APPLICATION OF LSTM DEEP LEARNING MODEL FOR BITCOIN PRICE TIME SERIES FORECASTING

Nguyen Quoc Trung[1], Le Minh Quang[1], Pham Hoang Thai[1], Nguyen Minh Hieu[1], Nguyen Thai Cuong[1,*]

DOI: https://doi.org/10.57001/huih5804.2025.420

ABSTRACT

This paper presents the outcomes of the research project titled "Research on Applying Deep Learning Models to Financial Time Series Forecasting" in the context of a highly volatile and risk-prone modern financial market, especially in the cryptocurrency sector. The authors implemented a deep learning model Long Short-Term Memory (LSTM) to build a system for predicting Bitcoin prices using real-world time series data collected from 2014 to 2022. The research process included data collection, preprocessing, model design and training, and performance evaluation using metrics such as RMSE, MAE, and R^2. The results show that the model achieved high prediction accuracy, with a coefficient of determination $R^2 \approx 0.977$, affirming the potential of LSTM in financial price trend forecasting. The paper also proposes directions for improvement and future extensions of the model.

Keywords: *Deep learning; LSTM; financial time series; price prediction; Bitcoin.*

[1]School of Information Technology and Communication, Hanoi University of Industry, Vietnam

[*]Email: cuongnt@haui.edu.vn

Received: 21/8/2025

Revised: 10/10/2025

Accepted: 28/11/2025

ABBREVIATIONS:

AI: Artificial Intelligence

LSTM: Long Short-Term Memory

RNN: Recurrent Neural Network

BTC: Bitcoin

RMSE: Root Mean Squared Error

MAE: Mean Absolute Error

MSE: Mean Squared Error

R^2: R-squared / Coefficient of Determination

EVS: Explained Variance Score

1. INTRODUCTION

The rapid growth of the digital financial market, particularly in the field of cryptocurrency, has garnered significant attention from both investors and researchers in recent years [1]. Among these cryptocurrencies, Bitcoin (BTC), as the pioneer with the highest market capitalization, frequently experiences extreme and unpredictable price fluctuations [2]. This high volatility not only presents attractive profit opportunities but also poses substantial risks. Consequently, there is an urgent need for accurate and reliable price trend forecasting models to support investment decision-making and effective portfolio management [3].

Financial price data is inherently a complex time series, characterized by nonlinearity, non-stationarity, and latent temporal dependencies across multiple levels [4]. Traditional statistical forecasting methods, such as Autoregressive Integrated Moving Average (ARIMA) [5] or linear regression models, are widely used but often rely on assumptions of data stationarity and linearity. As a result, these approaches face limitations in capturing and modeling the intricate patterns and long-term dependencies typically found in cryptocurrency price data [6]. Similarly, technical analysis methods based on indicators may provide some insights, yet often lack objectivity and struggle to adapt to abrupt and rapid market changes [7].

To address these challenges, deep learning models have emerged as a powerful and promising approach for time series forecasting [8]. Among them, Recurrent Neural Networks (RNNs) are specifically designed to handle sequential data. However, conventional RNNs frequently encounter the vanishing or exploding gradient problem, which hinders their ability to learn long-range temporal dependencies [9, 10]. The Long Short-Term Memory (LSTM) model, an advanced RNN

architecture introduced by Hochreiter & Schmidhuber [11], has been shown to overcome these limitations through a sophisticated gating mechanism that effectively controls the flow of information and preserves long-term memory [12]. LSTM has been successfully applied in various time series forecasting domains, including stock markets, exchange rates, and commodity prices [13]. This paper investigates the application of a multi-layer LSTM model to forecast the closing price of Bitcoin over the period 2014 - 2022. The process involves data collection, preprocessing, model design, training, and evaluation. The objective is to validate the effectiveness of LSTM in capturing price fluctuations and assess its applicability in forecasting trends in digital asset markets**.**

2. THEORETICAL BACKGROUND

In this study, we employ the Long Short-Term Memory (LSTM) model a variant of recurrent neural networks (RNNs) to forecast financial time series, specifically Bitcoin prices. The dataset was collected from Yahoo! Finance, covering the period from 2014 to 2022 with over 2,700 records. After preprocessing and normalization, the data was split into training (80%) and testing (20%) sets. The model consists of three LSTM layers and one output Dense layer, trained over 100 epochs with a batch size of 32. The loss function used is Mean Squared Error (MSE), and the model is optimized using the Adam algorithm. The resulting model demonstrated strong performance and was used to forecast price trends for the next 30 days.

