

# Attention-Enhanced Long Short-Term Memory for Bitcoin Price Prediction<sup>\*</sup>

Wiktoria Plechta

*Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44100 Gliwice, POLAND*

## Abstract

Predicting the value of Bitcoin is important from a risk management perspective because it can optimize profits. This task relies on historical data analysis and predictions of the next. This paper presents a study on predicting Bitcoin prices using deep learning models. Standard LSTM models were compared with Attention-augmented variants to evaluate how effectively each model adapts to Bitcoin's frequent fluctuations and extreme price changes. The comparative analysis provides insights into the strengths of standard LSTMs and the potential of Attention mechanisms in financial time series forecasting.

## Keywords

rnn, lstm, attention, bitcoin price prediction

## 1. Introduction

Cryptocurrencies, especially Bitcoin, have garnered immense interest from investors and researchers due to their high volatility and potential for significant returns. However, accurately predicting Bitcoin's price is challenging, as it is influenced by complex factors such as market sentiment, trading volumes, and global economic events. Machine learning models, particularly neural networks, have achieved great success in financial [1]. Among these, Long Short-Term Memory (LSTM) networks capture long-range dependencies and sequential patterns, making them suitable for time series forecasting [2, 3].

Despite their strengths, LSTM networks can struggle when applied to highly volatile financial data. A key limitation of standard LSTM models is their tendency to weigh all input points in a sequence equally [4, 5]. This may lead to the loss of critical information from particularly influential points in time. The attention mechanism was introduced as an extension to recurrent neural networks [6] - it allows these models to focus selectively on more important elements in the input sequence. By doing so, attention-enhanced LSTM models could improve predictive performance, especially in volatile assets like Bitcoin.

This paper introduces an Attention-Enhanced Long Short-Term Memory (AE-LSTM) model for Bitcoin price prediction. It conducts a comparative analysis to evaluate the effects of adding an attention mechanism. We will compare the AE-LSTM models with standard LSTM models to assess whether attention mechanisms enhance prediction accuracy and robustness in this context. Through this comparison, the study seeks to provide insights into the practical impact of attention layers in financial forecasting.


---

*IVUS 2025: INFORMATION SOCIETY AND UNIVERSITY STUDIES 2024, May 17, Kaunas, Lithuania*

✉ [wp311004@student.polsl.pl](mailto:wp311004@student.polsl.pl) (W. Plechta)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

## 2. LSTM model

Long Short-Term Memory (LSTM) cells are a complex type of Recurrent Neural Network designed to handle sequential data and capture long-range dependencies. They are built to address the problem of vanishing gradients, where information from earlier in the sequence fades as it passes through many layers or timesteps. LSTM cells use a gating mechanism that decides which information to keep, forget, or pass along to the next time step. This is crucial for tasks like language modeling, where the meaning of a word often depends on the context set by previous words. LSTM cells also work well on many sequential data types beyond text, including time series, audio, video, and other data where temporal or sequential information is essential.

LSTM cell architecture is designed to address the problem of long-term dependencies by using multiple gates and a cell state that flows through the network with minimal modifications. LSTMs have three gates (input, forget, and output) that control the flow of information:

- Input gate: Controls how much new information should be added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Here,  $i_t$  represents the input gate's activation, and  $\tilde{C}_t$  is the candidate cell state, which provides potential new information to add to the cell state.

- Forget gate: Controls how much of the previous cell state should be kept.

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

Here,  $f_t$  is the forget gate output,  $W_f$  and  $b_f$  are weights and biases,  $h_{t-1}$  is the previous hidden state, and  $x_t$  is the current input.

- Output gate: Controls the final output of the LSTM cell for the current timestep.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Here,  $o_t$  is the output gate's activation, and  $h_t$  is the hidden state

It also consists of two cell states:

- Cell State: allows to carry information from earlier in the sequence to later stages without drastic changes unless controlled explicitly by the gates.
- Hidden State: encode a kind of characterization of the previous t-steps data, focusing on most recent t-step.

### 2.1. LSTM models used in these experiments

We propose a network architecture to predict Bitcoin price with six layers: input, two recurrent layers, two dropout layers and a dense output layer. The network processes sequences of historical data, each containing 60 timesteps and five features, enabling analysis based on a sequence of multiple past values. The details of the modeled networks are described as follows:

**Table 1**  
LSTM models

Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
LSTM	(None, 60, 50)	11,200	LSTM	(None, 60, 100)	42400
dropout	(None, 60, 50)	0	dropout	(None, 60, 100)	0
LSTM_1	(None, 50)	20,200	LSTM_1	(None, 100)	80400
dropout_2	(None, 50)	0	dropout_2	(None, 100)	0
dropout_2	(None, 1)	51	dropout_2	(None, 1)	101
<b>Total params</b>		<b>31,452</b>	<b>Total params</b>		<b>122,901</b>
<b>Trainable params</b>		<b>31,452</b>	<b>Trainable params</b>		<b>122,901</b>
<b>Non-trainable params</b>		<b>0</b>	<b>Non-trainable params</b>		<b>0</b>

(a) Summary of LSTM model with 50 units per layer.

