Forecasting Bitcoin Price by Tuned Long Short Term Memory Model

Aleksandar Petrovic, Luka Jovanovic, Miodrag Zivkovic, Nebojsa Bacanin(✉), Nebojsa Budimirovic, and Marina Marjanovic

Singidunum University, Danijelova 32, 11000 Belgrade, Serbia
{aleksandar.petrovic,mzivkovic,nbacanin,mmarjanovic}@singidunum.ac.rs, luka.jovanovic.191@singimail.rs

Abstract. The interest for cryptocurrencies is high and hence this work focuses on providing a practical real-world application of the swarm metaheuristics and long short term memory model (LSTM). The goal is price forecasting which is interesting due to the high volatility of the cryptocurrencies. The authors apply LSTM for the solution of the problem which has been proven to reap results with this type of problem. The LSTM is further optimized by a swarm metaheuristic - arithmetic optimization algorithm (AOA). The solution was tested alongside familiar high-performing competitors with the use of standard metrics mean absolute error (MAE), mean squared error (MSE), mean absolute percentage error (MAPE), and root mean squared error (RMSE). These metrics have been used for comparison between the solutions, upon which the proposed solution obtained overall best performance that testifies to the improvement of the solution.

Keywords: cryptocurrency · prediction · arithmetic optimization algorithm · long short term memory model · optimization

1 Introduction

The potential of cryptocurrencies has been recognized to which testifies its recent end of one of its most fruitful eras that they have been through. Cryptocurrencies are digital currencies or rather computer code that serves as a medium of exchange. Through application of cryptographic algorithms the security and transparency factors are guaranteed. As a result, there is no need for a mediator as everyone on the chain can confirm the validity of a transaction. Additionally, due to its nature, cryptocurrencies have many advantages over the fiat currencies, which are the traditional currencies like the dollar and euro. Most importantly, the blockchain platform upon which cryptocurrencies are built adds one more key characteristic regarding its infrastructure. The characteristic is that the blockchain is decentralized, hence there is no central authority controlling the currency. This last feature plays a key role as it allows for independence and increases the freedom of its users.

N. Bacanin and H. Shaker (Eds.): ICIITB 2022, ACSR 104, pp. 187–202, 2023.
https://doi.org/10.2991/978-94-6463-110-4_14

Cryptocurrencies are expanding as more are designed and available for trading. This number has been only rising, even so largely from 2017 [50]. This crypto-fever resulted in attraction of a large amount of new users and investors. Unfortunately, not everyone can be a winner when it comes to any type of investment and especially in the case of cryptocurrencies. This market is currently going through its bear period, which means that value is generally decreasing [27]. This is yet another characteristic not be overlooked which is volatility. The price is subject to numerous transitions from extreme lows to extreme highs, and to capitalize on cryptocurrencies, the investor needs to take advantage of these periods.

This is the reason for applying powerful algorithms to the prediction problem of cryptocurrency value. This problem is NP-hard, meaning that is not solvable in polynomial time. Great optimizers for these types of problems have proven to be metaheuristic machine learning algorithms, especially from the swarm intelligence (SI) field. These stochastic evolutionary nature based algorithms have found a large domain of application as NP-hard problems are very much present in real-world tasks. To their advancement, SI algorithms are susceptible to improvements through hybridization, which makes this field even more interesting in terms of researching.

This work applies a long short term memory model for price prediction which is hybridized with the arithmetic optimization algorithm. These models are applied with this type of problem and only benefit from strong metaheuristic optimizers [30]. LSTM is considered as a variant of recurrent neural networks (RNN), and artificial neural networks have successfully been optimized in multiple different cases with these types of algorithms [35]. The Bitcoin (BTC) is the most popular and most valued cryptocurrency [43]. BTC heavily influences the market, and if the price graphs are not looking too good it also reflects on all the other currencies. This is the case with its rise as well. Note that all the previously mentioned concepts regarding cryptocurrencies apply to Bitcoin as well. Its importance is also due to it being the cryptocurrency that broke the ice and had the most support from the community. Hence, its price has been rising for the longest time and some consider it the number one currency [39], even though it is still highly unstable when not comparing it to other cryptocurrencies. This has resulted in building trust with the users which allowed for other cryptocurrencies to follow up, but note that they still trot behind BTC. The final result is a new medium of exchange accepted by the large masses and it is important because it holds potential to shake up the world economy.

Regarding the subject of value forecasting, there are two different types. Univariate, for singular variables, and multivariate for mutlitple. The authors in this work focus on univariate casting.

The written research is divided in sections and are described in following text: Sect. 2 provides the basics of the used technologies in the work and similar applications and solutions, Sect. 3 describes the AOA and the modifications made with the LSTM, Sect. 4 has a goal of explaining the testing conditions and the results of the said tests, and finally the Sect. 5 provides a summary of the work and revisits the key parts of the research.