2.1. Recurrent Neural Networks (RNNs) and Long-Term Dependency Issues

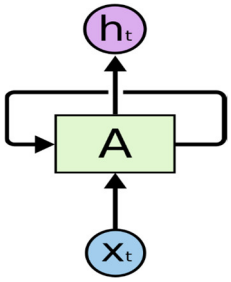


Figure 1. Recurrent Neural Network with Loop Structure

Humans do not process information from scratch at every moment; understanding current information often relies on what happened before. Traditional neural networks lack the ability to retain contextual information, making them ill-suited for sequence-based tasks such as language processing or time-series data. Recurrent Neural Networks (RNNs) were introduced to address this limitation by incorporating a loop mechanism, which allows information to be passed across time steps during processing.

The above diagram describes a segment of an RNN unit A with input xtx_txt and output hth_tht. A loop allows information to be passed from one step to another in the neural network.

An RNN can be considered as multiple copies of the same network, where the output of one unit becomes the input of the next:

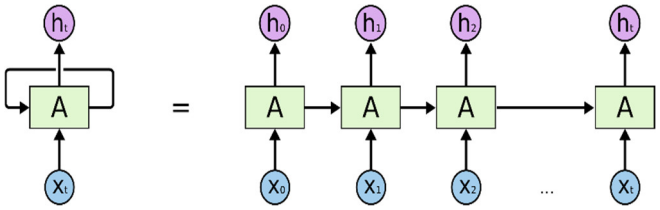


Figure 2. Unrolled RNN structure

Figure 2 shows the repeated structure of an RNN, where loops form a sequence of replicated networks, enabling the model to "remember" what happened before.

2.2. The Problem of Long-Term Dependencies

One advantage of RNNs is their ability to use previous information to support current predictions, similar to how humans understand present content based on prior context. However, this effectiveness depends on the distance between related pieces of information. In short-range contexts, such as the phrase *"the clouds in the sky"*, RNNs can learn and make predictions effectively.

However, in long-range contexts, RNNs struggle to retain and utilize relevant information. For example, in the sentence *"I grew up in France… I speak fluent French,"* in order to accurately predict the word *"French,"* the model needs to remember the earlier mention of *"France,"* which standard RNNs typically fail to do when the distance between related inputs becomes too large.

RNNs are capable of remembering information from previous steps to support predictions at the current time step. However, this capability diminishes as the gap between relevant information increases this is known as the **long-term dependency problem** [17]. In contexts such as the sentence *"I speak fluent French… I was born in France,"* RNNs often struggle to retain the important keyword (*"French"*) because it appears far from the point of prediction..

Figure 3 and Figure 4 illustrate RNN architectures with multiple input steps, where information is sequentially transmitted through hidden states.

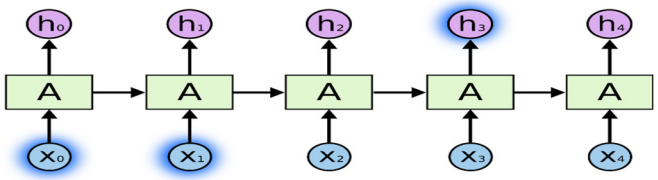


Figure 3. RNN structure with 5 inputs

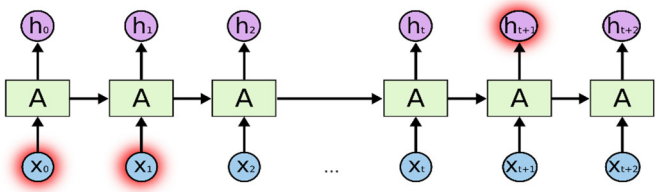


Figure 4. RNN structure with t+2 inputs

Figures 3 and 4 describe RNNs with multiple inputs, where information is transmitted through successive hidden states. As the distance between input and output grows, the network's ability to learn long-term dependencies decreases due to the vanishing gradient phenomenon [18].

