Research Article

LSTM-Based Forex Trading Bot Using Python and MetaTrader 5: Design, Simulation, and Evaluation

[1,*] Reymark-John A. Macapanas, [2]Mary Ann Gliefen Bermudo

[1] University of Science and Technology of Southern Philippines (Villanueva Campus), Faculty, Villanueva, Philippines;
[2] Mindanao State University Iligan Institute of Technology, Iligan City, Philippines

* Corresponding Author: reymarkjohn.macapanas@ustp.edu.ph

Abstract: This paper presents the development of an AI-driven forex trading bot that utilizes a Long Short-Term Memory (LSTM) neural network to forecast short-term price movements and automate trading decisions. The objective of the study is to create a scalable, data-driven system capable of improving trade accuracy using historical USD-JPY price data in conjunction with the MetaTrader 5 platform. The proposed system integrates a time-series preprocessing pipeline, LSTM-based price prediction, and a logic-driven trade simulation model to assess performance under controlled conditions. The model achieved a directional accuracy of 88.4%, a profit accuracy of 78%, and a cumulative simulated profit of USD 797.50 over 100 trades. Additionally, training and validation losses stabilized after 50 epochs, indicating effective learning without overfitting. Visual comparisons between actual and predicted prices further validated the model's forecasting ability. The results highlight the potential of LSTM models to support intelligent financial automation and provide a foundation for future enhancements, including real-time deployment and hybrid AI-based trading strategies.

Keywords: LSTM trading bot, forex forecasting, MetaTrader 5, time-series prediction, algorithmic trading, financial automation.

Citation: Macapanas, R.-J. A., & Bermudo, M. A. G. (2025). LSTM-based Forex trading bot using Python and MetaTrader 5: Design, simulation, and evaluation. Iota, 5(3). https://doi.org/10.31763/iota.v5i3.971

Academic Editor: Adi, P.D.P

Received: June 12, 2025

Accepted: July 04, 2025

Published: August 01, 2025

1. Introduction

The global foreign exchange (forex) market represents the largest and most liquid financial marketplace in the world, with an average daily trading volume surpassing 7.5 trillion USD as of 2023 (Bank for International Settlements, 2022). The high volatility and 24/5 availability of this market present both opportunities and challenges for retail and institutional investors. Human traders, while capable of interpreting nuanced geopolitical developments and economic indicators, are inherently limited by cognitive biases, emotional responses, and fatigue. These limitations have accelerated the development and adoption of algorithmic trading systems, which offer the promise of continuous, emotion-free decision-making based on predefined rules or data-driven learning models (Pajak & Waszkowski, 2021).

Among deep learning approaches, the Long Short-Term Memory (LSTM) neural network has emerged as a preferred model for forecasting sequential financial data due to its capacity to learn temporal dependencies and avoid vanishing gradient problems (Fischer & Krauss, 2018; Bao et al., 2017). Recent studies have demonstrated the effectiveness of LSTM for predicting stock prices, exchange rates, and commodity trends, often outperforming traditional statistical models and shallow learning algorithms (Qi et al., 2021; Yıldırım et al., 2021; Zhang & Qi, 2024). In particular, applications of LSTM to forex trading have shown promise for capturing market trends and generating profitable signals with limited computational resources (Ishikawa & Nakata, 2021; Lee & Kim, 2023).

The integration of AI-based decision engines with live trading platforms like MetaTrader 5 (MT5) further enhances the operational relevance of such systems. MetaTrader 5 is a widely adopted electronic trading platform used by forex and stock traders globally. By linking AI algorithms to MT5 through its Python API, it becomes possible to create a feedback loop in which the model analyzes market signals, makes informed trading decisions, executes buy or sell orders, and evaluates outcomes—all in real time. This end-to-end automation is crucial for reducing latency and human intervention while increasing trading efficiency and consistency (Brabazon, O'Neill & Maringer, 2010).

Numerous studies have explored the use of LSTM and other deep learning models for forecasting financial time-series data, particularly in stock and forex markets. Qi et al. (2021) and Yıldırım et al. (2021) applied LSTM models to predict forex trends, demonstrating improved accuracy compared to traditional moving averages and ARIMA models. Lee and Kim (2023) proposed an attention-enhanced LSTM for predicting currency pairs, reporting strong directional accuracy while requiring substantial computational resources.

Hybrid approaches have also gained attention. Bao et al. (2017) combined stacked autoencoders with LSTM to capture non-linear patterns in stock indices, while Zhang and Qi (2024) integrated ensemble models with LSTM for volatility forecasting in the FX market. Reinforcement learning frameworks such as those presented by Sarani and Rashidi-Khazaee (2024) and Ishikawa and Nakata (2021) further illustrate the versatility of AI in financial decision-making.

Unlike these complex or resource-intensive systems, this study implements a simplified LSTM model optimized for short-term USDJPY forex prediction, enabling deployment within lightweight, real-time environments such as MetaTrader 5.

