

Article

Return Rate Prediction in Blockchain Financial Products Using Deep Learning

Noura Metawa ^{1,2}, Mohamemd I. Alghamdi ³, Ibrahim M. El-Hasnony ⁴ and Mohamed Elhoseny ^{4,*}

¹ College of Business Administration, University of Sharjah, Sharjah P.O. Box 27272, United Arab Emirates; nmetawa@sharjah.ac.ae

² Faculty of Commerce, Mansoura University, Mansoura 35516, Egypt

³ Department of Computer Science, Al-Baha University, Al-Bahah 1988, Saudi Arabia; mialmushilah@bu.edu.sa

⁴ Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt; ibrahimhesin2005@mans.edu.eg

* Correspondence: melhoseny@ieee.org; Tel.: +9-715-027-254-41

Abstract: Recently, bitcoin-based blockchain technologies have received significant interest among investors. They have concentrated on the prediction of return and risk rates of the financial product. So, an automated tool to predict the return rate of bitcoin is needed for financial products. The recently designed machine learning and deep learning models pave the way for the return rate prediction process. In this aspect, this study develops an intelligent return rate predictive approach using deep learning for blockchain financial products (RRP-DLBFP). The proposed RRP-DLBFP technique involves designing a long short-term memory (LSTM) model for the predictive analysis of return rate. In addition, Adam optimizer is applied to optimally adjust the LSTM model's hyperparameters, consequently increasing the predictive performance. The learning rate of the LSTM model is adjusted using the oppositional glowworm swarm optimization (OGSO) algorithm. The design of the OGSO algorithm to optimize the LSTM hyperparameters for bitcoin return rate prediction shows the novelty of the work. To ensure the supreme performance of the RRP-DLBFP technique, the Ethereum (ETH) return rate is chosen as the target, and the simulation results are investigated in different measures. The simulation outcomes highlighted the supremacy of the RRP-DLBFP technique over the current state of art techniques in terms of diverse evaluation parameters. For the MSE, the proposed RRP-DLBFP has 0.0435 and 0.0655 compared to an average of 0.6139 and 0.723 for compared methods in training and testing, respectively.

Keywords: blockchain; financial products; predictive model; deep learning; Adam optimizer; LSTM model



Citation: Metawa, N.; Alghamdi, M.I.; El-Hasnony, I.M.; Elhoseny, M. Return Rate Prediction in Blockchain Financial Products Using Deep Learning. *Sustainability* **2021**, *13*, 11901. <https://doi.org/10.3390/su132111901>

Academic Editors: Muhammad Fazal Ijaz and Abu ul Hassan Sarwar Rana

Received: 5 September 2021

Accepted: 25 October 2021

Published: 28 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, economic globalization has rapidly developed, and together, different aspects restraining industrial development have been overcome with the fastest-developing resources. Rapid growth in economic markets has been observed [1,2]. The media of economic development determine economic marketplaces. It controls the allocation of the entire public and economic scheme while becoming an important part of economic development [3,4]. The global expansion of the internet has led to the growth of many internet-based financial products such as Baidu Economic Management, Yu'EBao, etc.; and this growth has an important impact on society. Recently, a new internet economics scheme developed by world influences such as peer-to-peer (P2P), crowdfunding, digital currency, and Blockchain may play a major part in developing the worldwide financial marketplace [5]. Blockchain models are determined as a systematic advance where the interference has greater impacts by transmitting the function of businesses from centralized to decentralized forms. It also always changes the untrustworthy agent without needing

entity-based modules. Instead [6], it alters the methods in which all transactions are identified and activates huge potential in various sectors such as Decentralized Autonomous organizations (DAC) and Multi-Party Computation (MPC) in the government sector. It consist of 3 evolutionary stages namely Blockchain (1.0, 2.0, & 3.0). In the beginning, Blockchain 1.0 was determined to be the commercial application using digital payment, money transfer, and remittance that acquired hugely-dispersed applications of derivatives and Bitcoin. Later, Blockchain 2.0 is used by the contract, entire financial challenges, markets, and financial fields that are extensively employed, not just for minor cash transactions. Lastly, Blockchain 3.0 represents application through the marketplace and money, particularly in the areas of health, science, government, etc. [7,8].

Blockchain technology uses decentralized storage to store massive amounts of data linked to the current blocks for the initial blocks using intelligent contracts. InterPlanetary File System (IPFS), BigchainDB, LitecoinDB, MoneroDB, Swarm, SiacoinDB, are used through decentralized databases in everyday instances [9–11]. The IPFS is a distributed, peer to peer and decentralized database that is transmitted and linked standard file. IPFS is a huge storage network that is exploited via blockchain method for IoTs software for maximal efficacy. Now, some research is associated with the proposed method that was studied. Using a method developed by the researcher, the writers in [12] used a fixed proportion for different character traits. They utilize the support vector machine (SVM) method for identifying and classifying tasks. Eventually, the experiment result shows that the presented approach has attained greater performance. While in [13], the authors discovered the Taiwanese stock marketplaces and applied SVM dependent genetic algorithm (GA) method. The previous study used the POS system to solve the optimum privacy portfolio method achieved by higher performances. Authors in [14] employed a group of ML approaches to process the prediction assessment on the nikkei 225 indexes. Hence, the result shows that SVM gained optimum results amongst these four modules.