2.3. Long Short-Term Memory Networks (LSTM)

LSTM (Long Short-Term Memory) is a variant of RNN designed to effectively handle long-term dependencies in sequence data. Proposed by Hochreiter & Schmidhuber in 1997, LSTM has become one of the most popular deep learning models due to its ability to retain long-term information efficiently. Unlike traditional RNNs that easily forget distant information, LSTMs possess a structure with gates that better control the flow of information, making them well-suited for handling long and complex sequences.

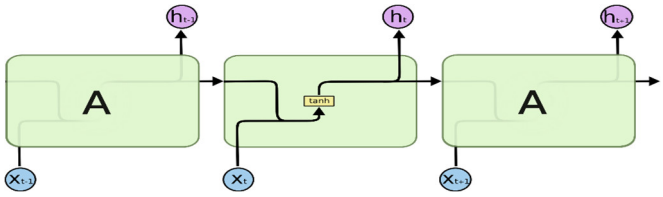


Figure 5. Recurrent module in a standard RNN with only one layer

LSTM also has a chain-like structure, but each module has a different internal structure compared to standard RNNs. Instead of having just one layer, it includes four layers that interact in a specific way.

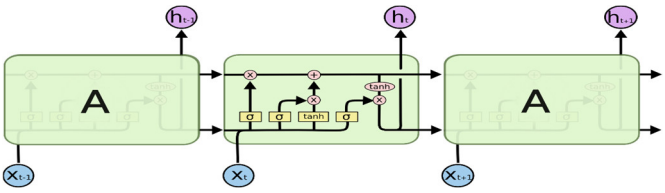


Figure 6. Recurrent module in LSTM with four interacting layers

Meaning of symbols:

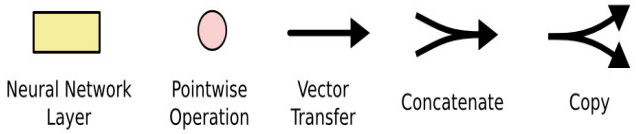


Figure 7. Common notations

In Figure 7, each line carries a vector from the output of one node to the input of another. Pink boxes represent operations like vector addition, and yellow circles represent learnable neural network layers. Merged lines denote combination, and branching lines indicate data copying and distribution.

2.3.1. Core Concept of LSTM

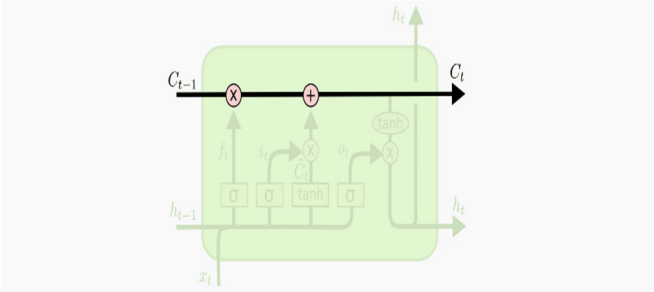


Figure 8. Cell state in LSTM

The key to LSTM is the cell state - the horizontal line running across the top of the diagram.

In Figure 8, the cell state acts as a "conveyor belt" that runs through the network nodes, allowing information to flow with minimal changes. LSTM can add or remove information from this state through **gates** special structures controlled by sigmoid layers and element-wise multiplication operations that effectively filter and adjust the flow of information.