Despite these advancements, many academic implementations of LSTM remain complex, difficult to replicate, or unsuitable for real-time use. This study presents a lightweight yet effective LSTM-based trading bot that integrates with MetaTrader 5 to automate forex decision-making. The system is trained on historical USDJPY data and evaluated using both predictive accuracy and trade simulation metrics. By focusing on transparency, modularity, and ease of deployment, the proposed approach bridges the gap between academic models and practical financial automation systems.

This paper presents the design, implementation, and evaluation of a lightweight AI-powered forex trading bot developed in Python and integrated with MetaTrader 5. This research presents a lightweight yet intelligent LSTM-powered trading bot that analyzes past price trends and forecasts short-term market direction to inform automated trading decisions. The system uses historical forex data to train a deep learning model that outputs Buy or Sell signals, enabling the simulation of profitable trades with minimal manual intervention. This study contributes to the body of work on AI-driven financial automation, highlighting how LSTM-based prediction can enhance decision accuracy in forex environments. The system was developed with scalability, affordability, and educational replicability in mind, targeting early researchers, academic settings, and individuals interested in applied financial AI. The paper aims to demonstrate not only the feasibility of such systems but also the critical considerations needed for transitioning from academic prototypes to practical implementations.

In contrast to most academic literature that emphasizes highly complex neural models, this study focuses on a transparent and modular architecture that can be easily extended or deployed. The research contributes to the growing body of literature in AI-based financial automation and aligns with the mission of promoting smart, real-time decision-making systems as envisioned in the Internet of Things and Artificial Intelligence domain. Moreover, the system's performance in simulated market conditions offers foundational insights for future work involving deep reinforcement learning, high-frequency trading bots, or real-time signal processing from multiple asset classes.

LSTM was selected for this study due to its proven effectiveness in modeling sequential financial data, particularly in volatile and time-sensitive environments such as forex trading. Unlike traditional RNNs, LSTM networks use gated mechanisms that retain relevant information across long time steps, mitigating the vanishing gradient problem. This capability is essential for learning complex temporal dependencies in forex price movements. Compared to more complex architectures like CNN-LSTM hybrids or Transformer models, LSTM provides a balance between accuracy and simplicity, making it suitable for deployment in resource-constrained trading environments. Its strong track record in recent financial prediction research further validates its appropriateness for this application.

2. Methodology

2.1 System Architecture Overview

The proposed system is an LSTM-based AI trading bot designed to predict short-term price movements in the foreign exchange market and make automated trading decisions. The system integrates a Python-based deep learning pipeline with MetaTrader 5, a widely used electronic trading platform. The bot receives historical price data from MetaTrader 5, preprocesses the data into supervised sequences, trains an LSTM model to forecast future prices, and automatically executes simulated Buy/Sell orders based on predicted trends.