(b) Summary of both LSTM models with 100 units per layer.

### 3. Attention mechanism

Attention in Neural Networks was modeled on the human ability to focus on specific information regardless of all the others. Therefore, it allows models to center on high-value input data. The attention mechanism is usually implemented in two primary steps: first, determining the attention weights across the input data, and second, creating the context vector by applying these weights to the input [7]. Mathematically, given an input sequence  $(x_1, x_2, \dots, x_T)$  and a query vector  $q$ , the attention scores can be computed in different ways, including:

- Dot-Product Attention: The score for each input element  $x_i$  is calculated as the dot product between  $q$  and  $x_i$ :

$$\text{score}(q, x_i) = q \cdot x_i$$

- Scaled Dot-Product Attention: To normalize the dot-product, the scores are divided by  $\sqrt{d_k}$  (where  $d_k$  is the dimension of the key vectors) to prevent large values in deeper networks:

$$\text{score}(q, x_i) = \frac{q \cdot x_i}{\sqrt{d_k}}$$

- Additive Attention: This form of attention uses a small neural network to learn a non-linear transformation of the query and each input element. It's computed as:

$$\text{score}(q, x_i) = v^T \tanh(W_q q + W_x x_i)$$

where  $W_q$ ,  $W_x$ , and  $v$  are learned weight matrices.

Once the raw attention scores are computed, they are normalized using a softmax function to produce attention weights that sum to 1:

$$\alpha_i = \frac{\exp(\text{score}(q, x_i))}{\sum_{j=1}^T \exp(\text{score}(q, x_j))}$$

These weights,  $\alpha_i$ , represent the model's attention to each element  $x_i$  in the sequence. The attention weights are then used to calculate the context vector  $c$ , which is a weighted sum of

the input vectors:

$$c = \sum_{i=1}^T \alpha_i x_i$$

This context vector  $c$  contains information from the input sequence, emphasizing the most relevant elements based on the attention weights.

### 3.1. Attention -Enhanced models used in these experiments

For our models, we used a custom attention layer. The attention scores are calculated using an additive attention mechanism in the attention layer function. The function first applies two dense layers with ‘tanh’ activations to the ‘inputs’. These layers produce scores for each element in the input sequence. The output from the ‘tanh’ layers is then flattened and passed through a ‘softmax’ layer to normalize the scores into a probability distribution to ensure all attention weights sum to 1. The resulting output selectively highlights significant parts of the input, enabling the model to focus on critical information for subsequent processing. As in standard LSTM models, we created three Attention Enhanced models: one with 50 units of LSTM layers and 0.3 dropouts, the second with the same dropout but 100 units per layer, and the last with 100 units but 0.6 dropouts. The details of the modeled networks are shown below.

**Table 2**

LSTM models with attention

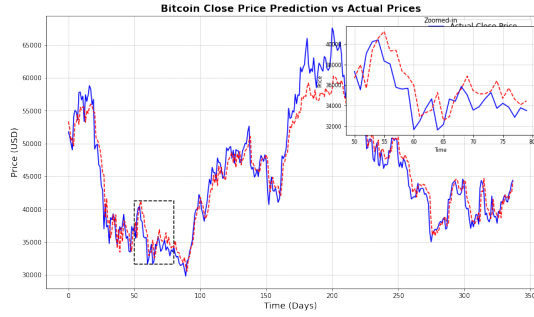
Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
LSTM	(None, 60, 50)	11,200	LSTM	(None, 60, 100)	42,400
dropout	(None, 60, 100)	0	dropout	(None, 60, 100)	0
dense	(None, 60,1)	51	dense	(None, 60, 1)	101
dense_1	(None, 60, 1)	2	dense_1	(None, 60, 1)	2
flatten	(None, 60)	0	flatten	(None, 60)	0
activation	(None, 60)	0	activation	(None, 60)	0
repeat_vector	(None, 50, 60)	0	repeat_vector	(None, 100, 60)	0
permute	(None, 60, 50)	0	permute	(None, 60, 100)	0
multiply	(None, 60, 50)	0	multiply	(None, 60, 100)	0
LSTM_1	(None, 50)	20200	LSTM_1	(None, 50)	80,400
dropout_2	(None, 50)	0	dropout_2	(None, 100)	0
dense	(None, 1)	51	dense	(None, 1)	51
<b>Total params</b>		<b>31,504</b>	<b>Total params</b>		<b>123,004</b>
<b>Trainable params</b>		<b>31,504</b>	<b>Trainable params</b>		<b>123,004</b>
<b>Non-trainable params</b>		<b>0</b>	<b>Non-trainable params</b>		<b>0</b>

(a) Summary of Attention - Enhanced model with 50 units per layer.