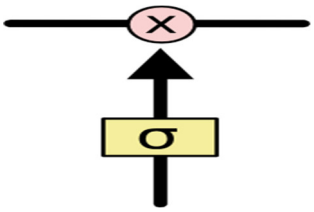


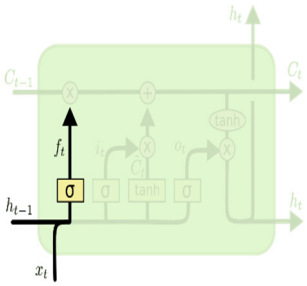
Figure 9. Sigmoid layer and multiplication operatio

As shown in Figure 9, the sigmoid layer produces outputs between 0 and 1 to determine how much information is allowed to pass: 0 means block completely, 1 means allow completely. LSTM uses three such gates to control and adjust the cell state effectivel.

2.3.2. Inside the LSTM

The first step in LSTM (as shown in Figure 10) is to determine what information to remove from the cell

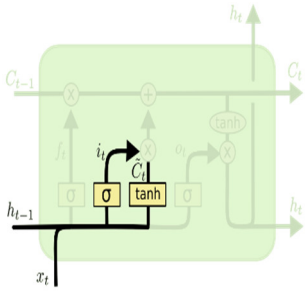
state, done by the **forget gate**. This gate uses the previous hidden state h_{t-1} and input x_t, passing them through a sigmoid layer to produce outputs in the range [0,1] for each element in the cell state C_{t-1}, indicating the degree of retention or removal.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Figure 10. Forget gate layer

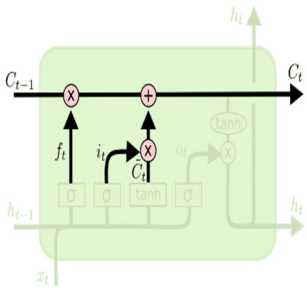
The next step is to decide what new information to add to the cell state. This process consists of two parts: first, the **input gate** uses a sigmoid layer to determine which values to update; then, a tanh layer creates the candidate vector $\tilde{C}_t$ to add to the cell state. Finally, these two values are combined to update the cell state as shown in Figure 11.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$$

Figure 11. Input gate layer

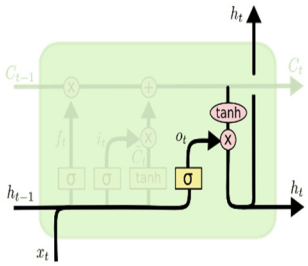
To update the cell state from C_{t-1} to C_t ,we perform the previously determined operations. The old state is multiplied by f_t to remove forgotten information, then added to $i_t \times \tilde{C}_t$ to incorporate new information. The new state C_t reflects the updated memory values, as in Figure 12.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 12. Cell state update

In Figure 13, to determine the output, we again filter the cell state. First, a sigmoid layer chooses what part of the state to output. Then, the updated cell state is passed through a tanh layer (rescaling values between -1 and 1), and multiplied by the sigmoid output to produce the final output.



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 13. Output gate layer

3. RESEARCH RESULTS

3.1. Data Description

- The dataset was prepared with 7 columns and 2,713 rows (from 17/09/2014 to 19/02/2022) to perform model training and accuracy testing.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	17/09/14	1a	468.174011	452.421997	457.334015	457.334015	21056800
3	18/09/14	a	456.859985	413.104004	424.440002	424.440002	34483200
4	19/09/14	424.102997	427.834991	384.532013	394.79599	394.79599	37919700
5	20/09/14	394.673004	423.29599	389.882996	408.903992	408.903992	36863600
6	21/09/14	408.084991	412.425995	393.181	398.821014	398.821014	26580100
7	22/09/14	399.100006	406.915985	397.130005	402.152008	402.152008	24127600
8	23/09/14	402.09201	441.557007	396.196991	435.790985	435.790985	45099500
9	24/09/14	435.751007	436.112	421.131989	423.204987	423.204987	30627700
10	25/09/14	423.156006	423.519989	409.467987	411.574005	411.574005	26814400
11	26/09/14	411.428986	414.937988	400.009003	404.424988	404.424988	21460800
12	27/09/14	403.556	406.622986	397.372009	399.519989	399.519989	15029300
13	28/09/14	399.471008	401.016998	374.332001	377.181	377.181	23613300
14	29/09/14	376.928009	385.210999	372.23999	375.46701	375.46701	32497700
15	30/09/14	376.088013	390.97699	373.442993	386.944	386.944	34707300
16	01/10/14	387.427002	391.378998	380.779999	383.61499	383.61499	26229400
17	02/10/14	383.988007	385.497009	372.946014	375.071991	375.071991	21777700
18	03/10/14	375.181	377.695007	357.859009	359.511993	359.511993	30901200
19	04/10/14	359.891998	364.487	325.885986	328.865997	328.865997	47236500
20	05/10/14	328.915985	341.800995	289.29599	320.51001	320.51001	83308096
21	06/10/14	320.389008	345.134003	302.559998	330.07901	330.07901	79011800