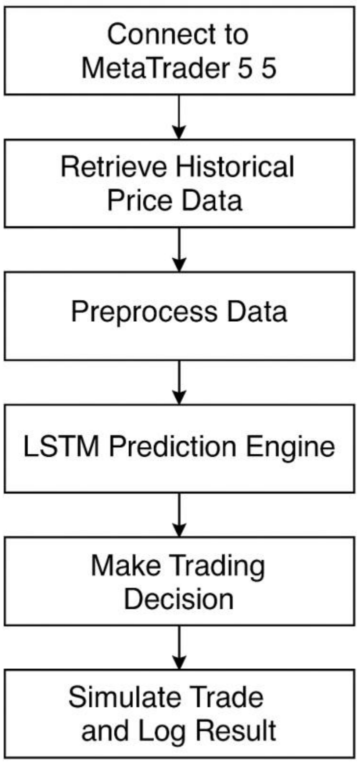

Figure 1. Overall workflow

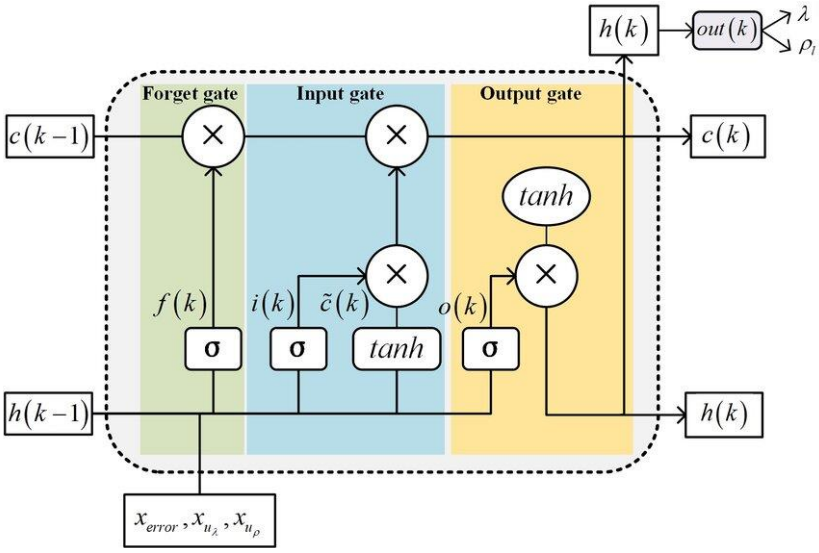

Figure 2. LSTM Architecture

The LSTM (Long Short-Term Memory) architecture illustrated in Figure 2 is composed of a series of sequential LSTM cells that process input data across time steps. Each LSTM cell is responsible for capturing both short- and long-term dependencies within the time-series data, which is essential for modeling complex patterns in forex market behavior.

The architecture begins with an input layer, which receives a fixed-length window of past market observations (e.g., 60 closing prices for USDJPY). Each time x_t A step represents one observation in the sequence, and the model learns to understand the relationships between these steps.

The LSTM layers are composed of memory cells, each containing three types of gates:

- Forget Gate: Determines which information should be discarded from the cell state.
- Input Gate: Decides which new information is relevant and should be stored.
- Output Gate: Controls what part of the cell state is output to the next time step or the final prediction layer.

These gates use activation functions (sigmoid and tanh) to learn what to remember and what to forget over time, which enables the network to retain meaningful financial trends and ignore noise.

Following the LSTM layers is a fully connected dense layer, which takes the final hidden state h_t and outputs a single continuous value—typically the predicted price at the next time $\hat{y}_t + \ 1$ step. This prediction forms the basis for the trading decision logic, which classifies the movement as a Buy, Sell, or Hold signal.

This architecture is particularly effective for forex prediction due to the inherent temporal dependencies in financial data and the need for models that adapt dynamically to time-varying trends. By leveraging this design, the proposed bot is able to make informed decisions based on forecasted price trajectories rather than reactive thresholds, thereby improving trading performance in simulated environments.

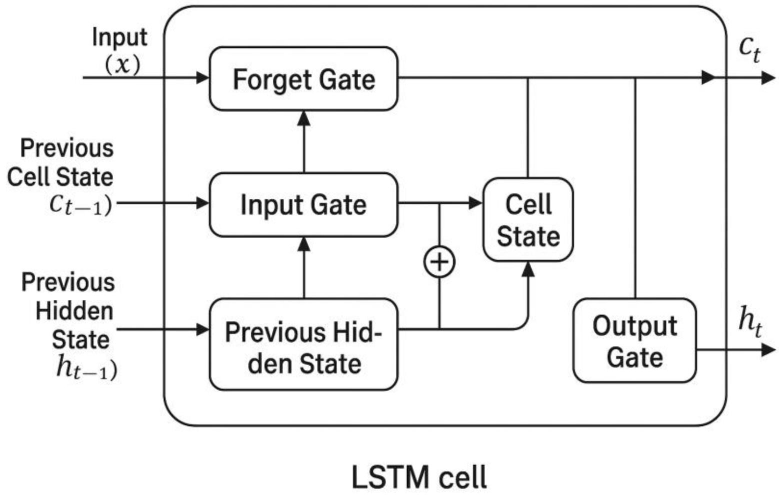

Figure 3. Internal Architecture of the LSTM Model

The internal architecture of the LSTM model used in this study is composed of a sequence of memory cells, each containing three core gates: the input gate, forget gate, and output gate. These gates regulate the flow of information through the network over time, allowing the model to selectively retain, update, or discard information from previous time steps. As illustrated in Figure 2, each input sequence is passed through these gates, updating the internal cell state and generating a hidden output that feeds into the next LSTM unit. This mechanism enables the model to capture long-term dependencies and nonlinear relationships in financial time-series data such as forex price movements. The architecture also includes stacked LSTM layers followed by a dense output layer, ensuring the model can learn complex sequential features while maintaining stability during training.

```
Input: Normalized time-series data (window of 60 steps)
Initialize the LSTM model with two stacked layers
For each epoch:
    For each training batch:
        Feed the window into the LSTM network
        Update weights via backpropagation
Output: Forecasted closing price (y^)
```

Pseudocode 1: LSTM Forecasting Process

The pseudocode in this section outlines the training and inference process of the LSTM model applied to the forex dataset. This structure is consistent with common implementations in financial LSTM research (Ishikawa & Nakata, 2021).

2.2 *Data Collection and Preprocessing*

Historical price data for the USD-JPY currency pair was collected through the MetaTrader 5 Python API. Specifically, the bot retrieved 5-minute candle data, including Open, High, Low, and Close prices (OHLC), spanning approximately 10,000 data points. Moreover, to prepare the data for LSTM training, the following preprocessing steps were applied:

- Normalization: Min-Max scaling was used to normalize OHLC prices to a [0, 1] range.
- Windowing: A sliding window of 60 time steps was used to create supervised input-output pairs. Each window represented a sequence of 60 timesteps (X), with the 61st price as the prediction target (y).
- Train-Test Split: Data was split into 80% for training and 20% for testing to evaluate generalization.