The yield rate of bitcoin is the subject of [15], which collects data from 2 June 2016, to 30 December 2018, totaling 943 pieces. PSO least-squares vector approaches (PSOLSSVR) are adapted to perform empirical analyses and model simulations on the collected information and conclude that the PSO neural network (BPNN), SVM machine learning (ML), and particle swarm optimization (PSO) least-square vector approaches have an optimal appropriate effect. The generative adversarial network (GAN)-MLP model is used in [16] to develop a new return rate forecasting approach for Blockchain financial goods. Blockchain financial goods' intelligent return rates can be predicted using the suggested system. Using the proposed method, stock price closings can be predicted with high accuracy by providing historical stock price information.

In [17], price predictions are implemented by the two ML approaches such as Logistic Regression (LR) and SVM, consisting of everyday ether cryptocurrency closing price through a time sequence. Various window distances are employed in ether cryptocurrency price predictions through the filter using different weights coefficients. In the training stage, a cross-validated technique is employed to construct a better accuracy method autonomous of the dataset based on daily collected data. The optimal least-square support vector machine (OLSSVM) algorithm was used by Sivaram et al. [18] to develop an effective return rate prediction strategy for Blockchain financial products. The LSSVM constraint optimization was carried out by combining differential evolution (DE) with grey wolf optimization (GWO), resulting in the OGWO strategy, the best one. Hybridization techniques are used to eliminate GWO's worst local problems while also increasing the diversity of the population.

The authors in [19] show that Bitcoin prices demonstrate long-term memory, though its trend gets reduced over time. It is observed that Bitcoin can be defined effectively using a random walk, displaying the sign of market maturity emerging; contrastingly, other cryptocurrencies such as Ethereum and Ripple offer evidence of increasing underlying memory behavior. The authors in [20] employed a battery of statistical tests to determine the price leadership among the two cryptocurrencies, namely Bitcoin and Ethereum.

This study designs an intelligent return rate predictive approach using deep learning for blockchain financial products (RRP-DLBFP). The proposed RRP-DLBFP technique involves developing a long short-term memory (LSTM) model for the predictive analysis of return rate. Adam optimizer is also used to fine-tune the hyperparameters of the LSTM model so that it can make better predictions with more accuracy. In addition, the learning rate of the LSTM model is adjusted using the oppositional glowworm swarm optimization (OGSO) algorithm. To ensure the supreme performance of the RRP-DLBFP technique, the Ethereum (ETH) returns are chosen as the goal, and various simulation results are examined. In short, the paper contribution is listed here.

- Design a new return rate predictive model using RRP-DLBFP for blockchain financial product
- Develop an LSTM model for the predictive analysis of return rate
- Propose an Adam optimizer to adjust the hyperparameters of the LSTM model optimally
- Design an OGSO algorithm for the optimal adjustment of learning rate in the LSTM model
- Validate the performance of the RRP-DLBFP technique under several aspects

The remainder of this paper is divided into five sections. Section 2 discusses the related studies to the literature on return rate prediction models. Section 3 introduces and summarizes the blockchain method, deep learning models, and Adam algorithm. Section 4 details the proposed RRP-DLBFP. Section 5 discusses the experiment's results. Section 6 summarizes and concludes the paper.

2. Related Work

Using machine learning algorithms to predict Bitcoin's price is a relatively new area of research. It was possible to make an 89 percent profit in three months by employing the [21] developed latent source model to predict Bitcoin's price. Text data gleaned from social media and other sources have also been used to predict Bitcoin's price. Three studies looked into sentiment analysis by combining support vector machines (SVMs) with Wikipedia view frequency and hash rate (HashN). Authors in [22] looked into the connection between Bitcoin's price and the number of tweets and Google Trends searches for the term "Bitcoin". As in [23], the authors anticipated transaction volume rather than Google Trends views for Bitcoin price in their analysis. The limitation of this study is the small sample size and the propensity for misinformation to spread over many discussion boards, media channels such as Twitter or, which artificially inflate and deflate prices. In [24], liquidity is extremely scarce on Bitcoin exchanges. As a result, market manipulation is more likely. As a result, opinions expressed on social media will be ignored. For the analysis of the Bitcoin Blockchain, researchers in [25] employed support vector machines (SVMs) and artificial neural networks. They found that traditional ANNs were 55 percent accurate in forecasting the Bitcoin price. They concluded that Blockchain data on its own has limited predictability. Prediction accuracy of above 97% was noted by [26], who used Blockchain data with Random Forests, SVM, and Binomial Generalised Linear Model to make their predictions. However, they did not cross-validate their models, which limited the findings' generalization. It has been shown that network hash rate, search engine views, and mining difficulty positively correlate with Bitcoin price using wavelets [27,28]. CoinDesk's research is based on these findings and incorporates data from the Blockchain such as hash rate and difficulty into its study. Similar financial-based prediction jobs such as stock prediction can be compared to Bitcoin price prediction. Several research groups have used the Multilayer Perceptron (MLP) for stock price prediction [29]. The MLP can study a single observation, but only in one step at a time [30]. A recurrent neural network (RNN) uses a context layer to store the output from each layer before looping back to it. In contrast to the MLP, the network now has some memory. The network's length is referred to as the "temporal window length" (TWL). This study finds that internal states play a considerable role in the series' temporal relationship, explicitly represented by [31]. Authors in [32] utilized a genetic algorithm and an RNN for network optimization; they successfully predicted