2 Background and Related Works

2.1 Blockchain and Cryptocurrencies

Blockchain technology had the most success with the crypto projects, but it is important to note that this is not its only use. Blockhains are still experimented with, but serious potential has been noticed with its use in healthcare systems [54] and even election fraud prevention [37]. Blockchain is a software that relies on the previous block to confirm the validity of the current one, and the chain goes so on until the first block. To deepen the concept further, not only is the oldest block the parent of all other blocks, but this information can be reached from all other blocks. This information is available to all users in the network and it required to be confirmed before a transaction is made. This concept is robust and transparent. Cryptocurrencies exist on the blockchain or can be rather considered the blockchain itself if we take into account that all the users from the blockcahin own a part of it which is represented with a numerical value of the currency. This technology allows for all the concepts previously discussed which are the key advantages of the cryptocurrencies. The advantage that was not previously talked about in this paper is the accessibility which is higher than with the case of fiat currencies. Cryptocurrencies are as a result flexible, can change ownership, and mostly fast. The speed of the transactions depends on the users requesting transactions at the moment, as someone still has to validate them. These are usually large nodes in the network, or put in other words large currency holders. Validators receive a percentage of each transaction they process, as this process requires power as well as the mining of the currency.

2.2 Long Short Term Memory Model

The strongest influence on the creation of artificial neural networks (ANN) had the brain of humans and the way in which it learns. The process of transmitting and interpreting signals by neurons is implemented as an algorithm. This type of network applies deep learning and can learn from examples.

RNNs are ANNs that provide two-way data flow and store the information within the network. Future outcomes are influenced by the previously obtained input results. Due to this characteristic, the RNNs are suitable for the time-series prediction problems. The memory mechanism is realized through the cell replacement in a traditional network by the memory cells from hidden layers. The structure of the memory cells is divided into three gates: input, output, and forget. This allows for selective storage and deletion (release) of the data. The first step is to filter the data through the forget gate f_t where rather the data should be discarded or not is decided. Equation (1) describes the forget gate f_t.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

for the forget gate f_c is used, with the range of values from 0 to 1, σ represents the sigmoid function, variable matrices are W_f and U_f, while the bias vector is b_f.

The following stem applies selection to data for storing in memory cells. The sigmoid function selects the values for renewal for a given input gate i_t by Eq. (2).

$$i_t = \sigma(W_i X_t + U_i h_{t-1} + b_i) \tag{2}$$

while the range for the i_t is the same as with the previously described f_t, and the b_i W_i, U_i are learnable parameters.

The function *tanh* provides the potential vectors for updating C_t from the Eq. (3)

$$C_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

in which the learnable parameters are b_c, W_c U_c.

After the selection of data for storing the given cell state C_t is calculated by Eq. (4)

$$C_t = F_t \odot C_{t-1} + i_t \odot C_t \tag{4}$$

where the signification and element-wise multiplication is $\odot$, date to be erased is C_{t-1}, the selected data is represented by $f_t \odot C_t$, while the $i_t \odot C_t$ shows the data to be stored in a memory cell C_t.

The hidden state h_T is defined the output gate o_t provided by Eq. (5).

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

where the range for o_t is the same as for the two previous gates, and the input gate learnable parameters are b_o W_o and U_o.

Lastly, the output value h_t is represented by the product of o_t and the *tanh* value of C_t shown in Eq. (6)

$$h_t = o_t \odot tanh(C_t) \tag{6}$$

2.3 Swarm Intelligence and Literature Review

The increase in popularity of machine learning reflected on swarm intelligence as well. The developments in these fields have resulted in solutions highly capable of solving problems of high complexity. Swarm intelligence solutions are inspired by the behaviors of species that exhibit swarm-like behavior. These behaviors can regard movement, search for food or road, communication, etc. The large number of units and cooperation towards the common goal translates well into algorithmic form as swarm algorithms are population-based, stochastic, and evolutionary.

The recent success of swarm intelligence is due to their ability to successfully and efficiently solve NP-hard problems. However, this cannot be achieved in a single iteration and such runs of the algorithm do not guarantee an optimal solution. Hence, this probability is increased progressively as the algorithm is allowed more iterations.

Furthermore, no algorithm performs well for each use case, as the no free lunch theorem [51] states. Consequentially, the researches experiment with various algorithms for various problems. Interesting use cases include medical diagnosis applications [15,21,25,33,42,49], wireless sensor network optimizations [5,10,13,46,57,66], stock price predictions [17], as well as intrusion detection [2,31,40,55,56,60,65] and plant classifying task [18].

Additionally, swarm intelligence solutions have become popular due to one more characteristic of theirs. Such solutions are highly adaptable and optimizable. The process that became a standard for further improvements of the basic solution is hybridization. Algorithms can be combined with other metaheuristic solutions to provide a higher performing algorithm or they can be combined with a machine learning solution to optimize its actions. Some examples are optimizing cloud computing scheduling, edge and fog computing [3,6,16,24,28,45,59], feature selection [9,20,23,32,36,47,61], dropout regularization [12], COVID-19 detection and fake news detection [26,58,62–64], tuning artificial neural networks [4,7,8,11,14,19,44], text clustering [22,48] and cryptocurrency price prediction as well [41].

3 Proposed Method

3.1 The Original Arithmetic Optimization Algorithm

Recently introduced arithmetic optimization algorithm was created by Abualigah et al. in 2021 [1]. This algorithm is population-based and stochastic. As mentioned in this work, the guarantee for optimal solution is reached through iterations. AOA is complex in terms of rules used as it employs different mathematical metaheuristics based on the situation. The arithmetic operations that are applied are addition, subtraction, multiplication, and division. The main influence of these operations is the control between exploration and exploitation phases. Note that the algorithm does not calculate the optimization problems' derivatives during their solution.