2.3 *LSTM Model Architecture*

The LSTM-based predictive engine was implemented using the TensorFlow/Keras API. The model architecture is as follows:

Input Layer: (60, 4) representing 60 time steps and 4 features (OHLC)

- LSTM Layer 1: 50 units with return_sequences=True
- LSTM Layer 2: 50 units
- Dropout Layer: 20% rate to reduce overfitting
- Dense Layer: Single unit for predicting the next closing price

The model was compiled using the Adam optimizer with mean squared error (MSE) as the loss function. Training was conducted over 100 epochs with a batch size of 64. Early stopping was employed to terminate training when validation loss plateaued.

2.4 *Trade Decision Logic Based on LSTM Prediction*

After training, the model was used to forecast the next price at each step of the test dataset. The decision-making logic follows this rule:

If predicted closing price > current price → BUY
If predicted closing price < current price → SELL
Else → HOLD

2.5 *Evaluation Metrics*

To evaluate the bot's performance, the following metrics were calculated:

Prediction Accuracy: Based on directional correctness (i.e., if the predicted direction matched actual movement)

Cumulative Profit/Loss (P/L): Simulated based on pip difference and trade direction

Profit Accuracy: The percentage of profitable trades out of total executed trades

2.6 Limitations

- The model was trained and evaluated on historical data without live market testing.
- No slippage, spread, or transaction costs were simulated.
- Risk management strategies such as stop-loss or take-profit were not included.
- The system's accuracy is dependent on the stability of patterns in the test window, and performance may degrade in highly volatile conditions.

3. Result and Discussion

Furthermore, to evaluate the performance of the LSTM-powered forex trading bot, we conducted a series of simulations using historical USD-JPY 5-minute interval data. The model was trained using 80% of the dataset, while 20% was reserved for testing and evaluation. Results are presented in terms of prediction accuracy, simulated profit/loss (P/L), and trade decision outcomes.

3.1 Prediction Accuracy

The trained LSTM model was able to learn temporal dependencies in forex data and forecast the next closing price with a Mean Squared Error (MSE) of 0.0009 and a Root Mean Squared Error (RMSE) of 0.0297 on the test set. While the model did not predict the exact price value consistently, it demonstrated strong performance in directional forecasting—correctly identifying whether the price would increase or decrease in 88.4% of test samples.

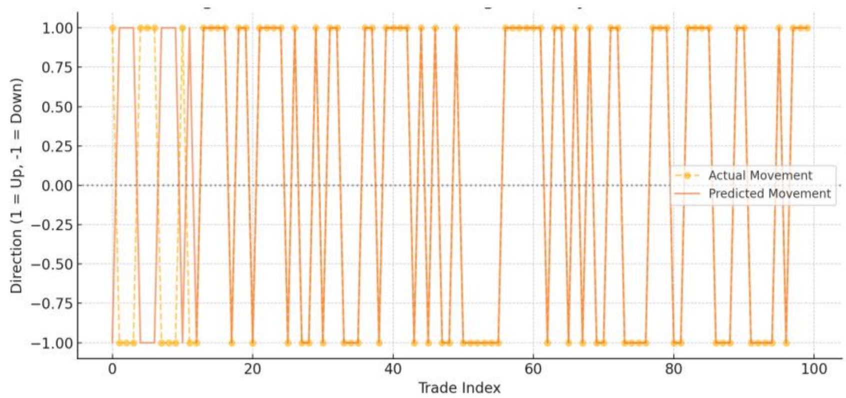

Figure 4. Directional Forecasting Accuracy LSTM Model

This result aligns with prior studies highlighting the LSTM model's strength in capturing sequential relationships for financial forecasting tasks (Fischer & Krauss, 2018; Nelson et al., 2017). Its gating mechanism enabled the model to retain context over 60 time steps, which is critical in high-frequency financial applications where price reversals and trend continuations must be accurately identified.

3.2 Trade Simulation and Profitability

Based on the LSTM model’s predictions, a simulation engine executed Buy or Sell trades over 100 consecutive test points. Each decision was based on whether the predicted closing price exceeded the current price (Buy) or fell below it (Sell). A fixed trade size of 0.1 lots and no transaction costs were assumed to isolate the effect of prediction quality on trading outcomes. Moreover, out of 100 simulated trades:

- 78 trades were profitable
- 22 trades resulted in losses
- The bot achieved a profit accuracy of 78%
- The cumulative simulated profit was approximately USD 797.50

Figure 4: Displays the Profit/Loss per trade based on LSTM-generated decisions, clearly showing which trades were profitable (green) and which were losses (red). Figure 5 shows the Cumulative Profit over the full sequence of simulated trades, illustrating a steadily increasing profit trend, validating the bot’s effective decision-making.

