Published online on the page: https://jurnal.ilcagam.org/index.php/jsti

J S T I

Jurnal Serumpun Teknik Informatika

| ISSN (*Online*) 3123-965x | DOI: https://doi.org/10.66485/jsti.v1i2.21

Predicting Bitcoin Price Trends Using an LSTM Model Based on Multi-Variable Technical Indicators

Rizky Parlika[1,], Ilham Asy'ari[2], Rizky Ananda Ramadhan[3]*

[1,3]Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia
[2]Universitas Persada Bunda, Pekanbaru, Indonesia

Article Information

Article History:
Submit: 26 Maret 2026
Accepted: 25 April 2026
Published: 30 April 2026

Keywords

Bitcoin; Deep Learning; LSTM; Price Prediction; Technical Indicators

Correspondence

E-mail: iamasyari@gmail.com*

ABSTRACT

The sharp price fluctuations in the cryptocurrency market, particularly in Bitcoin (BTC), create significant risks while simultaneously offering speculative profit potential for investors. Traditional analytical methods are often ineffective in detecting non-linear patterns present in stochastic financial time series data. This study proposes the application of a Deep Learning model utilizing the Long Short-Term Memory (LSTM) architecture to project the directional trend of Bitcoin prices (whether upward or downward) for the upcoming one-hour period. In the model's development, historical price data is integrated with a set of crucial technical variables, including the Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Exponential Moving Average (EMA), which serve as input attributes to enhance accuracy. Market data is retrieved in real-time via the Binance API, covering the last 1000 candlesticks. Experimental results using a Stacked LSTM architecture demonstrate that the model achieves an accuracy rate of 51.08% on the test data. Although this classification accuracy is considered moderate, a simple backtesting simulation indicates a positive profitability potential of 2.88% with a win rate of 48.39%. The output of this research also includes a web-based system prototype that integrates a Python backend with a visual interface for real-time monitoring of prediction signals.

1. Introduction

Over the past ten years, cryptocurrency has evolved from a mere technological experiment into a major global investment instrument. As the digital asset with the highest market capitalization, Bitcoin (BTC) is often used as the primary barometer for determining overall market trends (Nakamoto, 2008). However, the highly volatile nature of the cryptocurrency market, its 24/7 operation, and the lack of centralized regulation make price forecasting a far more complex challenge than in conventional stock markets (Ningsih et al., 2024; Saadah & Salsabila, 2021). Wibowo dan Somya (2023) It should be noted that this complexity requires the use of more advanced computational methods than standard statistical analysis alone. A number of recent studies have shown that deep learning algorithms such as LSTM and GRU are capable of delivering superior performance compared to traditional methods when analyzing dynamic cryptocurrency time series data (Andromeda et al., 2025).

Traditional technical approaches that rely on chart interpretation and mathematical formulas tend to be highly subjective. Meanwhile, classical statistical models often face challenges when mapping nonlinear correlations in financial data that contains a lot of noise (Hiskiawan et al., 2025). Therefore, methods based on artificial intelligence, particularly deep learning, are emerging as a promising solution. Long Short-Term Memory (LSTM), a variant of Recurrent Neural Networks (RNN), has proven effective at processing sequential data while addressing the vanishing gradient problem. (Nirraca & Hartati, 2024; Sujjada et al., 2024; Tamami & Arifin, 2024).

Previous research in Indonesia has demonstrated the effectiveness of LSTM in predicting cryptocurrency prices. Irawan and Utami (2025) For example, developing a hybrid model that successfully improves the accuracy of Bitcoin predictions. Similarly, Saepulrohman and Pratama (2023) applied a specific modification of the LSTM method to project price direction, yielding quite significant results. On the other hand, Rizkilloh (2022) highlights the robustness of LSTM in handling high volatility. Furthermore, the integration of technical indicators such as RSI and MACD has also proven effective in confirming crypto market trends (Siregar, 2025; Talahaturuson et al., 2022). This study aims to combine the Stacked LSTM architecture with multivariate features (OHLCV and technical indicators) for short-term price direction classification, an approach considered more practical for day traders (Corrs et al., 2025).

2. Research Methods

2.1 System Flowchart

This forecasting system is designed with a systematic and integrated workflow, covering stages ranging from the acquisition of real-time market data and the extraction of technical features to the visualization of forecast results for users (Sidiq & Nurzaman, 2024). This design was chosen to minimize latency, ensuring that the entire process runs automatically without repeated manual intervention. This automation is crucial for the system to operate in real time and adapt to the fast-paced dynamics of the crypto market. Details of the system's operational logic can be seen in the Flowchart (Figure 1).