Randomly generated solutions make up the population in the initialization phase. The population is represented as a matrix from which the best is selected in every iteration, marked as best-obtained. Math optimizer accelerated (MOA) is used for calculation of the phase selection.

$$MOA(C_{Iter}) = Min + C_{Iter} \times (\frac{Max - Min}{M_{Iter}}) \tag{7}$$

for which the t-th iteration value of function is represented as $MOA(C_{Iter})$, current iteration is represented as C_{Iter}, while the M_{Iter} represents the maximum number of iterations in 1-M_{Iter}. Minimum and maximum of the accelerated functions are given as Min and Max, respectively.

Exploration is performed if the selected operations are division (D) or multiplication (M). These operations are suitable for this phase due to heir high distribution of values. Consequently, they cannot easily approach the solution for which the other two arithmetics are used. For the goal of exploration it is needed to search the domain randomly and find a near-optimal solution. Exploration is performed by following equations:

$$x_{i,j}(C_{Iter}+1) = \begin{cases} best(x_j) \div (MOP + \epsilon) \times ((UB_j - LB_j) \times \mu + LB_j), \ r_2 < 0.5 \\ best(x_j) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j) \ otherwise \end{cases} \tag{8}$$

where the i-th solution in the following iteration (x_i) is given as $x_i(C_{Iter}+1)$. The j-th position of the i-th solution at the current iteration is $j(C_{Iter})$. j-th position so far best-obtained solution is denoted as $best(x_j)$. A small integer number is given as ϵ symbol. For the upper and lower boundaries respectively are used UB_j and LB_j. The adjustment of the search process is performed by control parameter μ which has a fixed value of 0.5.

$$MOP(C_{Iter}) = 1 - \frac{C_{Iter}^{1/\alpha}}{M_{Iter}^{1/\alpha}} \tag{9}$$

in which the math optimizer probability coefficient is given as MOP, while the value of the function at the t-th iteration is given as MOP at C_{Iter}. Fixed value 5 is given to the α parameter that controls the accuracy of exploitation through the iterations.

Addition (A) and subtraction (S) as mentioned before are applied to the exploitation process. Due to their higher density of solutions they perform better over the previous two operations in the exploitation phase. The choice of A or S is conditioned by MOA.

$$x_{i,j}(C_{Iter}+1) = \begin{cases} best(x_j) - MOP \times ((UB_j - LB_j) \times \mu + LB_j), \ r_3 < 0.5 \\ best(x_j) + MOP \times ((UB_j - LB_j) \times \mu + LB_j) \ otherwise \end{cases} \tag{10}$$

The A and S search operators tend to avoidance of the local optima. Finding the optimal solution while maintaining the diversity of the solution is the main goal of these two operations. The pseudo-code of the AOA follows in the Algorithm 1.

3.2 Proposed Optimized LSTM

The AOA algorithm was implemented in a framework with a goal to tune the hyperparameters of the LSTM. The proposed method has been given the name LSTM-AOA. The ranges for the optimized parameters have been set to the following:

- number of neurons - $[20, 200]$, integer
- learning rate - $[0.0001, 0.01]$, continuous
- training epochs - $[100, 300]$, integer

Algorithm 1. Original AOA pseudo-code

```
Initialization of the parameters α, μ of the Arithmetic Optimization Algorithm
Random initialization of the positions of the solutions. (Solutions: i = 1, ..., N)
while C_Iter < M_Iter do
    Calculation of the Fitness Function (FF) for the given solutions
    Finding the best solution (Determined best so far).
    Updating the MOA value by the Eq. 7.
    Updating the MOP value by the Eq. 9.
    for i = 1 to Solutions do
        for j = 1 to Positions do
            Generation of random values between [0, 1](r1, r2, andr3)
            if r1 > MOA then
                Exploration phase
                if r2 > 0.5 then
                    (1) Application of the Division math operator (D"÷").
                    Updating of the i-th solutions' positions using the first rule in Eq. 8
                else
                    (2) Application of the Multiplication math operator (M"×").
                    Updating of the i-th solutions' positions using the second rule in Eq. 8
                end if
            else
                Exploitation phase
                if r3 > 0.5 then
                    (1) Application of the Subtraction math operator (S"−").
                    Updating the i-th solutions' positions using the first rule in Eq. 10
                else
                    (2) Application of the Addition math operator (A"+").
                    Updating the i-th solutions' positions using the second rule in Eq. 10
                end if
            end if
        end for
    end for
    C_Iter = C_Iter + 1
end while
Return the best solution (x)
```

– dropout rate - $[0.001, 0.01]$, continuous

The framework has been developed in tensorflow 2.0/keras, while remaining parameters have been bounded to keras' defaults, with the recurrent dropout equal to 0.01.

4 Experimental Findings and Results

This section aims to describe the conditions upon which the testing was performed as well as the tests that were performed. Furthermore, after the description of the testing process follows the comparison of the results to other tested solutions. The same experimental setup was applied as in [38] since this research was inspired by the mentioned work and firm grounds for comparison were required to be established.

4.1 Metrics

The used metrics for evaluation include mean absolute error (MAE), mean squared error (MSE), mean absolute percentage error (MAPE), and root mean squared error (RMSE).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{p} - p_i)^2 \tag{11}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{p} - p_i)^2} \tag{12}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{p} - p_i| \tag{13}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{\hat{p} - p_i}{p_i}| \tag{14}$$

in which the predicted price is $\hat{p}$, the actual price p_i, and the total number of observations is N.