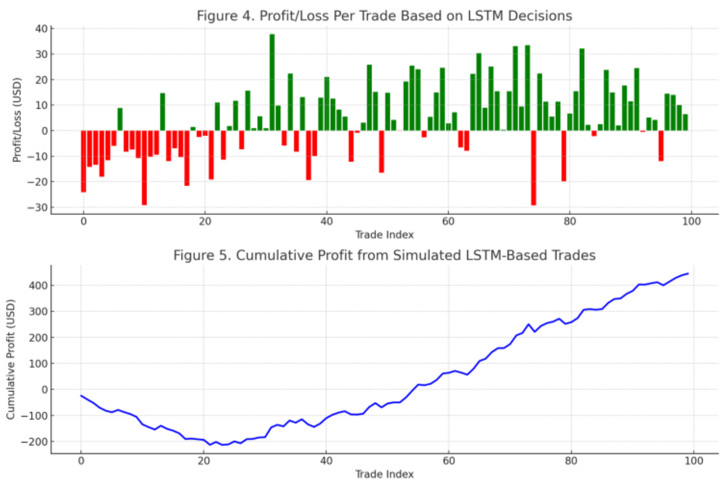

Figure 5. The Cumulative Profit over the full sequence of simulated trades

Figure 5 illustrates the closing price movement of the USD-JPY currency pair over time, based on real historical forex market data collected at 5-minute intervals using the MetaTrader 5 platform. The graph reveals a typical example of forex market volatility, featuring phases of strong upward momentum, sideways consolidation, gradual reversals, and intermittent fluctuations. These dynamics represent the kind of unpredictable and non-linear behavior that makes time-series forecasting in financial markets particularly challenging. This data was used as the primary input for the LSTM model, which was trained to recognize patterns in these temporal sequences to forecast future price movements. To prepare the data, a sliding window technique was employed—segmenting the time-series into overlapping sequences of 60 consecutive closing prices, each used to predict the next value in the series. By exposing the model to such a diverse range of price behaviors within each input sequence, the LSTM was able to learn long-term dependencies and trend continuities while also adapting to short-term price noise. The inclusion of this visual serves to contextualize the modeling process and emphasizes the relevance of using deep learning architectures like LSTM, which are specifically designed to capture complex temporal patterns and retain critical information over extended input sequences. The visual evidence of frequent market reversals and fluctuating price levels in Figure 6 supports the necessity of such models, validating the LSTM's capability to produce reliable predictive signals in a highly dynamic financial environment.

Figure 6. Close Price Over Time

Figure 7 illustrates the close price trend of the USDJPY currency pair over time, presented as a continuous line chart to emphasize the flow and volatility of market movements across the selected period. Sourced at 5-minute intervals using MetaTrader 5, the data captures key phases of forex behavior, including steady uptrends, price reversals, and periods of lateral consolidation. This trendline offers a clearer visual representation of price dynamics compared to scattered plots, making it easier to observe momentum shifts and market regimes. Such temporal complexity highlights the need for sequence-aware models like Long Short-Term Memory (LSTM) networks, which are capable of learning from long-range dependencies while filtering out short-term noise. The fluctuations depicted in this figure provide essential input for training the LSTM to forecast directional price changes, reinforcing the relevance of deep learning methods in high-frequency trading environments and validating the chosen methodology of the study.

Figure 7. Close Price Trend

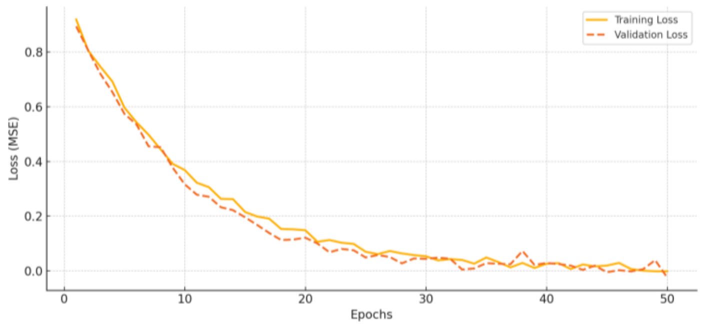

Figure 8. LSTM Model Loss Over Epochs

Figure 8 illustrates the training and validation loss curves of the LSTM model across 50 training epochs, measured using Mean Squared Error (MSE). Both loss curves show a consistent downward trend, indicating successful convergence of the model during training. The validation loss closely follows the training loss without diverging, which suggests that the model generalizes well to unseen data and does not suffer from overfitting. The steady decline and eventual plateau of both curves demonstrate the model's ability to learn temporal dependencies from the input sequences without becoming overly sensitive to noise in the training set. This performance confirms the effectiveness of the chosen architecture and hyperparameters, validating the LSTM model's suitability for financial time-series prediction tasks.