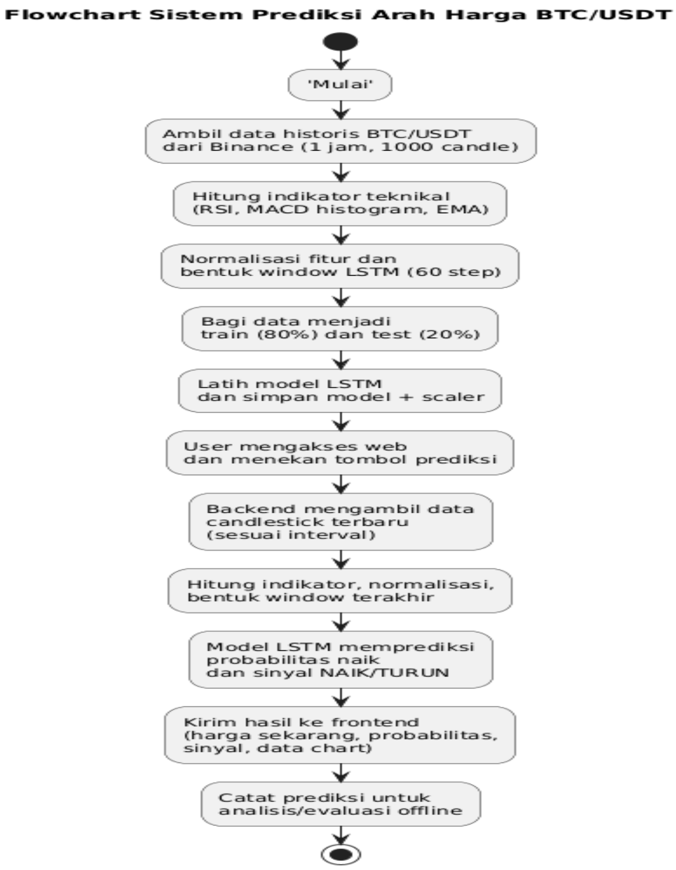

Figure 1. Prediction System Flowchart

The process begins with the system automatically fetching the latest candlestick data via the Binance API. This raw data is then processed to calculate technical indicators (RSI, MACD, EMA) and normalized. The prepared data is then fed into a trained LSTM model to generate price direction probabilities. Finally, the prediction results are sent to the web interface to be displayed as charts and trading signals.

2.2 Web-Based System Design Method

This system, developed as a web application, employs a structured approach tailored to the needs of time series analysis. The integration of the LSTM algorithm into the web platform has proven to make it easier for users to monitor predictions in real time (Sidiq & Nurzaman, 2024). The development phases include data requirements analysis, Deep Learning architecture design, code implementation (Python & Web), and system testing and validation (Juventauricula et al., 2024).

2.3 Data Collection

The research data is sourced from Binance's public API (Binance, 2024). The asset used is the BTC/USDT pair with a 1-hour timeframe. The system is designed to automatically retrieve the last 1,000 candlesticks at runtime, including the open, high, low, close, and volume. The use of an API allows the system to always operate based on the latest price data (Kristianti & Pranatawijaya, 2023).

Data retrieval is limited to the last 1,000 candlesticks (approximately 41 days for a 1-hour interval) due to practical limitations on a single Binance API request without advanced pagination mechanisms. This limitation affects the scope of the training data, which may not fully represent the entire market cycle (such as complete bull and bear market phases), potentially impacting the model's ability to generalize to different market conditions.

2.4 Feature Engineering

To enrich the model's information, technical indicators calculated from closing price data were added. Sembiring et al. (2021) emphasizes that moving average-based indicators are very helpful in identifying price trends in digital assets.

1. RSI: A momentum analysis tool used to assess the speed and magnitude of price changes. The RSI is used to identify market conditions that are either overbought or oversold (Wilder, 1978). These indicators are often combined in crypto sentiment analysis to validate trends (Parlika et al., 2020). Calculated over a 14-day period.
2. MACD Histogram: Measures the distance between the MACD line (the difference between the 12-period EMA and the 26-period EMA) and the signal line (9-period EMA), which is effective for detecting trend strength (Appel & Gerald, 2005).
3. EMA: An exponential moving average that places greater weight on recent data, calculated using a 20-period setting to capture short-term trends (Edwards et al., 2007).