stock returns. However, there are other types of RNNs, such as (LSTM) the Long Short Term Memory network. While Elman RNN can remember and retrieve data regardless of the significance or weight of a feature, these models may also selectively remember and forget data. In [33], for a time series prediction challenge, authors used an LSTM and discovered that it worked just as well as an RNN. This kind of model is likewise utilized in this instance. The large amount of computation required while training the RNN and the LSTM is a drawback. In other words, training 50 individual MLP models with a 50-day network is comparable. Since NVIDIA released its CUDA framework in 2006, many applications have benefited from the GPU's extraordinary parallelism, including machine learning. Authors in [34] stated that using a GPU instead of a CPU accelerated the testing and training of their ANN model three times. Similarly, in [35], authors found that putting an SVM on a GPU was eighty times faster than using a CPU-based SVM method for classification. In addition, the CPU implementation took nine times as long to train. Training a deep neural network for image identification on a GPU instead of a CPU resulted in forty times quicker training speeds for [36]. LSTM and RNN models use both the CPU and GPU because of the obvious advantages of doing so. In the automotive components supply chain, rough set theory (RST) and ABC analysis were proposed to combine to verify inventory for demand forecasts and ordering decisions. In the initial stage, the customer places a specific order with the reseller for specific spare parts. After that, a classification is created using the ABC analysis of the demand projection for each time. Next, the average weighted approach based on the mileage of the cars registered in the workshops uses each period's starting and ending mileage. This was accomplished by performing an ABC analysis and then creating an RST model to forecast which ABC group each element will belong to in the future. A method for controlling spare parts inventory based on machine learning for demand forecasting has been proposed [37]. Pre-processing, weight determination, and extraction are all parts of the strategy. It is necessary to train historical forecasting examples for each subset corresponding to the effect mechanism models before extracting the data using ELM and SVM learning methods. By equalizing the accuracy of the two models, the prediction model is firm-minded. However, the upkeep of weather-related spare parts is complicated. As mentioned in [38], a new hybrid model, abbreviated as WT-Adam-LSTM, uses Adam and wavelet transforms to improve LSTM neural network's price predicting accuracy. When nonlinear electricity price sequences are decomposed using a wavelet transform, the variance of the processed data will be more stable. This will allow algorithms such as Adam, which is one of the most efficient stochastic gradient-based optimizers, and LSTM to capture the behaviour of electricity prices better.

3. Methodology

3.1. Blockchain

The term "blockchain" [39] refers to a data structure that sequentially records transactions and facilitates them as a distributed record set. Then, it is divided into two sections: header and transaction, and it stores information about transaction specifics. In these data, you'll find both the source and destination addresses included as well. Using a cryptographic digest, each block generates its unique ID. The header stores the hash of the opposing block and thereby links the blocks together. It is primarily for marketing purposes that the structure is dubbed "Blockchain." When this link is viewed differently, it is denoted as a partial hash collision, which requires significant computing power to determine the hash function. Since all block references are predecessors, a change in a single bit could change the corresponding hash. It should be recalculated in the decreased order at the end. The presence of a lengthy chain, when it grows as a block, is unchangeable and ensures the security of preserved transactions, as is well-established.

The immutability, auditability, and nonrepudiation of transactions are all guaranteed to be possible using blockchain technology. As a huge distributed ledger, this technology ensures digital transactions are secure and private. It denotes the route to gaining consensus

among disparate groups of individuals. Security of transaction data is generally the responsibility of both banks and notaries, referred to as ‘trusted third parties’. As a result of this model’s registries being viewed as public and furnished in a decentralized manner via vast system applicants, such entities are eliminated.

As shown in Figure 1 [39], the bitcoin network keeps tabs on the major operations of all the nodes in use at every level. As previously said, it is an overlay network built on top of another one. It also creates new levels of network abstraction with innovative security benefits that are then deployed.