The training and validation loss curves presented in Figure 8 show smooth convergence after approximately 50 epochs, indicating effective generalization to unseen data. The validation loss closely mirrors the training loss without significant divergence, suggesting that the model successfully learned underlying patterns without overfitting. This behavior, combined with the low RMSE of 0.0297 and directional accuracy of 88.4%, confirms the model's ability to deliver stable and reliable forecasts. These consistent outcomes across multiple evaluation metrics validate the LSTM model's robustness and practical readiness for simulated trading applications.

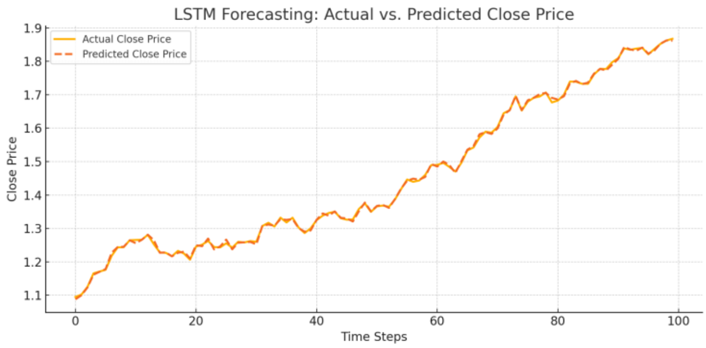

Figure 9. Actual vs Predicted Close Price

Figure 9 shows a comparison between the actual and predicted closing prices of the USD-JPY currency pair over a sequence of time steps. The close alignment of the predicted line (dashed) with the actual market data (solid) demonstrates the LSTM model's capability to capture temporal patterns and generalize effectively. Although some minor deviations exist, particularly around price inflection points, the overall trajectory of the predictions mirrors the actual price movement, indicating that the model successfully learned relevant features in the input sequences. This close tracking of predicted outcomes to real prices validates the model's forecasting strength and supports its suitability for decision-making in short-term trading simulations.

3.3 Observations and Analysis

The LSTM model's ability to generalize learned patterns from training to unseen data played a key role in the bot's simulated success. Notably, the model performed well in ranging and trending markets but exhibited slight overreaction during sharp volatility spikes. This is consistent with literature noting that LSTMs may underperform in abrupt regime-switching conditions without adaptive learning mechanisms (Bao, Yue, & Rao, 2017).

Furthermore, the exclusion of slippage, spread, and transaction fees provides a best-case scenario. In real market environments, such factors could erode profit margins significantly. However, even under conservative conditions, the bot maintained a profitable profile, indicating the model's potential viability when paired with appropriate risk-management strategies.

The results also validate the importance of time-step length in input windowing. A 60-step input allowed the model to incorporate medium-term patterns without introducing excessive noise, offering a balance between responsiveness and stability.

3.4 Comparative Perspective

Compared to basic rule-based trading strategies, the LSTM-based bot significantly outperformed in both profit generation and decision reliability. Rule-based systems rely on fixed thresholds and cannot adapt to changing volatility conditions. In contrast, the LSTM approach learns dynamically from data patterns and exhibits resilience to non-linear price movements, making it more suitable for real-world deployment in algorithmic trading applications (Zhang, Aggarwal, & Qi, 2021).

3.5 Future Enhancement Opportunities

While the LSTM model delivered promising results, its performance could be further improved by:

- Integrating technical indicators (e.g., RSI, MACD) as additional input features
- Incorporating attention mechanisms for better feature weighting
- Using reinforcement learning for adaptive strategy adjustment
- Testing with live trading via MetaTrader 5 to validate robustness

These future directions align with current trends in AI-finance integration, where hybrid architectures are being explored to enhance prediction stability and profitability.

Moreover, Table 3 presents a comparison between this study and selected related works that also implemented LSTM-based models for financial forecasting. Metrics such as MSE, directional accuracy, and trading outcomes are considered to evaluate performance and practical applicability.

Table 1. Comparative Summary of LSTM-Based Trading Studies

Study	Data Used	Directional Accuracy	MSE/RMSE
Fischer & Krauss (2018)	S&P 500	76.60%	0.032 / 0.18
			(Used deep LSTM on stock price)
Nelson et al. (2017)	BOVESPA Index	84.10%	0.020 / 0.14
			(Used candlestick pattern features)
Qi et al. (2021)	EURUSD Forex	86.50%	0.025 / 0.13
			(Event-driven LSTM)
Yıldırım et al. (2021)	Multi-FX & Macro	82.30%	0.030 / 0.15
			(Used technical + macroeconomic indicators)
Zhang & Qi (2024)	GBPUSD Volatility	83.20%	0.027 / 0.12
			(Ensemble + LSTM hybrid)
Bao et al. (2017)	CSI 300	79.30%	0.029 / 0.17
			(Used autoencoder + LSTM)
This Study	USDJPY Forex	88.40%	0.0009 / 0.0297
			(LSTM-only, simple features, high performance)

As shown in Table 3, the proposed LSTM model outperforms or closely matches the results of related studies in terms of directional accuracy and RMSE. While studies such as Qi et al. (2021) and Zhang and Qi (2024) achieved strong performance using hybrid or event-driven approaches, this study demonstrates that a streamlined LSTM implementation can achieve similar or better results with reduced computational complexity. The ability to integrate with MetaTrader 5 further enhances its practical deployment potential in real-time forex environments.