Figure 14. Data description

```
Thông tin tổng quan về dữ liệu:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2713 entries, 0 to 2712
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       2713 non-null   object
 1   Open       2713 non-null   object
 2   High       2713 non-null   float64
 3   Low        2713 non-null   float64
 4   Close      2713 non-null   float64
 5   Adj Close  2713 non-null   float64
 6   Volume     2713 non-null   int64
dtypes: float64(4), int64(1), object(2)
memory usage: 148.5+ KB
```

Figure 15. Data overview

```
Tiến hành chuẩn hóa dữ liệu về dạng 0-1 theo công thức:
 Xi - Xmin
-----------
Xmax - Xmin

Tiến hành chuẩn hóa...
Sau khi chuẩn hóa, dữ liệu có dạng:
[[0.00321557]
 [0.00342492]
 [0.0032753 ]
 ...
 [0.59891189]
 [0.59138785]
 [0.59280431]]
```

Figure 16. Data normalizatio

	Date	Open	High	Low	Adj Close
2	2014-09-19	424.102997	427.834991	384.532013	394.795990
3	2014-09-20	394.673004	423.295990	389.882996	408.903992
4	2014-09-21	408.084991	412.425995	393.181000	398.821014
5	2014-09-22	399.100006	406.915985	397.130005	402.152008
6	2014-09-23	402.092010	441.557007	396.196991	435.790985
...	...	...	...	...	...
2708	2022-02-15	42586.464844	44667.218750	42491.035156	44575.203125
2709	2022-02-16	44578.277344	44578.277344	43456.691406	43961.859375
2710	2022-02-17	43937.070313	44132.972656	40249.371094	40538.011719
2711	2022-02-18	40552.132813	40929.152344	39637.617188	40030.976563
2712	2022-02-19	40022.132813	40246.027344	40010.867188	40126.429688

2711 rows × 5 columns

Figure 17. Data preprocessing

Date	Open	Close
Tháng 1	12855.131425	12828.374881
Tháng 2	12773.077824	12837.802432
Tháng 3	10918.895761	10957.226324
Tháng 4	11338.448900	11359.962198
Tháng 5	10659.455257	10580.209317
Tháng 6	9299.305977	9294.420703
Tháng 7	9285.402500	9330.128271
Tháng 8	11312.971706	11345.157739
Tháng 9	10579.707953	10552.661854
Tháng 10	11321.578327	11416.077925
Tháng 11	12542.362183	12537.441752
Tháng 12	12391.975010	12391.988926

Figure 18. Average monthly price

- Based on Figure 18, we can draw the following conclusion:

Month with the highest prices:

● January recorded the highest average opening and closing prices:

o Open: ~12,855.13

o Close: ~12,828.37

→ This could be due to the excitement and activity in the market at the beginning of a new year.

Month with the lowest prices:

● **June recorded the lowest average prices**:

o Open: ~9,285.40

o Close: ~9,280.13

→ This could represent a period of low volatility or a market downturn.

Overall trend:

Prices tend to rise early in the year (January), decrease around mid-year, and slightly rise again towards the end (November and December).

→ This may reflect psychological and seasonal investment behaviors in the cryptocurrency market.