4.2 Dataset

The dataset that was used in the experiments was retrieved from https://finance.yahoo.com/, that shows daily fluctuations of the Bitcoin cryptocurrency. The observed period was 1.1.2020. - 6.10.2022, and the close price for Bitcoin was used. First 70% of entries were used as training set, while the remaining 30% was utilized for testing, as shown in Fig. 1.

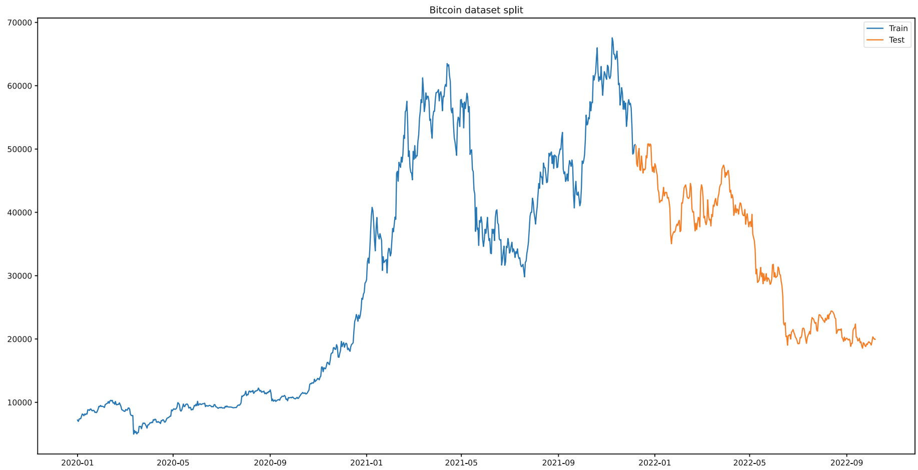

Fig. 1. Dataset visualization

4.3 Experiments

The suggested LSTM-AOA method has been put into the comparison with five other cutting-edge algorithms implemented in the same framework. The contenders were seeker optimization algorithm (SOA) [29], firefly algorithm (FA) [52], sine cosine algorithm (SCA) [1], bat algorithm (BA) [53] and artificial bee colony metaheuristics (ABC) [34]. The contender algorithms were implemented independently, with the default control parameter values retrieved from their respective publications. For clarity, the methods were named LSTM-SOA, LSTM-FA, LSTM-SCA, LSTM-BA and LSTM-ABC. All implemented metaheuristics have been executed with five solutions in the populace, with eight rounds in six independent runs. The experiments used 6 lags, while the forecasting was executed 3 steps ahead.

The results of all observed models are summarized in Tables 1 and 2, where the best achieved metric in each category is highlighted in bold text. Table 1 shows the error indicators for all evaluated models. The proposed LSTM-AOA method obtained superior level of performance, as it achieved the best values for all error indicators for one-step, two-step, three-step ahead as well as overall

Table 1. Indicators convergence of observed algorithms on Bitcoin dataset

	Error indicator	LSTM-AOA	LSTM-SOA	LSTM-FA	LSTM-SCA	LSTM-BA	LSTM-ABC
One-step ahead	R2	**0.982563**	0.980532	0.974923	0.979675	0.980360	0.978740
	MAE	**0.016094**	0.017160	0.018060	0.016316	0.016285	0.016555
	MSE	**0.000451**	0.000458	0.000599	0.000499	0.000469	0.000527
	RMSE	**0.021240**	0.021397	0.024473	0.022347	0.021648	0.022967
Two-step ahead	R2	**0.983242**	0.979163	0.977829	0.977322	0.975396	0.980789
	MAE	**0.015247**	0.018489	0.017562	0.017335	0.019820	0.016343
	MSE	**0.000429**	0.000495	0.000520	0.000542	0.000583	0.000469
	RMSE	**0.020723**	0.022258	0.022793	0.023277	0.024153	0.021663
Three-step ahead	R2	**0.983029**	0.980286	0.980588	0.978978	0.979813	0.979554
	MAE	**0.015625**	0.017808	0.016756	0.017403	0.017181	0.016479
	MSE	**0.000426**	0.000463	0.000454	0.000497	0.000488	0.000502
	RMSE	**0.020634**	0.021522	0.021316	0.022288	0.022083	0.022408
Overall results	R2	**0.982946**	0.979992	0.977766	0.978671	0.978546	0.979697
	MAE	**0.015655**	0.017819	0.017459	0.017018	0.017762	0.016459
	MSE	**0.000435**	0.000472	0.000524	0.000513	0.000513	0.000500
	RMSE	**0.020867**	0.021729	0.022897	0.022642	0.022654	0.022352

Table 2. Overall results of algorithms on Bitcoin dataset

Method	LSTM-AOA	LSTM-SOA	LSTM-FA	LSTM-SCA	LSTM-BA	LSTM-ABC
Best	**4.35E-04**	4.72E-04	5.24E-04	5.13E-04	5.13E-04	5.00E-04
Worst	**4.74E-04**	4.79E-04	5.25E-04	5.16E-04	7.71E-04	5.82E-04
Mean	**4.48E-04**	4.77E-04	5.25E-04	5.14E-04	6.17E-04	5.27E-04
Median	**4.35E-04**	4.79E-04	5.25E-04	5.13E-04	5.66E-04	5.00E-04
Std	1.80E-05	3.13E-06	**1.71E-07**	1.48E-06	1.11E-04	3.88E-05
Var	3.25E-10	9.78E-12	**2.94E-14**	2.19E-12	1.23E-08	1.51E-09

results. Similarly, the suggested LSTM-AOA obtained superior overall results as shown in Table 2 with respect to the best, worst, mean and median values. LSTM-SOA obtained the second best results, in front of LSTM-SCA and LSTM-FA.