2.5 Data Preprocessing

Time-series data is scale-sensitive. Therefore, all features are normalized using the Min-Max Scaling technique to the range [0, 1] so that the gradient convergence process during model training runs more efficiently (Goodfellow et al., 2016).

The dataset was created using the Sliding Window method. The window size was set to 60, meaning the model uses data from the last 60 hours to predict the price direction at the 61st hour. The target label (y) is binary:

1) y=1 (Up): If Close(t+1) > Close (t)

2) y=0 (Down): If Close(t+1) ≤ Close(t)

The entire sample dataset is divided chronologically, with 80% used for model training (740 samples) and the remaining 20% as test data (186 samples). Data is split chronologically to avoid information leakage from future data (look-ahead bias), as explained by Lopez de Prado (2018).

2.6 Technology Implementation

This decision support system is web-based and uses a client-server architecture. The backend uses the Python programming language with the FastAPI framework to handle business logic, deep learning model training (TensorFlow/Keras), and API communication. The frontend is built using HTML/CSS/JavaScript for interactive graphical visualization. Data exchange between the frontend and backend is conducted via a REST API.

2.7 LSTM Model Architecture

The model was built using the TensorFlow/Keras framework with a Stacked LSTM architecture (Gers et al., 2000):

1. Input Layer: Dimensions (60, 4), representing 60 time steps and 4 features.
2. LSTM Layer 1: 64 neurons, return sequences=True.
3. Dropout Layer 1: Ratio 0.2 to prevent overfitting (Srivastava et al., 2014).
4. LSTM Layer 2: 32 neurons.
5. Dropout Layer 2: Ratio of 0.2.
6. Output Layer: Dense layer with 1 neuron and a Sigmoid activation function to produce class probabilities (0–1).

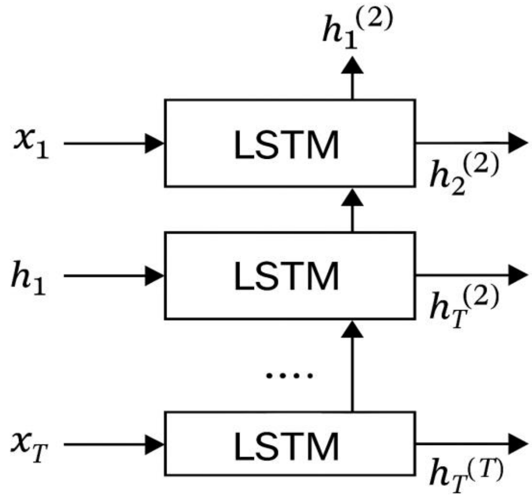

Figure 2. Prediction System Flowchart

The model was trained using the Adam optimizer (Kingma & Ba, 2015) using an adaptive learning rate and the Binary Cross-Entropy loss function. Training was conducted over 50 epochs with a batch size of 64.

3. Results and Discussion

3.1 Implementation and Interface Results

The web application developed features a main interface module to help users monitor price movements and predictive signals.