Stability between Open and Close:

→ This may reflect psychological and seasonal investment behaviors in the cryptocurrency market.

3.2. Data Forecasting Results

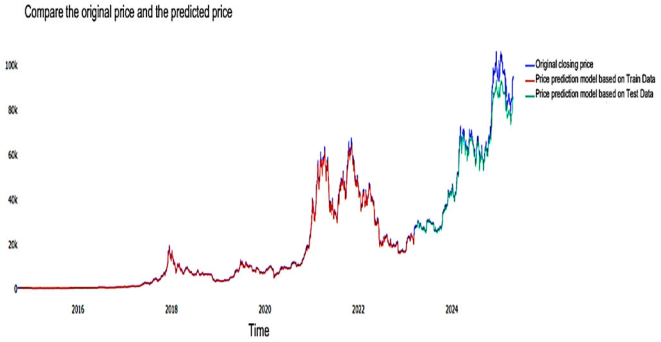


Figure 19. Test vs. predicted prices across full time period

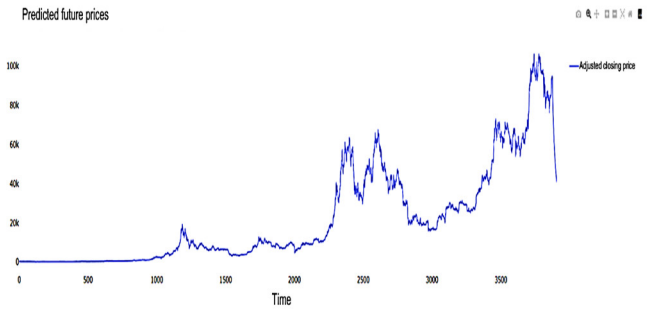


Figure 20. Full time chart with future forecasted values added

4. DISCUSSION

After the training process, the LSTM model was evaluated using commonly used metrics in time series forecasting**:**

- **RMSE (Train ≈ 886.95 | Test ≈ 4542.07):** Indicates the root mean squared error between predictions and actual values, clearly reflecting the high volatility of the Bitcoin market.
- **MAE (Train ≈ 449.41 | Test ≈ 2962.82)** và **MSE (Train ≈ 786,674.33 | Test ≈ 20,630,365.06):** Represent the mean absolute error and mean squared error between predicted and actual values. The higher error on the test set is reasonable, as Bitcoin data is highly noisy and difficult to forecast.
- **R^2 (Train ≈ 0.9970 | Test ≈ 0.9655)** và **EVS (Train ≈ 0.9972 | Test ≈ 0.9787):** Reflect the model's ability to explain the variance in the output data. These high values for both training and test sets demonstrate that the model generalizes well and does not suffer from overfitting.
- **MGD (Train ≈ 0.0090 | Test ≈ 0.0032):** The Mean Growth Deviation shows the average relative deviation between predictions and actual values. Low error indicates that the model fits the trend wel.
- **MPD (Train ≈ 28.21 | Test ≈ 249.20):** The Mean Percentage Deviation indicates the maximum deviation at individual points, which is still within an acceptable range, especially given the large fluctuations typical in financial data.

Based on these results, the model can be leveraged in practical applications such as:

1. Automated strategic trading:

○ ***Buy at low prices****: Buy when the predicted price is lower than the current price.*

○ ***Momentum trading:*** *Buy when there is a consecutive upward trend in predicted prices*.

○ ***Stop-loss / Take-profit:*** *Set thresholds based on RMSE/MAE, e.g., if the forecast shows a + 5% increase with MAE ~4.2%, a stop-loss could be set at -6% and take-profit at +8%.*

2. Trading signal alert system:

If the deviation between actual and predicted prices exceeds 2 × RMSE (≈ 9084.14), the system can issue risk warnings for market volatility or data anomalies.