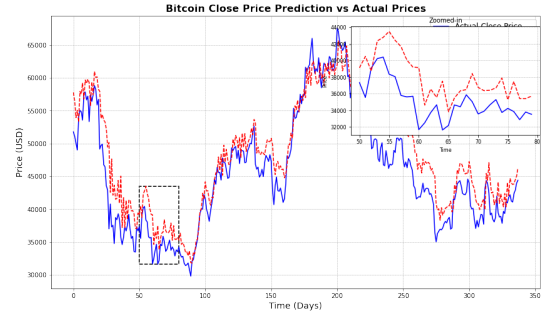
(b) Summary of both Attention-Enhanced models with 100 units per layer.

## 4. Bitcoin dataset

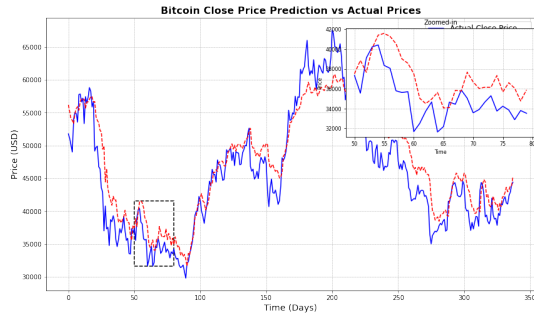
The dataset used contained daily information on Bitcoin prices and consisted of seven columns. For this study, we used five of them: Open (the opening price of Bitcoin), High (the highest price Bitcoin reached), Low (the lowest price of Bitcoin), Close (the closing price of Bitcoin),



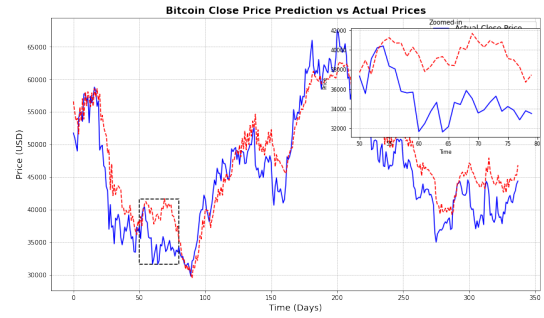
(a) LSTM Model prediction for a network with 50 units.



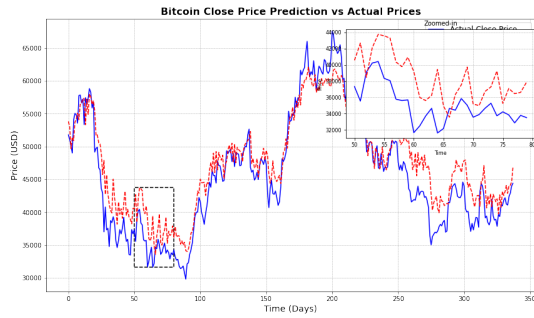
(b) LSTM Model prediction for a network with 100 units and dropout as 0.3.



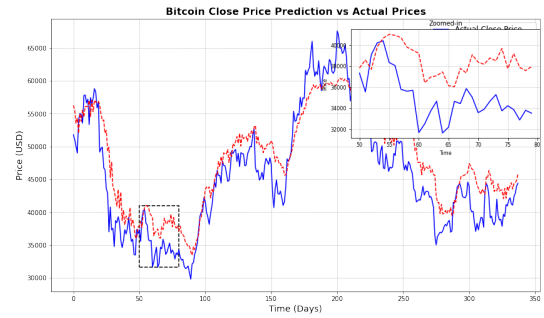
(c) LSTM Model prediction for a network with 100 units and dropout as 0.6.



(d) LSTM with Attention Model prediction for a network with 50 units.



(e) LSTM with Attention Model prediction for a network with 100 units and dropout as 0.3.



(f) LSTM with Attention Model prediction for a network with 100 units and dropout as 0.6.

**Figure 1:** Comparison of LSTM and LSTM with Attention Model predictions on first data.

and Volume (the total volume of Bitcoin traded that day). The first 1,000 records were excluded from the dataset to improve model performance. The dataset is available online on Kaggle<sup>1</sup>. The model was built without applying scaling or normalization to the data, as this decision was made to observe the raw relationships and patterns in the dataset, preserving the original feature distributions. The dataset was then split into two subsets: training and validation, with an 80:20 split applied, allocating 80% of the data to the training set and the remaining 20% to

<sup>1</sup><https://www.kaggle.com/datasets/shahidk3075/bitcoin-price-prediction-dataset>

the validation set.