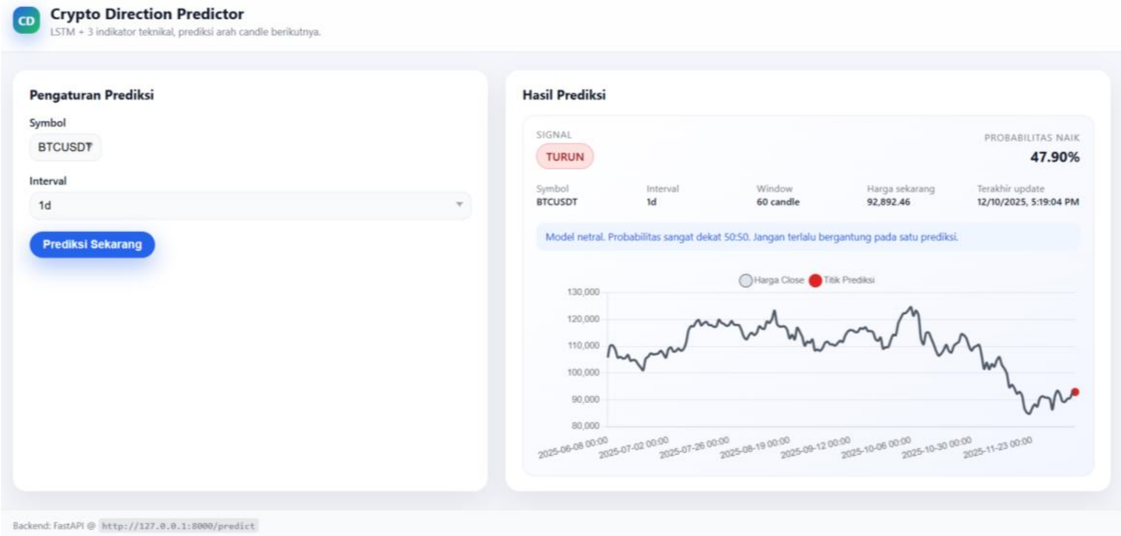

Figure 3. System Main Interface

In Figure 3, the app's main interface is divided into two main panels. The left panel, "Prediction Settings," allows users to select an asset pair (e.g., BTC/USDT) and a time interval (e.g., 1d) before starting the prediction process. The right panel, 'Prediction Results,' presents the model's output comprehensively, including directional signal indicators (Up/Down) and the probability percentage of an increase. Below the main indicators, there is meta information such as the window size and current price, followed by a Confidence Banner—a blue message box that provides contextual advice based on the model's confidence level (e.g., a warning when the model is neutral). The bottom of the panel displays an interactive line chart that visualizes historical closing price movements (Close Price) and marks prediction points (Prediction Points) to provide users with a visual representation of future trends.

3.2 Model Training Results

The LSTM model was trained using 740 sample windows (80% of the total 926 windows) and validated on the remaining 186 sample windows as test data. The training process was conducted over 50 epochs with a batch size of 64, using the Adam optimizer and the Binary Cross-Entropy loss function, as described in the Methods section.

Overall, the training loss decreased significantly during the first few epochs before eventually stabilizing, while the validation loss showed a similar trend. This indicates the absence of excessive overfitting. Additionally, the training and validation accuracies were at relatively similar levels, indicating that the model was able to recognize patterns from the training data well without losing its ability to generalize to the test data.

The trained model and scaler are saved as separate files so they can be reused in the offline evaluation phase or in real-time web services. This supports a system design that integrates the modeling module and the user interface.

3.3 Model Performance Evaluation Results

The evaluation was performed on the test dataset (the last 186 hours). The results of the classification matrix evaluation are presented in Table 1.

Table 1. Classification Results

Class	Precision	Recall	F1-Score	Support
Down (0)	0,71	0,05	0,10	94
Up (1)	0,50	0,98	0,66	92

Macro avg	0,61	0,52	0,38	186
Weighted avg	0,61	0,51	0,38	186

The results show that the model has very high sensitivity (Recall) for predicting price increases (0.98), but is weak at predicting price decreases. This indicates a bullish bias in the model, likely caused by the predominantly positive market trend during the training data period. This bias is consistent with Febriansyah's findings (Sujjada et al., 2024) which states that single LSTM models often struggle to capture minority patterns in imbalanced data without an attention mechanism or data balancing.

3.3 Classification Evaluation Results

Classification evaluation was performed on 186 samples from the test data window. The evaluation process began by calculating the probability of the "Up" class (sigmoid output) for each sample, then applying several thresholds around 0.5. Testing showed that a threshold of 0.5 yielded the best accuracy on the test data, at 51.08%.

The model's prediction distribution against the actual classes can be seen in the following confusion matrix:

1) Actual Down (0): 5 samples were classified as Down, 89 samples were classified as Up.
2) Actual Up (1): 2 samples were classified as Down, 90 samples were classified as Up.

Table 2. Confusion matrix showing model performance on the test data

Current / Forecast	Forecast: Down (0)	Forecast: Up (1)
Current Down 0	5	89
Current Up 1	2	90

In Table 2, the high number of false positives (89 cases) indicates that the model tends to generate "Buy" (Up) signals even when prices are actually falling. This suggests a risk of over-optimism in the model, which could potentially result in incorrect entry signals during bearish market conditions. Conversely, the very low number of False Negatives (2 cases) indicates that the model rarely misses actual upward price momentum. The practical implication is that traders are advised to use the "UP" signal from this model as an initial filter, but still require additional confirmation from other indicators to validate entry positions in order to mitigate the risk of False Positives.