Aiming to graphically show the supremacy of the LSTM-AOA method over the contending algorithms, Fig. 2 shows the convergence graph of the indicators, convergence of the objective function (classification error scale), box plots and violin plots of the objective function for all included metaheuristics, respectively. It is possible to note that LSTM-AOA converges significantly faster than other contenders. Finally, the Bitcoin closing price forecast for the best run of the LSTM-AOA model is shown in Fig. 3, that displays predictions for 3 steps ahead, overlapped over the actual close values.

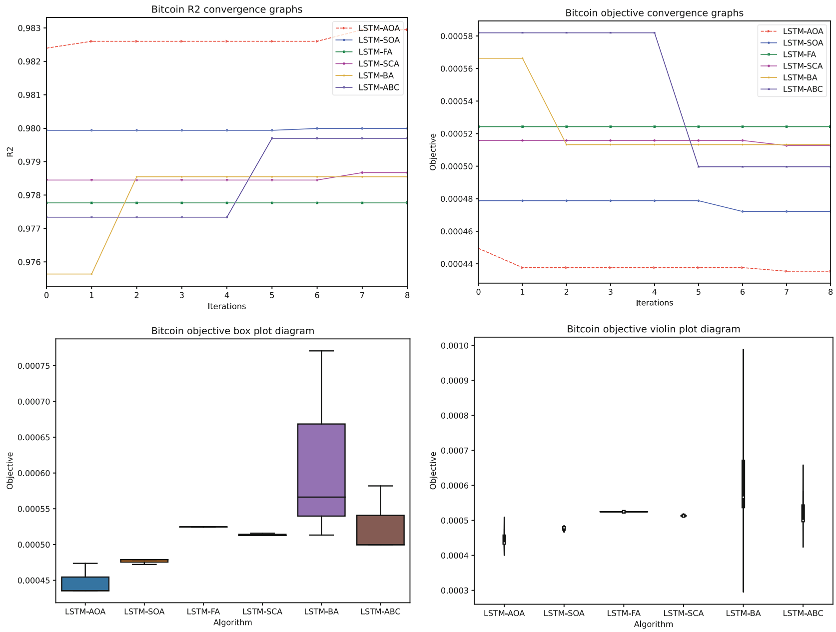

Fig. 2. Convergence graphs, box plot and violin diagrams for all observed methods

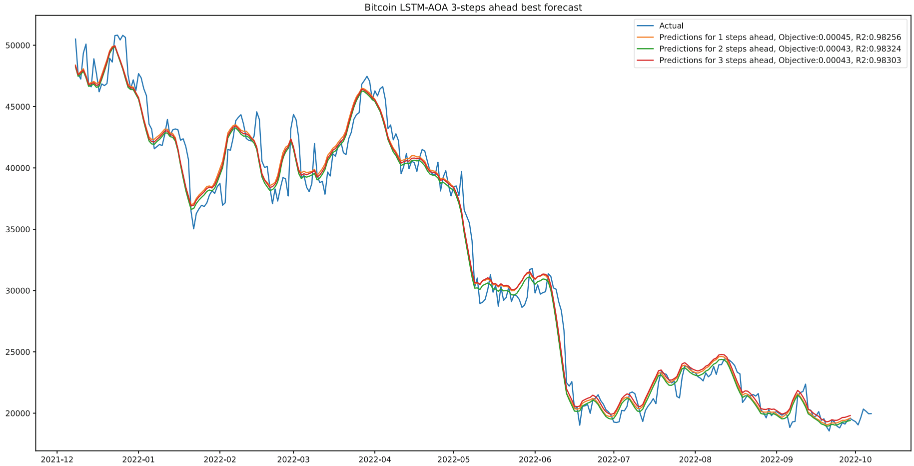

Fig. 3. The best predicted results obtained by LSTM-AOA method

5 Conclusion

The topic of cryptocurrencies still attracts interest and has yet to be explored. The price of the currency is the most interesting part, besides the technological improvements made available by blockcahin. This work aimed to provide a more sophisticated solution for predicting the prices. The AOA and LSTM hybrid prediction model proved successful over testing periods and fared better than its competitors. The improvements were achieved. Due to the variety of application of swarm metaheuristics, the authors tend to explore other solutions from the same field with this problem.