## 5. Experiments

All models were trained using the Adam optimizer and the Mean Absolute Percentage Error loss function:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

The choice of the following loss function was determined by its measuring percentage error. This helps the model focus on relative accuracy rather than absolute error, which is valuable when dealing with Bitcoin's large price fluctuations. All models were trained for 1,000 epochs with a batch size of 32. We evaluated each model by having them predict the same set of 350 values, with the resulting predictions displayed in the graphs below. Overall, each model achieved relatively strong performance. However, the standard LSTM models handle Bitcoin's rapid fluctuations and sharp increases more effectively than other architectures. This limitation was observed regardless of the AE-LSTM models' number of units or dropout rates.

The prediction using the LSTM model with 50 units in the hidden layer is shown in Fig. 1a, wherein the enlarged fragment, we can see a good fit, which is mainly slightly shifted in time. Furthermore, it should be noted that the LSTM network does not achieve the maximum values. Increasing the number of units to 100 allowed to achieve the maximum values but also a worse fit over time, as the values are higher than they should be (see Fig. 1b). Doubling the dropout value contributed to obtaining similar results (see Fig. 1c), taking into account the smoothing of the prediction curve, which results in a worse fit to the real values.

Extending the model with 50-unit attention layers resulted in a much higher error due to the deviation of the predicted values from the actual values. In Fig. 1d, it can be seen that the prediction values are often above or below the actual values. Increasing the units to 100 allowed for a more accurate fit, as shown in Fig. 1e. However, the small spike values in the enlarged fragment are much higher relative to the prediction. Increasing the input values to 0.6 minimized the spikes, but the error increased due to a worse fit relative to the actual values. In the context of applying attention mechanisms to the prediction task, it was noticed that it was easier to obtain higher values in the case of spikes than without. However, the fitting of the curves concerning error was obtained using the network without the attention mechanism.

## 6. Conclusion

In this study, we introduced an Attention-Enhanced Long Short-Term Memory (AE-LSTM) model to explore the impact of incorporating an attention mechanism for Bitcoin price prediction. Our objective was to assess whether adding attention would improve the predictive power of LSTM networks. By comparing models with and without attention under varying configurations of dropout rates and the number of LSTM units, we aimed to analyze attention's effect across different architectures thoroughly.

The results indicate that the AE-LSTM and the standard LSTM models performed well. Surprisingly, the LSTM model without attention achieved slightly better predictive accuracy overall, suggesting that the attention mechanism did not consistently enhance performance in this context. These findings highlight that attention mechanisms are powerful tools for focusing on relevant input features in many applications. However, they do not necessarily benefit all models equally, particularly in time-series forecasting tasks with high-frequency volatility like Bitcoin price data.

Additionally, the inclusion of attention added more computational complexity, resulting in longer training times and increased resource demands. This increased complexity must be considered carefully, as the modest improvements seen in some configurations do not universally justify the higher computational cost.

Overall, this research suggests that while attention mechanisms have high potential, they should be applied selectively, with careful tuning and evaluation against simpler architectures. Future research could investigate alternative attention formulations or adaptive configurations that selectively apply attention to specific sequence components, potentially reducing computational load while still capturing relevant dependencies.

## Acknowledgments

This work was supported by the Rector's mentoring project "Spread your wings" at the Silesian University of Technology.

## References

- [1] K. Zhang, G. Zhong, J. Dong, S. Wang, Y. Wang, Stock market prediction based on generative adversarial network, *Procedia computer science* 147 (2019) 400–406.
- [2] K. Prokop, D. Połap, G. Srivastava, Agv quality of service throughput prediction via neural networks, in: *2023 IEEE International Conference on Big Data (BigData)*, IEEE, 2023, pp. 2493–2498.
- [3] D. Połap, G. Srivastava, A. Jaszcz, Energy consumption prediction model for smart homes via decentralized federated learning with lstm, *IEEE Transactions on Consumer Electronics* (2023).
- [4] I. Strumberger, M. Zivkovic, V. R. R. Thumiki, A. Djordjevic, J. Gajic, N. Bacanin, Multivariate bitcoin price prediction based on tuned bidirectional long short-term memory network and enhanced reptile search algorithm, in: *International Conference on Information and Software Technologies*, Springer, 2023, pp. 38–52.
- [5] T. K. Toai, R. Senkerik, I. Zelinka, A. Ulrich, V. T. X. Hanh, V. M. Huan, Arima for short-term and lstm for long-term in daily bitcoin price prediction, in: *International Conference on Artificial Intelligence and Soft Computing*, Springer, 2022, pp. 131–143.
- [6] G. Brauwers, F. Frasincar, A general survey on attention mechanisms in deep learning, *IEEE Transactions on Knowledge and Data Engineering* 35 (2021) 3279–3298.
- [7] Z. Niu, G. Zhong, H. Yu, A review on the attention mechanism of deep learning, *Neuro-computing* 452 (2021) 48–62.