This "Up" prediction bias is most likely a consequence of the characteristics of the dataset used. By limiting data collection to only the last 1,000 candlesticks, the training sample available to the model is heavily influenced by the market trends occurring within that time window. If those 1,000 candles are dominated by a bullish period or a consistent price increase, the LSTM model will tend to learn that an upward trend is the most common and safe pattern to predict (Dutta et al., 2020). As a result, the model struggles to generalize when faced with bearish or sideways market conditions, leading to a high error rate in the "Down" class.

3.4 Signal Analysis Results (Backtesting)

To examine the practical implications of the model's signals, a simple trading simulation was conducted using a test dataset consisting of 186 window samples. The strategy employed was to open a long position at the current closing price if the model predicted "Up" and close it at the closing price of the next candle. If the model predicted "Down," no position was opened (wait and see).

1) Total Return: +2.88%
2) Win Rate: 48.39%

Although the statistical accuracy is moderate and the Win Rate is below 50%, the simulation shows a potential positive profitability of 2.88% during the testing period. This aligns with the principle that in trading strategies, the risk-reward ratio (the magnitude of profit when winning vs. loss when losing) is often more important than mere accuracy in predicting direction alone (Sadewo et al., 2025). In other words, the model successfully captured periods of significant price increases even though its winning frequency was not particularly high.

4. Conclusion

This study successfully designed and implemented a web-based Bitcoin price direction prediction system using the LSTM Deep Learning method integrated with three major technical indicators: RSI, MACD, and EMA. Based on the test results, the developed model was able to predict the direction of Bitcoin price movements (Up/Down) for the next hour with an accuracy of 51.08% on the test data. However, evaluation using a Confusion Matrix revealed a significant prediction bias toward the majority class ("Up"), with a Recall value of 0.98 for the 'Up' class, while the "Down" class had only 0.05. This indicates that the model tends to be overly optimistic due to the training data being dominated by an upward price trend. Although the accuracy of directional predictions is not particularly high, a simple backtesting simulation yielded positive profitability potential with a Total Return of +2.88% and a Win Rate of 48.39% during the testing period, indicating that the model is capable of capturing significant upward price momentum and providing added value as a decision support tool. Furthermore, the implementation of the system as an interactive web application has proven effective in presenting real-time prediction information, making it easier for users to monitor trading signals and market price movements.

.

References

Andromeda, R. S., Anisa, N., & Winarsih, S. (2025). *Perbandingan Kinerja Metode LSTM dan GRU dalam Prediksi Harga Close Cryptocurrency Performance Comparison of LSTM and GRU Methods in Predicting Cryptocurrency Closing Prices*. *14*, 366–379. https://doi.org/10.32520/stmsi.v14i1.4880.

Appel, & Gerald. (2005). *Technical Analysis: Power Tools for Active Investors*. FT Press.

Binance. (2024). *Binance API documentation*.

Corrs, A. A., Syam, A., & Aris, V. (2025). *Analisis Sentimen Tren Cryptocurrency Menggunakan Machine Learning*. *3*(4). https://doi.org/10.31004/riggs.v3i4.486

Dutta, A., Kumar, S., & Basu, M. (2020). *A Gated Recurrent Unit Approach to Bitcoin Price Prediction*. *Poyser 2017*. https://doi.org/10.3390/jrfm13020023

Edwards, R. D., Magee, J., & Bassetti, W. H. C. (2007). *Technical Analysis of Stock Trends* (9th ed.). CRC Press.

Gers, Schmidhuber, F. A., Cummins, J., & Fred. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10). https://doi.org/10.1162/089976600300015015

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Hiskiawan, P., William, J., Feliepe, L., & Jansel, T. (2025). *A Hybrid Data Science Framework for Forecasting Bitcoin Prices using Traditional and AI Models*. *9*(5), 2089–2101. https://doi.org/10.30871/jaic.v9i5.10631

Irawan, R., & Utami, E. (2025). *Prediksi Harga Bitcoin Menggunakan Model Hibrida Transformer Dan Long Short Term Memory Magister of Informatics , Amikom University , Yogyakarta , Indonesia Bitcoin Price Prediction Using a Hybrid Transformer and Long Short-Term Memory Model*. *5*(9), 2866–2877. https://doi.org/10.52436/1.jpti.1091

Juventauricula, P., Hanggara, B. T., & Pramono, D. (2024). *Pengembangan Sistem Informasi Point of Sale (POS) berbasis Web menggunakan Pendekatan Metode Waterfall (Studi Kasus : Restoran Altari)*. *8*(1), 97–106.