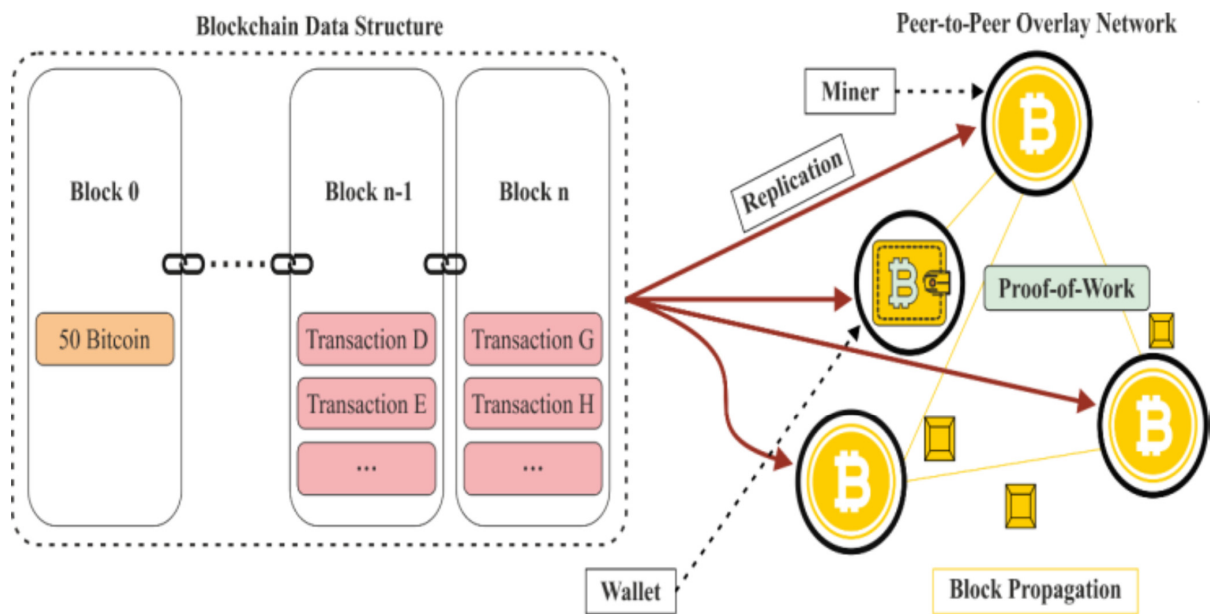


Figure 1. Overview of the Bitcoin system of ledgers.

3.2. LSTM

RNN neural networks are effective models, and they differ from standard feed-forward networks since their neuron connections are not limited to only one direction. In other words, neurons in an RNN network can send information to a previous or the same layer. Nodes between layers are disconnected in a typical neural network model since the layers are all connected. This general neural network can’t solve numerous complex problems. Predicting the next word in a sentence requires using the preceding word since sentences do not contain independent words [40]. The RNN is a cyclic neural network, which means that the current output is also linked to the preceding result. Because the network remembers earlier information, it can use that knowledge in calculating the current production, which will be connected to all of the nodes in the network. Thus, the hidden layer’s input includes both the input layer’s output and the hidden layer’s output from an earlier time.

As shown in Figure 2, this neural network chunk A receives the input values and calculates the previous output at each step. It then outputs the result. It is like having many copies of the same neural network in your brain. RNN can theoretically process any length of sequence data. Memory blocks instead of neurons are used in LSTM networks, with each memory block comprising gates that regulate the output and status of the block. It also has a memory for the most recent sequence, as well as a smarter-than-classical-neuron-like component. A sigmoid activation unit controls each gate in the block, which works with the input sequence.

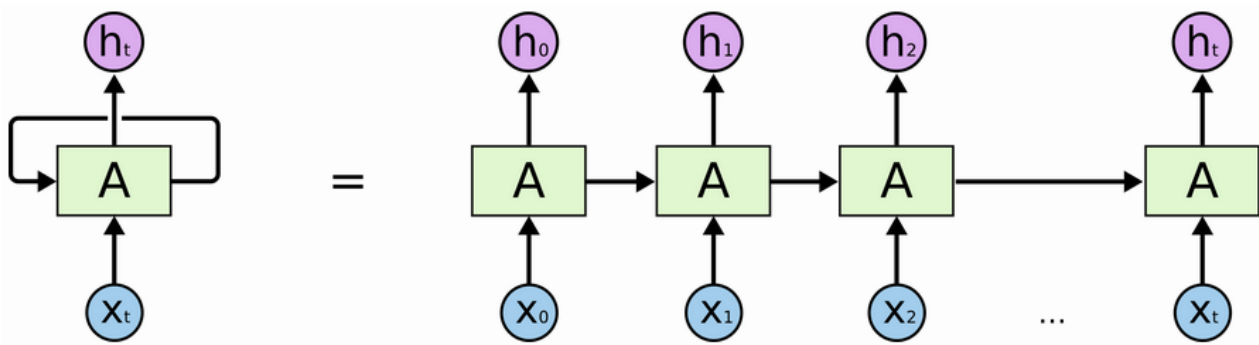


Figure 2. A recurrent neural network that has been unrolled.