References

1. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A.H.: The arithmetic optimization algorithm. Computer methods in applied mechanics and engineering **376**, 113609 (2021)
2. AlHosni, N., Jovanovic, L., Antonijevic, M., Bukumira, M., Zivkovic, M., Strumberger, I., Mani, J.P., Bacanin, N.: The xgboost model for network intrusion detection boosted by enhanced sine cosine algorithm. In: International Conference on Image Processing and Capsule Networks. pp. 213–228. Springer (2022)
3. Antonijevic, M., Strumberger, I., Lazarevic, S., Bacanin, N., Mladenovic, D., Jovanovic, D.: Robust encrypted face recognition robot based on bit slicing and fourier transform for cloud environments. Journal of Electronic Imaging **31**(6), 061808 (2022)
4. Bacanin, N., Alhazmi, K., Zivkovic, M., Venkatachalam, K., Bezdan, T., Nebhen, J.: Training multi-layer perceptron with enhanced brain storm optimization metaheuristics. Comput. Mater. Contin **70**, 4199–4215 (2022)

5. Bacanin, N., Antonijevic, M., Bezdan, T., Zivkovic, M., Rashid, T.A.: Wireless sensor networks localization by improved whale optimization algorithm. In: Proceedings of 2nd International Conference on Artificial Intelligence: Advances and Applications. pp. 769–783. Springer (2022)
6. Bacanin, N., Antonijevic, M., Bezdan, T., Zivkovic, M., Venkatachalam, K., Malebary, S.: Energy efficient offloading mechanism using particle swarm optimization in 5g enabled edge nodes. Cluster Computing pp. 1–12 (2022)
7. Bacanin, N., Bezdan, T., Venkatachalam, K., Zivkovic, M., Strumberger, I., Abouhawwash, M., Ahmed, A.B.: Artificial neural networks hidden unit and weight connection optimization by quasi-refection-based learning artificial bee colony algorithm. IEEE Access **9**, 169135–169155 (2021)
8. Bacanin, N., Bezdan, T., Zivkovic, M., Chhabra, A.: Weight optimization in artificial neural network training by improved monarch butterfly algorithm. In: Mobile Computing and Sustainable Informatics, pp. 397–409. Springer (2022)
9. Bacanin, N., Petrovic, A., Zivkovic, M., Bezdan, T., Chhabra, A.: Enhanced salp swarm algorithm for feature selection. In: International Conference on Intelligent and Fuzzy Systems. pp. 483–491. Springer (2021)
10. Bacanin, N., Sarac, M., Budimirovic, N., Zivkovic, M., AlZubi, A.A., Bashir, A.K.: Smart wireless health care system using graph lstm pollution prediction and dragonfly node localization. Sustainable Computing: Informatics and Systems **35**, 100711 (2022)
11. Bacanin, N., Stoean, C., Zivkovic, M., Jovanovic, D., Antonijevic, M., Mladenovic, D.: Multi-swarm algorithm for extreme learning machine optimization. Sensors **22**(11), 4204 (2022)
12. Bacanin, N., Stoean, R., Zivkovic, M., Petrovic, A., Rashid, T.A., Bezdan, T.: Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization. Mathematics **9**(21), 2705 (2021)
13. Bacanin, N., Tuba, E., Zivkovic, M., Strumberger, I., Tuba, M.: Whale optimization algorithm with exploratory move for wireless sensor networks localization. In: International conference on hybrid intelligent systems. pp. 328–338. Springer (2019)
14. Bacanin, N., Vukobrat, N., Zivkovic, M., Bezdan, T., Strumberger, I.: Improved harris hawks optimization adapted for artificial neural network training. In: International Conference on Intelligent and Fuzzy Systems. pp. 281–289. Springer (2021)
15. Bacanin, N., Zivkovic, M., Al-Turjman, F., Venkatachalam, K., Trojovskỳ, P., Strumberger, I., Bezdan, T.: Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application. Scientific Reports **12**(1), 1–20 (2022)
16. Bacanin, N., Zivkovic, M., Bezdan, T., Venkatachalam, K., Abouhawwash, M.: Modified firefly algorithm for workflow scheduling in cloud-edge environment. Neural Computing and Applications **34**(11), 9043–9068 (2022)
17. Bacanin, N., Zivkovic, M., Jovanovic, L., Ivanovic, M., Rashid, T.A.: Training a multilayer perception for modeling stock price index predictions using modified whale optimization algorithm. In: Computational Vision and Bio-Inspired Computing, pp. 415–430. Springer (2022)
18. Bacanin, N., Zivkovic, M., Sarac, M., Petrovic, A., Strumberger, I., Antonijevic, M., Petrovic, A., Venkatachalam, K.: A novel multiswarm firefly algorithm: An application for plant classification. In: International Conference on Intelligent and Fuzzy Systems. pp. 1007–1016. Springer (2022)

