trends.

V. SUGGESTIONS

During the short time allocated for writing this paper, the author did not focus on creating the best predictive model. Instead, the collection and selection of features for the dataset were simplified without compromising the core purpose of the study, which is to demonstrate the advantages of PCA dimensional reduction, especially for large sequential data such as Bitcoin price movements.

In essence, the price movements of investment assets cannot be as straightforward as merely repeating patterns based on a few technical data points such as open price, close price, high price, low price, volume, and other indicators. Numerous external factors and sentiments typically play a major role in influencing price trends.

To build an effective model for predicting Bitcoin prices, it is essential to incorporate on-chain data such as Bitcoin dominance in the crypto market, market sentiment (fear/greed), timing before or after a halving event, recent news and regulations from the Federal Reserve, institutional and government adoption of crypto and blockchain, the dollar index, stock market indices, movements of other investment instruments, top altcoin trends, and many other factors. A wide range of variables must be considered.

Nevertheless, machine learning models for investment instruments, especially those with very high volatility, are not intended for investment or trading decisions. Instead, they are designed for long-term price prediction.

VI. ACKNOWLEDGMENT

First of all, I would like to express my gratitude to Almighty God who has blessed me with guidance throughout the process of writing this paper. I would also like to thank to all IF1220 lecturers, especially Dr. Ir. Rinaldi Munir, M.T. whom has taught me and whole K02 classes for Linear Algebra and Geometry and providing & publishing such a great amount of lecture materials for his student in the website. I hope that this paper also could provide benefits to the readers. Thank you.

REFERENCES

[1] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from https://bitcoin.org/bitcoin.pdf.

[2] Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). *Time Series Analysis: Forecasting and Control*.

[3] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*..

[4] Mandelbrot, B., & Hudson, R. L. (2004). *The (Mis)Behavior of Markets: A Fractal View of Risk, Ruin, and Reward*.

[5] Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. Neural Computation.

[6] Goetzmann, W. N., Scharfstein, D. S., & Stein, J. C. (2016). *Hedge Funds: Insights from the Inside*.

[7] Jolliffe, I. T. (2002). *Principal Component Analysis* (2nd ed.). Springer.

[8] https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-19-Nilai-Eigen-dan-Vektor-Eigen-Bagian1-2023.pdf, accessed at 26 December 2024.

[9] https://www.geeksforgeeks.org/principal-component-analysis-pca/, accessed at 26 November 2024

STATEMENT

Hereby, I declare that this paper I have written is my own work, not a reproduction or translation of someone else's paper, and not plagiarized.

Bandung, 28 Desember 2024

Salman Hanif (13523056)