Depending on whether they are activated, the state changes, and new information enters the block. A unit has three different kinds of gates (see Figure 3) [41].

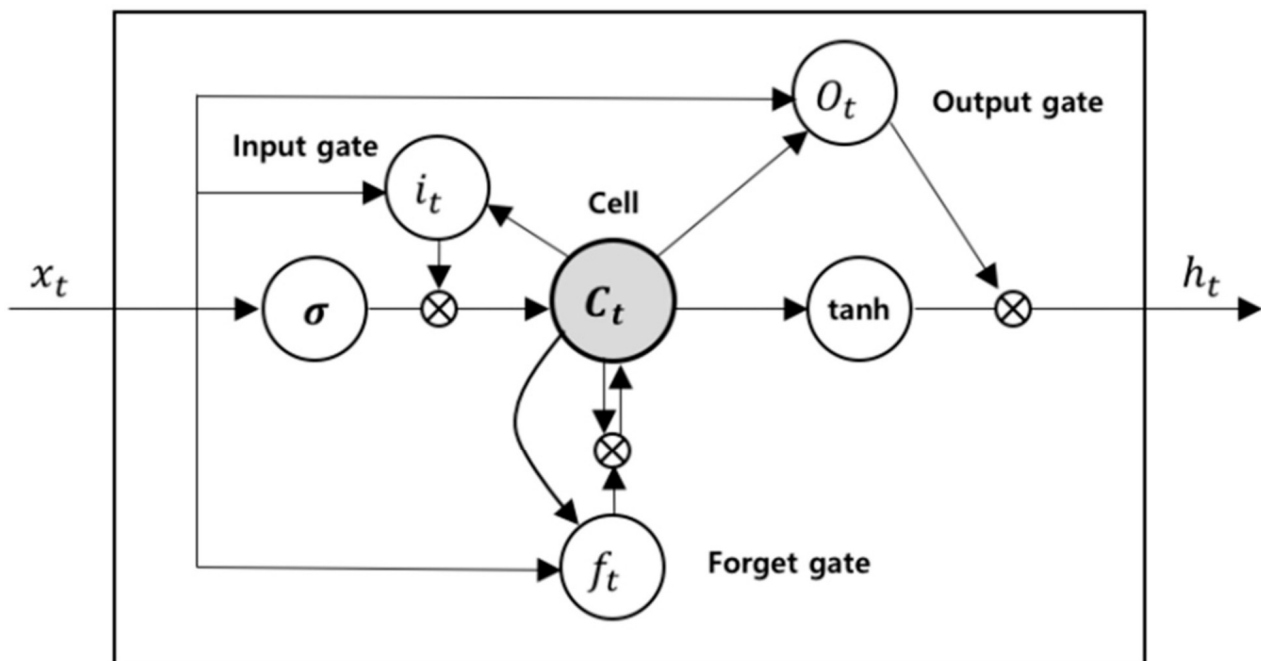


Figure 3. Gated long-short-term memory (LSTM) cell.

3.3. Adam Optimizer

Adam [42] is a stochastic objective function first-order gradient-based optimization technique based on adaptive estimations of lower-order moments. The method is easy to implement, has excellent computational efficiency, requires little memory, and rescales the gradient's diagonal. It is also a good fit for problems with a lot of information or parameters. Using Algorithm 1, we can see Adam's pseudocode [11].

After finding the parameters, β_1 , α , β_2 , and the random objective function $f(\theta)$, we must initialize the first-moment vector, the parameter vector, the time step, and the second-moment vector as given in the procedure. In order to reach convergence of the parameter, the loop iteratively updates the various components of the model until convergence is reached.

Algorithm 1. The proposed method for optimizing the LSTM model's parameters. The elementwise square g_t^2 is represented by $g_t \odot g_t$. Default machine learning settings that are effective so far include $\beta_1 = 0.9, \alpha = 0.001, \epsilon = 10^{-8}$. and $\beta_2 = 0.999$ and when working with vectors, you must always do things element by element. With β_2^t and β_1^t we denote β_2 and β_1 to the power t .