19. Basha, J., Bacanin, N., Vukobrat, N., Zivkovic, M., Venkatachalam, K., Hubálovskỳ, S., Trojovskỳ, P.: Chaotic harris hawks optimization with quasi-reflection-based learning: an application to enhance cnn design. Sensors **21**(19), 6654 (2021)
20. Bezdan, T., Cvetnic, D., Gajic, L., Zivkovic, M., Strumberger, I., Bacanin, N.: Feature selection by firefly algorithm with improved initialization strategy. In: 7th Conference on the Engineering of Computer Based Systems. pp. 1–8 (2021)
21. Bezdan, T., Milosevic, S., Venkatachalam, K., Zivkovic, M., Bacanin, N., Strumberger, I.: Optimizing convolutional neural network by hybridized elephant herding optimization algorithm for magnetic resonance image classification of glioma brain tumor grade. In: 2021 Zooming Innovation in Consumer Technologies Conference (ZINC). pp. 171–176. IEEE (2021)
22. Bezdan, T., Stoean, C., Naamany, A.A., Bacanin, N., Rashid, T.A., Zivkovic, M., Venkatachalam, K.: Hybrid fruit-fly optimization algorithm with k-means for text document clustering. Mathematics **9**(16), 1929 (2021)
23. Bezdan, T., Zivkovic, M., Bacanin, N., Chhabra, A., Suresh, M.: Feature selection by hybrid brain storm optimization algorithm for covid-19 classification. Journal of Computational Biology (2022)
24. Bezdan, T., Zivkovic, M., Bacanin, N., Strumberger, I., Tuba, E., Tuba, M.: Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. Journal of Intelligent & Fuzzy Systems **42**(1), 411–423 (2022)
25. Bezdan, T., Zivkovic, M., Tuba, E., Strumberger, I., Bacanin, N., Tuba, M.: Glioma brain tumor grade classification from mri using convolutional neural networks designed by modified fa. In: International conference on intelligent and fuzzy systems. pp. 955–963. Springer (2020)
26. Budimirovic, N., Prabhu, E., Antonijevic, M., Zivkovic, M., Bacanin, N., Strumberger, I., Venkatachalam, K.: Covid-19 severity prediction using enhanced whale with salp swarm feature classification. Computers, Materials and Continua pp. 1685–1698 (2022)
27. Chowdhury, M.S.R., Damianov, D.S., Elsayed, A.H.: Bubbles and crashes in cryptocurrencies: Interdependence, contagion, or asset rotation? Finance Research Letters **46**, 102494 (2022)
28. Cologni, A., Manera, M.: The asymmetric effects of oil shocks on output growth: A markov–switching analysis for the g-7 countries. Economic Modelling **26**(1), 1–29 (2009)
29. Dai, C., Zhu, Y., Chen, W.: Seeker optimization algorithm. In: International conference on computational and information science. pp. 167–176. Springer (2006)
30. Hansun, S., Wicaksana, A., Khaliq, A.Q.: Multivariate cryptocurrency prediction: comparative analysis of three recurrent neural networks approaches. Journal of Big Data **9**(1), 1–15 (2022)
31. Jovanovic, D., Antonijevic, M., Stankovic, M., Zivkovic, M., Tanaskovic, M., Bacanin, N.: Tuning machine learning models using a group search firefly algorithm for credit card fraud detection. Mathematics **10**(13), 2272 (2022)
32. Jovanovic, D., Marjanovic, M., Antonijevic, M., Zivkovic, M., Budimirovic, N., Bacanin, N.: Feature selection by improved sand cat swarm optimizer for intrusion detection. In: 2022 International Conference on Artificial Intelligence in Everything (AIE). pp. 685–690. IEEE (2022)
33. Jovanovic, L., Zivkovic, M., Antonijevic, M., Jovanovic, D., Ivanovic, M., Jassim, H.S.: An emperor penguin optimizer application for medical diagnostics. In: 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC). pp. 191–196. IEEE (2022)

34. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of global optimization **39**(3), 459–471 (2007)
35. Kayarvizhy, N., Kanmani, S., Uthariaraj, R.: Ann models optimized using swarm intelligence algorithms. WSEAS Transactions on Computers **13**(45), 501–519 (2014)
36. Latha, R., Saravana Balaji, B., Bacanin, N., Strumberger, I., Zivkovic, M., Kabiljo, M.: Feature selection using grey wolf optimization with random differential grouping. Comput. Syst. Sci. Eng. **43**(1), 317–332 (2022)
37. Moura, T., Gomes, A.: Blockchain voting and its effects on election transparency and voter confidence. In: Proceedings of the 18th annual international conference on digital government research. pp. 574–575 (2017)
38. Patel, M.M., Tanwar, S., Gupta, R., Kumar, N.: A deep learning-based cryptocurrency price prediction scheme for financial institutions. Journal of information security and applications **55**, 102583 (2020)
39. Peterson, T.: Why bitcoin dominates. Available at SSRN 3417334 (2019)
40. Petrovic, A., Bacanin, N., Zivkovic, M., Marjanovic, M., Antonijevic, M., Strumberger, I.: The adaboost approach tuned by firefly metaheuristics for fraud detection. In: 2022 IEEE World Conference on Applied Intelligence and Computing (AIC). pp. 834–839. IEEE (2022)
41. Petrovic, A., Strumberger, I., Bezdan, T., Jassim, H.S., Nassor, S.S.: Cryptocurrency price prediction by using hybrid machine learning and beetle antennae search approach. In: 2021 29th Telecommunications Forum (TELFOR). pp. 1–4. IEEE (2021)
42. Prakash, S., Kumar, M.V., Ram, S.R., Zivkovic, M., Bacanin, N., Antonijevic, M.: Hybrid glfil enhancement and encoder animal migration classification for breast cancer detection. Comput. Syst. Sci. Eng. **41**(2), 735–749 (2022)
43. Prayogo, G.: Bitcoin, regulation and the importance of national legal reform. Asian Journal of Law and Jurisprudence **1**(1), 1–9 (2018)
44. Salb, M., Bacanin, N., Zivkovic, M., Antonijevic, M., Marjanovic, M., Strumberger, I.: Extreme learning machine tuning by original sine cosine algorithm. In: 2022 IEEE World Conference on Applied Intelligence and Computing (AIC). pp. 143–148. IEEE (2022)
45. Sreekanth, G., Ahmed, S.A.N., Sarac, M., Strumberger, I., Bacanin, N., Zivkovic, M.: Mobile fog computing by using sdn/nfv on 5g edge nodes. Comput. Syst. Sci. Eng. **41**(2), 751–765 (2022)
46. Strumberger, I., Bezdan, T., Ivanovic, M., Jovanovic, L.: Improving energy usage in wireless sensor networks by whale optimization algorithm. In: 2021 29th Telecommunications Forum (TELFOR). pp. 1–4. IEEE (2021)
47. Strumberger, I., Rakic, A., Stanojlovic, S., Arandjelovic, J., Bezdan, T., Zivkovic, M., Bacanin, N.: Feature selection by hybrid binary ant lion optimizer with covid-19 dataset. In: 2021 29th Telecommunications Forum (TELFOR). pp. 1–4. IEEE (2021)
48. Strumberger, I., Tuba, E., Bacanin, N., Zivkovic, M., Beko, M., Tuba, M.: Designing convolutional neural network architecture by the firefly algorithm. In: 2019 International Young Engineers Forum (YEF-ECE). pp. 59–65. IEEE (2019)
49. Tair, M., Bacanin, N., Zivkovic, M., Venkatachalam, K.: A chaotic oppositional whale optimisation algorithm with firefly search for medical diagnostics. Comput. Mater. Contin **72**, 959–982 (2022)