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 1–15.

Kristianti, N., & Pranatawijaya, V. H. (2023). *PEMANFAATAN API INDODAX PADA PERDAGANGAN ASET DIGITAL CRYPTOBERBASIS WEBSITE*. *17*(1), 49–57. https://doi.org/10.47111/jti.v17i1.6709

Lopez de Prado, M. (2018). *Advances in Financial Machine Learning*. Wiley. https://doi.org/10.2139/ssrn.3365271

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system [White paper]*.

Ningsih, S. R., Wardhani, R. S., & Anggita, W. (2024). *Pengaruh Bitcoin dan Nilai Tukar Rupiah Terhadap Harga Saham (Studi Pada Sub Sektor Perbankan Yang Terdaftar di Bursa Efek Indonesia Periode 2020-2022)*. *5*(2), 1029–1035. https://doi.org/10.55681/jige.v5i2.2681

Nirraca, M., & Hartati, E. (2024). *Prediksi harga bitcoin menggunakan metode long short term memory*. *07*(01), 55–65. https://doi.org/10.32502/digital.v7i1.7974

Parlika, R., Pradika, S. I., Hakim, A. M., & M, K. R. N. (2020). *Analisis Sentimen Twitter Terhadap Bitcoin dan Cryptocurrency Berbasis Python TextBlob*. *2*, 33–37. https://doi.org/10.33005/jifti.v2i2.22

Rizkilloh, M. F. (2022). *Prediksi harga cryptocurrency menggunakan algoritma Long Short Term Memory (LSTM)*. Universitas Islam Indonesia.

Saadah, S., & Salsabila, H. (2021). Prediksi harga Bitcoin menggunakan metode Random Forest. *Jurnal Komputer Terapan*, *7*(1), 24–32. https://doi.org/10.35143/jkt.v7i1.4618

Sadewo, B., Rahman, H., & Syah, D. H. (2025). *Designing an Automated Trading System Based on Oscar ' s Grind Money Management with Bollinger Bands and ADX for Trading XAU / USD on the MetaTrader 5 Platform*. *4*(4), 616–625.

Saepulrohman, A., & Pratama, S. Z. (2023). *Prediksi Arah Harga Bitcoin Berdasarkan Manipulasi Metode Long Short-Term Memory (LSTM)*. *3*(1), 15–24. https://doi.org/10.33751/interval.v3i1.7290

Sembiring, F., Gustian, D., Erfina, A., & Vikriansyah, Y. (2021). Analisis tingkat akurasi algoritma Moving Average dalam prediksi pergerakan uang elektronik Bitcoin. *Prosiding Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri (SENIATI)*. https://doi.org/10.35889/jutisi.v10i1.577

Sidiq, M. A., & Nurzaman, F. (2024). Rancang Bangun Aplikasi Untuk Prediksi Harga Bitcoin Menggunakan Algoritma Long Short-Term Memory. *IKRA-ITH Informatika: Jurnal Komputer Dan Informatika*, *9*(2), 29–36. https://doi.org/10.37817/ikraith-informatika.v9i2.4387

Siregar, S. M. (2025). *Analisis teknikal saham menggunakan metode MACD, RSI, dan Stochastic Oscillator*. Politeknik Negeri Sriwijaya.

Sujjada, A., Sembiring, F., & Febriansyah. (2024). *Prediksi Harga Bitcoin Menggunakan Algoritma Long ShortTerm Memory*. 450–459.

Talahaturuson, E. Y., Junaedi, L., & Gumelar, A. B. (2022). *Analisis Sentimen Pergerakan Harga Mata Uang Kripto (Cryptocurrency) Menggunakan TextBlob- NLTK (Natural Language Toolkit)*. *5*(2), 44–50.

Tamami, G., & Arifin, M. (2024). *Penggunaan LSTM dalam Membangun Prediksi Penjualan untuk Aplikasi Laptop Lens*. *14*(2), 301–308. https://doi.org/10.37859/jf.v14i2.7372

Wibowo, Y. D., & Somya, R. (2023). *Penerapan Data Mining Untuk Memprediksi Harga Cryptocurrency Ethereum Menggunakan Metode Backpropagation Neural Network*. *10*(1), 273–286.

Wilder, J. W. (1978). *New concepts in technical trading systems*. Trend Research.