Inout: α : sizeOfstep, $f(\theta)$: Stochastic objective function uses parameters $\theta, \beta_2, \beta_1 \in [0, 1]$:
 Estimates with exponential decay rate for the instant, θ_0 = Initial vector

$m_0 = \mathbf{0}$ (initial value of first moment vector)

$t_0 = \mathbf{0}$ (initial value of second moment vector)

$t = 0$ (initial timestep)

Output $\hat{\theta}$, (Return parameters θ_t)

1. Start
 2. loop θ_t not converged do
 3. $t = t + 1$
 4. $g_t = \nabla_{\theta} f_t(\theta_{t-1})$ (Gradients at time t of stochastic objective)
 5. $m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (First moment estimate updated)
 6. $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Second raw moment estimate updated)
 7. $\hat{m}_t = m_t / (1 - \beta_1^t)$ (New first moment estimate updated)
 8. $\hat{v}_t = v_t / (1 - \beta_2^t)$ (New second raw moment estimate updated)
 9. $\theta_t = \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Parameters updated)
 10. End loop
 11. Return θ_t (Resulting parameters)
 12. Stop
-

4. The Proposed RRP-DLBFP Model Design

In this study, an RRP-DLBFP technique is designed to predict the return rate of financial blockchain products. The LSTM learns the dependency that ranges amongst arbitrary longer time intervals. An LSTM resolves the reducing gradient issue by replacing a typical neuron with a challenging LSTM unit framework. The LSTM unit develops in the application of related nodes. Necessary units of the LSTM framework [10] are determined under:

Constant error carousel (CEC): A significant element with recurrent link of unit weight. The recurrent link depicts a feedback loop at the time step as 1. The CEC's activation is an internal state which helps memory of prior information.

Input Gate (IG): The multiplicative units which protect the data secured from the CEC in unwanted input interruption.

Output Gate (OG): The multiplicative units protect another unit in an interference with the data stored in CEC.

An input-and-output-gates control activates the CEC. During the training stage, an input gate proposes training to data inside the CEC. An input gate has been assigned with a value of zero. Similarly, an output gate learns the time to release a data flow from the CEC. If the gate is closed, activation is applied inside the memory cell. It activates the error signals to flow over issues of the reducing gradient.

The structure of LSTM units contains a forget gate which is utilized for residual issues. The fundamental element of LSTM units is given under.

Input: An LSTM unit executes the current vector demonstrated as x_n , and the output saved in the last step is signified as h_{n-1} . The weighted inputs were summarized and changed by tanh activation that was demonstrated in z_n .

IG: This gate reads x_n and h_{n-1} , computes the weighted sum, and applies sigmoid activation. So, the outcomes were improved with z_n , and input flow has been given as the memory cell.

Forget gate (FG): When the network introduces a new order, afterward, a forget gate reads x_n and h_{n-1} and applies the sigmoid activation to the weighted input. Lastly, f_n is improved as a cell state from the preceding time step, in which s_{n-1} has been stimulated to forget the unimportant memory data.

Memory cell: The CEC and recurrent edge can be included with unit weights. The present cell state s_n has been computed as removing unwanted data in the previous time step and attaining the relevant data in the current input.

OG: This gate was utilized to the weighted sum of x_n and h_{n-1} and executed sigmoid activation to manage the LSTM blocks' data flow.

Output: The simulation result of an LSTM unit h_n is estimated by altering the cell state c_n with $\tan h$ and improving the output gate. The working rule of LSTM is defined as the implemented function:

$$i_n = \sigma(W_i \cdot [w_n, h_{n-1}] + b_i) \quad (1)$$

$$f_n = \sigma(W_f \cdot [w_n, h_{n-1}] + b_f) \quad (2)$$

$$o_n = \sigma(W_o \cdot [w_n, h_{n-1}] + b_o) \quad (3)$$

$$c_n = f_n \odot c_{n-1} + i_n \odot \tan h(W_c \cdot [w_n, h_{n-1}] + b_c) \quad (4)$$

$$h_n = o_n \odot \tan h(c_n) \quad (5)$$

where i, f , and o represent the input, forget, and output gates, correspondingly. During this technique, σ implies the sigmoid functions executed to manage every iteration. Afterward, $\{W_i, W_f, W_o, W_c, b_i, b_f, b_o, b_c\}$ signifies the attributes that exist learned in training.

For LSTM neural networks, Adam [38] optimizes the target function $f(\theta)$ (the mean squared error was utilized in the proposed model), intending to find parameters that minimize the mean squared error. Instead of using a stationary target, Adam uses sparse gradients and accomplishes a form of step size annealing without any additional effort.

The Adam manner implements dynamic alteration of several parameters with computing the gradient 1st-order moment estimation m_t and 2nd-order moment estimation v_t , as illustrated in Equations (6)–(8), where β_1 and β_2 correspondingly represent the 1st-order exponential damping decrement and 2nd-order exponential damping decrement. g_t stands for the gradient of parameters at time step t from the loss function $J_{\text{sparse}}(W, b)$.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \cdot g_t \quad (6)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (7)$$

$$g_t \leftarrow \nabla_{\theta} J_t(\theta_{t-1}) \quad (8)$$

Computer bias-corrected to m_t and v_t :

$$m'_t = \frac{m_t}{1 - \beta_1^t} \quad (9)$$

$$v'_t = \frac{v_t}{1 - \beta_2^t} \quad (10)$$

Update parameters:

$$\theta_{t+1} = \theta_t - \frac{\gamma}{\sqrt{v'_t} + \zeta} \cdot m'_t \quad (11)$$

γ indicates the updating step size, ζ attains the small constant to prevent the denominator that exists 0. In addition, the learning rate of the LSTM model is adjusted using the OGSO algorithm.