50. Wątorek, M., Drożdż, S., Kwapień, J., Minati, L., Oświęcimka, P., Stanuszek, M.: Multiscale characteristics of the emerging global cryptocurrency market. Physics Reports **901**, 1–82 (2021)
51. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE transactions on evolutionary computation **1**(1), 67–82 (1997)
52. Yang, X.S.: Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. pp. 169–178. Springer (2009)
53. Yang, X.S.: Bat algorithm for multi-objective optimisation. International Journal of Bio-Inspired Computation **3**(5), 267–274 (2011)
54. Zhang, M., Ji, Y.: Blockchain for healthcare records: A data perspective. PeerJ Preprints **6**, e26942v1 (2018)
55. Zivkovic, M., Bacanin, N., Arandjelovic, J., Rakic, A., Strumberger, I., Venkatachalam, K., Joseph, P.M.: Novel harris hawks optimization and deep neural network approach for intrusion detection. In: Proceedings of International Joint Conference on Advances in Computational Intelligence. pp. 239–250. Springer (2022)
56. Zivkovic, M., Bacanin, N., Arandjelovic, J., Strumberger, I., Venkatachalam, K.: Firefly algorithm and deep neural network approach for intrusion detection. In: Applications of Artificial Intelligence and Machine Learning, pp. 1–12. Springer (2022)
57. Zivkovic, M., Bacanin, N., Tuba, E., Strumberger, I., Bezdan, T., Tuba, M.: Wireless sensor networks life time optimization based on the improved firefly algorithm. In: 2020 International Wireless Communications and Mobile Computing (IWCMC). pp. 1176–1181. IEEE (2020)
58. Zivkovic, M., Bacanin, N., Venkatachalam, K., Nayyar, A., Djordjevic, A., Strumberger, I., Al-Turjman, F.: Covid-19 cases prediction by using hybrid machine learning and beetle antennae search approach. Sustainable Cities and Society **66**, 102669 (2021)
59. Zivkovic, M., Bezdan, T., Strumberger, I., Bacanin, N., Venkatachalam, K.: Improved harris hawks optimization algorithm for workflow scheduling challenge in cloud-edge environment. In: Computer Networks, Big Data and IoT, pp. 87–102. Springer (2021)
60. Zivkovic, M., Jovanovic, L., Ivanovic, M., Bacanin, N., Strumberger, I., Joseph, P.M.: Xgboost hyperparameters tuning by fitness-dependent optimizer for network intrusion detection. In: Communication and Intelligent Systems, pp. 947–962. Springer (2022)
61. Zivkovic, M., Jovanovic, L., Ivanovic, M., Krdzic, A., Bacanin, N., Strumberger, I.: Feature selection using modified sine cosine algorithm with covid-19 dataset. In: Evolutionary Computing and Mobile Sustainable Networks, pp. 15–31. Springer (2022)
62. Zivkovic, M., Petrovic, A., Bacanin, N., Milosevic, S., Veljic, V., Vesic, A.: The covid-19 images classification by mobilenetv3 and enhanced sine cosine metaheuristics. In: Mobile Computing and Sustainable Informatics, pp. 937–950. Springer (2022)
63. Zivkovic, M., Petrovic, A., Venkatachalam, K., Strumberger, I., Jassim, H.S., Bacanin, N.: Novel chaotic best firefly algorithm: Covid-19 fake news detection application. In: Advances in Swarm Intelligence, pp. 285–305. Springer (2023)
64. Zivkovic, M., Stoean, C., Petrovic, A., Bacanin, N., Strumberger, I., Zivkovic, T.: A novel method for covid-19 pandemic information fake news detection based on the arithmetic optimization algorithm. In: 2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). pp. 259–266. IEEE (2021)

65. Zivkovic, M., Tair, M., Venkatachalam, K., Bacanin, N., Hubálovskỳ, Š., Trojovskỳ, P.: Novel hybrid firefly algorithm: an application to enhance xgboost tuning for intrusion detection classification. PeerJ Computer Science **8**, e956 (2022)
66. Zivkovic, M., Zivkovic, T., Venkatachalam, K., Bacanin, N.: Enhanced dragonfly algorithm adapted for wireless sensor network lifetime optimization. In: Data intelligence and cognitive informatics, pp. 803–817. Springer (2021)