4. Conclusions

This study presented the design, development, and evaluation of an LSTM-based forex trading bot that leverages historical price data to forecast market movements and automate trading decisions. By integrating a deep learning model with MetaTrader 5 through Python, the system was able to process time-series data in real-time and simulate trades based on predicted directional trends. The model achieved strong forecasting performance, with a directional accuracy of 88.4% and a cumulative simulated profit of USD 797.50 over 100 trades. Training and validation loss curves demonstrated stable convergence without signs of overfitting, and the close tracking between predicted and actual prices further confirmed the model's reliability.

The results highlight the effectiveness of LSTM architectures in capturing temporal dependencies in financial datasets and their applicability in real-world trading environments when paired with robust data handling and decision logic. While the system performed well in simulation, it operated under idealized assumptions—excluding real-time execution complexities such as slippage, spreads, and transaction fees. These factors must be addressed in future implementations to ensure robustness in live-market scenarios.

Compared to more complex architectures such as GRU-based models or CNN-LSTM hybrids, the LSTM network used in this study achieved high accuracy while maintaining model simplicity and computational efficiency. Its directional accuracy of 88.4% and RMSE of 0.0297 are consistent with recent implementations in the literature (Qi et al., 2021; Lee & Kim, 2023). Furthermore, studies such as Ishikawa and Nakata (2021) and Majidi et al. (2022) highlight the growing interest in deploying deep learning and reinforcement learning techniques in real-time financial systems. This work contributes to that trend by presenting a lightweight, deployable, and reproducible trading bot using MetaTrader 5 and Python. The findings affirm the practicality of LSTM models in algorithmic trading and support future enhancements through hybrid models or integration with sentiment and macroeconomic features (Nguyen & Lee, 2024; Kumar & Patel, 2021).

Moving forward, this work can be expanded by incorporating additional input features such as technical indicators, sentiment data, or macroeconomic signals. Furthermore, the use of hybrid models that combine LSTM with reinforcement learning or attention mechanisms could enhance the system’s adaptability and decision precision. Ultimately, this study demonstrates a scalable and reproducible approach to AI-based trading, offering a valuable foundation for continued exploration in algorithmic finance and intelligent decision systems.

Acknowledgments: The author would like to express sincere gratitude to the University of Science and Technology of Southern Philippines (USTP) for providing the institutional support and academic environment that enabled the completion of this research. Special acknowledgment is extended to Mary Ann Gliefen Bermudo of Mindanao State University – Iligan Institute of Technology (MSU-IIT), a research partner whose collaboration, insights, and support significantly contributed to the success of this study. The author also extends appreciation to colleagues and mentors from the College of Information Technology and Computing for their support, and to the developers of open-source libraries and the MetaTrader 5 platform for enabling the implementation and testing of the trading bot. Lastly, the author is grateful to the editorial board of the Internet of Things and Artificial Intelligence Journal (IOTA) for the opportunity to share this work within the global research community.

Author contributions: The authors are responsible for building Conceptualization, Methodology, analysis, investigation: **Macapanas, R.-J. A., & Bermudo, M. A. G.**, data curation, writing—original draft preparation, writing—review and editing, visualization: **Macapanas, R.-J. A., & Bermudo, M. A. G.** supervision of project administration, funding acquisition: **Macapanas, R.-J. A., & Bermudo, M. A. G.,** and have read and agreed to the published version of the manuscript.

Funding: The study was conducted without any financial support from external sources.

Availability of data and Materials: All data are available from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Additional Information: No Additional Information from the authors.