GSO is assumed as an intelligence swarm optimized technique utilized for speeding up the luminescent feature of fireflies. This GSO technique has shared the glowworm swarm from the space solution and FF of all glowworm places [19]. The strong glowworm was higher brightness, and an optimal place was developed for maximum FF rate. The glowworm was conceived of the dynamic line of sight, for instance, the decision domain, the range of density for neighboring nodes. Conversely, the decision radius has been constrained if the glowworm travels near a similar type of robust fluorescence from the

decision domain. Obtaining the maximum value of iterations, every glowworm is placed from a better place. The process contained from the GSO techniques is demonstrated in Figure 4 [43].

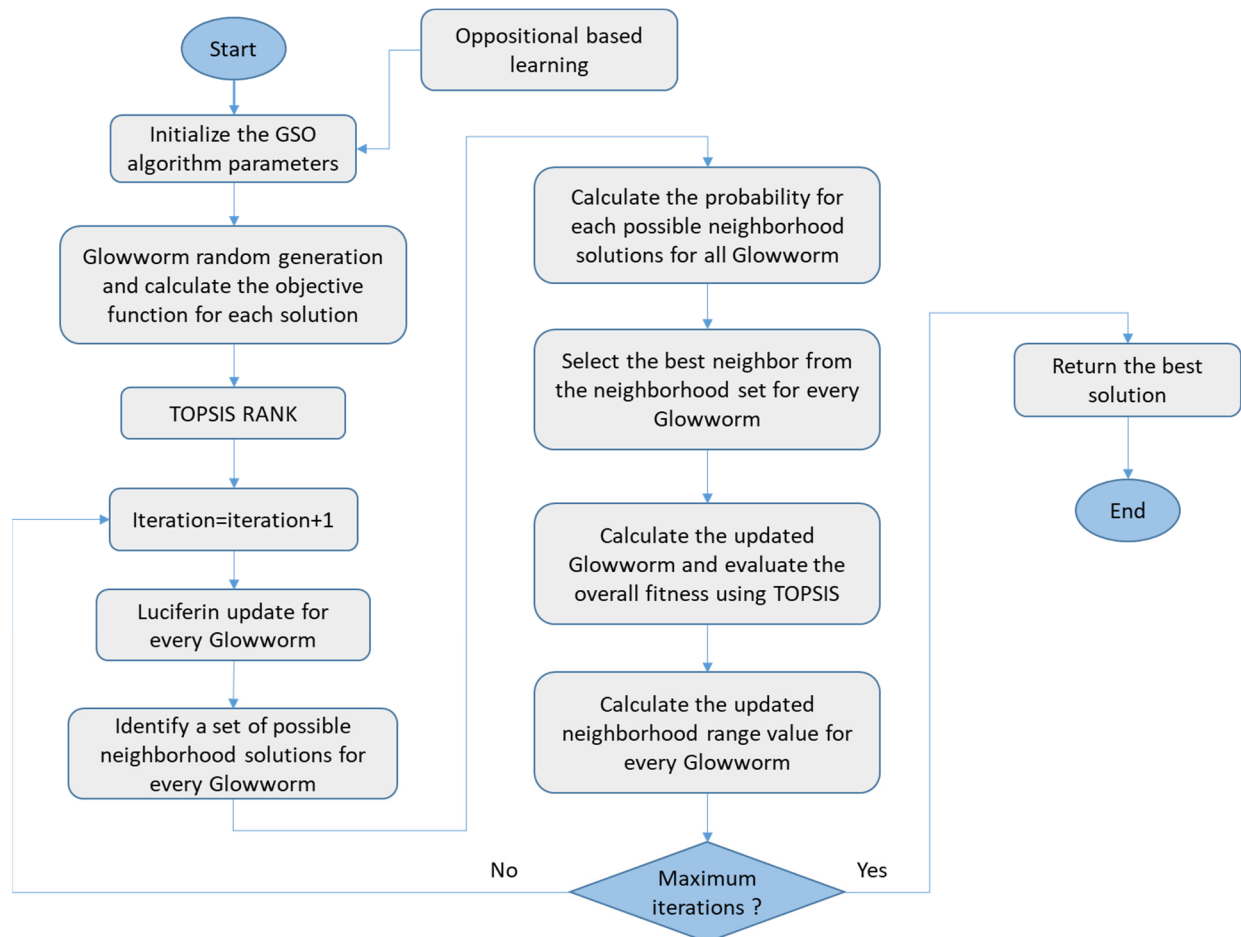


Figure 4. Diagram showing the steps in the OGSO algorithm.