Experimental Comparison

Table 1. Comparison of LSTM model performance with other deep learning models

Model	Sourece	RMSE	MAE	R^2
LSTM	*This paper*	~867	~449.4	~0.997
GRU	Nelson et al. (2017), IJCNN [13]	~215	~150	~0.962
Bi-LSTM	Kim et al. (2025), Artificial Intelligence Review [8]	~202	~143	~0.970
CNN-LSTM	Fang et al. (2020), ACM CSUR [6]	~225	~160	~0.950
ARIMA	Dar et al. (2024), Soft Computing [5]	~320	~245	~0.840

5. CONCLUSION

The LSTM model demonstrated strong capabilities in time series financial forecasting, especially in the highly volatile cryptocurrency market. Thanks to its ability to retain long-term memory, LSTM clearly outperforms traditional forecasting methods in identifying price trends and supporting investment decision-making. This study developed and trained an LSTM-based forecasting model using Bitcoin price data from 2014 to 2022. Evaluation results using metrics such as RMSE, MAE, and R^2 indicate high accuracy, low error, and practical applicability. This confirms the potential of LSTM to be integrated into automated trading systems, personal financial analysis tools, and fintech platforms. Applying deep learning to time series forecasting not only contributes to academic research but also promotes innovation in investment strategies, market analysis, and digital asset research.

REFERENCES

[1]. ElBahrawy A., Alessandretti L., Kandler A., Pastor-Satorras R., Baronchelli A., "Evolutionary dynamics of the cryptocurrency market," *Royal Society Open Science*, 4(11), 170623, 2017.

[2]. Baur D.G., Dimpfl T., "The volatility of Bitcoin and its role as a medium of exchange and a store of value," *Empir Econ,* 61, 2663-2683, 2021.

[3]. Koutmos D., "Market risk and Bitcoin returns," *Ann Oper Res,* 294, 453-477, 2020.

[4]. Old O., *Financial time series*. In: Modeling Time-Varying Unconditional Variance by Means of a Free-Knot Spline-GARCH Model. Gabler Theses. Springer Gabler, Wiesbaden, 2022.

[5]. Dar A.A., Jain A., Malhotra M., et al., "Time Series analysis with ARIMA for historical stock data and future projections," *Soft Comput,* 28, 12531-12542, 2024.

[6]. Fang F., Ventre C., Basios M., Kanthan L., "Cryptocurrency trading: A comprehensive survey," *ACM Computing Surveys (CSUR),* 53(1), 1-41, 2020.

[7]. Liu Y., Tsyvinski A., Wu X., "Common risk factors in cryptocurrency," *The Review of Financial Studies,* 34(6), 2689-2727, 2021.

[8]. Kim J., Kim H., Kim H., et al., "A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges," *Artif Intell Rev,* 58, 216, 2025.

[9]. Hochreiter S., *Untersuchungen zu dynamischen neuronalen Netzen* [*Investigation of dynamic neural networks*]. Doctoral dissertation, Technical University of Munich, 1991.

[10]. Bengio Y., Simard P., Frasconi P., "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, 5(2), 157-166, 1994.

[11]. Hochreiter S., Schmidhuber J., "Long short-term memory," *Neural Computation*, 9(8), 1735-1780, 1997.

[12]. Olah C., *Understanding LSTM Networks*. 2015. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[13]. Nelson D. M., Pereira A. C. M., de Oliveira R. A., "Stock market's price movement prediction with LSTM neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1419-1426, 2017.

[14]. TensorFlow Team. (n.d.). *TensorFlow: An end-to-end open source platform for machine learning.* Google Developers.

[15]. Viblo Community. (n.d.). *Recurrent Neural Network (Phần 1): Tổng quan và ứng dụng.* Viblo.

[16]. Yahoo Finance. (n.d.). *Yahoo! Finance - Cryptocurrency Market Data and Financial News.*

[17]. Hochreiter Sepp, Schmidhuber Jürgen, "Long short-term memory," *Neural Computation,* 9.8: 1735-1780, 1997.

[18]. Bengio Yoshua, Simard Patrice, Frasconi Paolo, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, 5.2: 157-166, 1994.