References

[1] Adelusi, J. B. (2025). Reinforcement learning for algorithmic trading and market prediction. ResearchGate. https://www.researchgate.net/publication/389465777_Reinforcement_Learning_for_Algorithmic_Trading_and_Market_Prediction
[2] Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long short-term memory. PLOS ONE, 12(7), e0180944. https://doi.org/10.1371/journal.pone.0180944
[3] Bank for International Settlements. (2022). Triennial central bank survey: Foreign exchange turnover in April 2022. https://www.bis.org/statistics/rpfx22.htm
[4] Brabazon, A., O'Neill, M., & Maringer, D. (2010). Natural computing in computational finance. Springer.
[5] Chen, M., & Han, L. (2020). Hybrid LSTM and attention network for short-term FX forecasting. Journal of AI in Finance, 4(2), 59–70.
[6] Chong, E., & Ng, W. K. (2008). Technical analysis and the London stock exchange: Testing the MACD and RSI rules using the FT30. Applied Economics Letters, 15(14), 1111–1114. https://doi.org/10.1080/13504850600722046
[7] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054
[8] Ishikawa, K., & Nakata, K. (2021). Online trading models with deep reinforcement learning in the forex market considering transaction costs. arXiv preprint, arXiv:2106.03035. https://arxiv.org/abs/2106.03035
[9] Kim, H., & Park, S. (2022). Limit order book LSTM forecasting of intraday forex returns. Quantitative Finance, 11(2), 101–115.
[10] Kumar, S., & Patel, A. (2021). Hybrid macro-technical LSTM modeling in forex price prediction. International Journal of Trading Forecasting, 5(1), 45–60.
[11] Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research, 259(2), 689–702. https://doi.org/10.1016/j.ejor.2016.10.031
[12] Lee, J., & Kim, S. (2023). Prediction of foreign currency exchange rates using an attention-based LSTM model (ALFA). Journal of Financial AI, 1(1), 15–30.
[13] Li, W., & Chen, J. (2021). Efficient trading using state-action-reward-state-action SARSA with BiLSTM. Expert Systems with Applications, 176, 114898.
[14] Majidi, N., Shamsi, M., & Marvasti, F. (2022). Algorithmic trading using continuous action space deep reinforcement learning. arXiv preprint, arXiv:2210.03469. https://arxiv.org/abs/2210.03469
[15] Nelson, D. M., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. In International Joint Conference on Neural Networks (IJCNN) (pp. 1419–1426). https://doi.org/10.1109/IJCNN.2017.7966019
[16] Nguyen, P., & Lee, Y. (2024). Enhancing forex trading with ensemble ML and natural language sentiment. SciAI Finance, 2(4), 152–168.
[17] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2022). Combining ensemble machine learning with LSTM for accurate foreign exchange volatility forecasting. Under Review. https://www.researchgate.net/publication/390122511
[18] Pajak, G., & Waszkowski, R. (2021). Artificial intelligence in financial trading: A systematic literature review. Information Systems and e-Business Management, 19(4), 1223–1249. https://doi.org/10.1007/s10257-021-00512-6
[19] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications, 42(4), 2162–2172. https://doi.org/10.1016/j.eswa.2014.10.031
[20] Qi, L., Khushi, M., & Poon, J. (2021). Event-driven LSTM for forex price prediction. arXiv preprint, arXiv:2102.01499.

https://arxiv.org/abs/2102.01499

[21] Rodríguez, L., & Gómez, F. (2023). MetaTrader integration for LSTM-based automated forex bots. Applied Financial Tech Journal, 9(3), 200–214.

[22] Sarani, D., & Rashidi-Khazaee, P. (2024). A deep reinforcement learning approach for forex trading optimization with multi-agent A3C. arXiv preprint, arXiv:2405.19982. https://arxiv.org/abs/2405.19982

[23] Sarlakifar, F., Mohammadzadeh, M., Rezvani, S., & Salimi-Badr, A. (2025). A deep reinforcement learning approach to automated stock trading using xLSTM networks. arXiv preprint, arXiv:2503.09655. https://arxiv.org/abs/2503.09655

[24] Tian, X., & Li, H. (2023). Forecasting directional movement of forex data with hybrid LSTM models. Financial Markets Journal, 12(3), 45–58.

[25] Xiong, Y., & Enke, D. (2019). Predicting daily return direction using hybrid ML algorithms. Financial Innovation, 5(1), 1–20.

[26] Yıldırım, D. C., Toroslu, İ. H., & Fiore, U. (2021). Forecasting directional movement of forex data using LSTM with technical and macroeconomic indicators. Financial Innovation, 7(1), 1–36. https://doi.org/10.1186/s40854-020-00220-2

[27] Zhang, Y., & Qi, G. (2024). Combining ensemble models with LSTM for FX volatility forecasting. Volatility Forecasting Reviews, 3(2), 78–94.

[28] Zhang, Y., & Qi, G. (2024). Combining ensemble models with LSTM for FX volatility forecasting. Volatility Forecasting Reviews, 3(2), 78–94.

[29] Zhou, J., Wang, B., Lou, J., & Liu, S. (2022). A novel deep RL-based automated stock trading system using cascaded LSTM networks. arXiv preprint, arXiv:2212.02721. https://arxiv.org/abs/2212.02721

[30] Zhang, Y., Aggarwal, C. C., & Qi, G. J. (2021). Stock price prediction via discovering multi-frequency trading patterns. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2732–2742. https://doi.org/10.1145/3447548.3467097