It involves five stages as given as follows:

- Fluorescence in concentration
- Neighboring set
- Decision domain radius
- Moving possibility
- Glowworm location

The fluorescence from the concentration updating techniques are written in the subsequent:

$$l_i(t) = (1 - \alpha)l_i(r - 1) + \beta f(x_i(t)), \quad (12)$$

But $l_i(f)$ signifies the fluorescence from the concentration of i th glowworms in time f , α refers to the fluorescence in volatilization coefficients, β represents fluorescence from development factor, $f(x)$ refers to the FF and $x_i(r)$ demonstrated the place of glowworms i in f time is provided as:

$$N_i(t) = \left\{ j : \|x_j(r) - x_i(t)\| < r_d^i; l_i(t) < l_j(t) \right\}, \quad (13)$$

whereas $N_i(f)$ refers the neighbor group of i th glowworms in time r and $r_d^i(r)$ stands for the radius of decision domains of i th glowworms in time f as follow:

$$r_d^i(t+1) = \min \left\{ r_s, \max \left\{ r_d^i(t) + \gamma(n_i - |N_i(t)|) \right\} \right\}, \quad (14)$$

Here r_s refers to the attained radius of a glowworm, γ denotes the values of decision domains, and n_i shows the neighboring threshold. The moving probabilities of an upgrading technique were provided under:

$$p_{ij}(r) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_t} l_k(t) - l_i(t)}, \quad (15)$$

where $p_{ij}(t)$ refers to the possibilities that glowworm i travels to glowworms j in r time as follows:

$$x_i(t+1) = x_i(t) + s \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right), \quad (16)$$

The OBL is the main aim in the effectual optimization technique to enhance the convergences speed of distinct heuristic improving techniques. The productive execution of OBL contributes to estimating the opposite and existing population from the similar generation to recognize optimal candidate solutions of provided issues. The OBL techniques are effectually utilized in distinct Meta-heuristics used to enhance convergences speed. The methods of the opposite amount are explained in OBL.

Let $N \in N[x, y]$ represent the real numbers. An opposite number N^0 is provided as:

$$N^0 = x + y - N \quad (17)$$

In the d -dimensional searching region, the representation can be extended as follows:

$$N_i^0 = x_i + y_i - N_i \quad (18)$$

Here (N_1, N_2, \dots, N_d) defines the d -dimensional exploring area and $N_i[x_i, y_i]$, $i = 1, 2, \dots, d$. In the OBO, this technique of OBL has been utilized in this beginning process of the GSO technique and in every iteration from the application of jump rate.

5. Experimental Validation

Ethereum (ETH) return rates are used as targets, and simulations are run to ensure a more accurate forecast for time series data. This model has been shown to be more precise. The Python tool simulates the proposed model. Data were collected from January 2018 to December 2018 over 365 days to measure the return on bitcoin. It is possible to verify the bitcoin return rate using model checking and experimental simulations. For this reason, the dataset is split up into two parts: training data and testing data, with the training data comprising 80% of the total.

The proposed model RRP-DLBEP is evaluated against a set of current and traditional methods to validate its performance, such as GANMLP [16], PSOLSSVR [15], SVM [12–17], BPNN [18], GA-SVM [13], ANN [18], and Random Walk. One of these methods is GANMLP [16], which predicts the new return rate of Blockchain financial products. Using the best most miniature square support vector machine for blockchain financial product revenue prediction, the SVM [18] is another approach. It is also tested against the conventional BPNN and PSOLSSVR models.

Detailed predictive performance of the RRP-DLBFP technique takes place in this section. Table 1 investigates the analysis of the results of the RRP-DLBFP technique in terms of MSE and MAPE under the training and testing sets.

Table 1. Comparison study of RRP-DLBFP under different aspects.

Models	Training Dataset		Testing Dataset	
	MSE	MAPE	MSE	MAPE
RRP-DLBFP	0.0435	2.9845	0.0655	3.9856
GANMLP	0.0698	3.1902	0.0962	4.2890
PSOLSSVR	0.0701	3.2126	0.0973	4.3531
SVM	0.1091	4.7237	0.1132	4.5721
BPNN	0.0712	3.2356	0.1021	4.7372
GA-SVM	0.0945	4.4697	0.1032	4.6938
ANN	0.0978	4.5860	0.1076	4.7139
Random Walk	0.1014	4.3146	0.1034	4.3154

Figure 5 showcases the MSE investigation of the RRP-DLBFP technique under the applied training set. The figure illustrated that the SVM, GA-SVM, ANN, and Random walk techniques have resulted in poor outcomes with the maximum MSE of 0.1091, 0.0945, 0.0978, and 0.104. Along with that, the PSOLSSVR and BPNN techniques have obtained a somewhat better performance with the MSE of 0.0701 and 0.0712, respectively. In line with this, the GANMLP technique has accomplished a moderately reasonable MSE of 0.0698. However, the RRP-DLBFP technique has attained an effective outcome with a lower MSE of 0.0435.

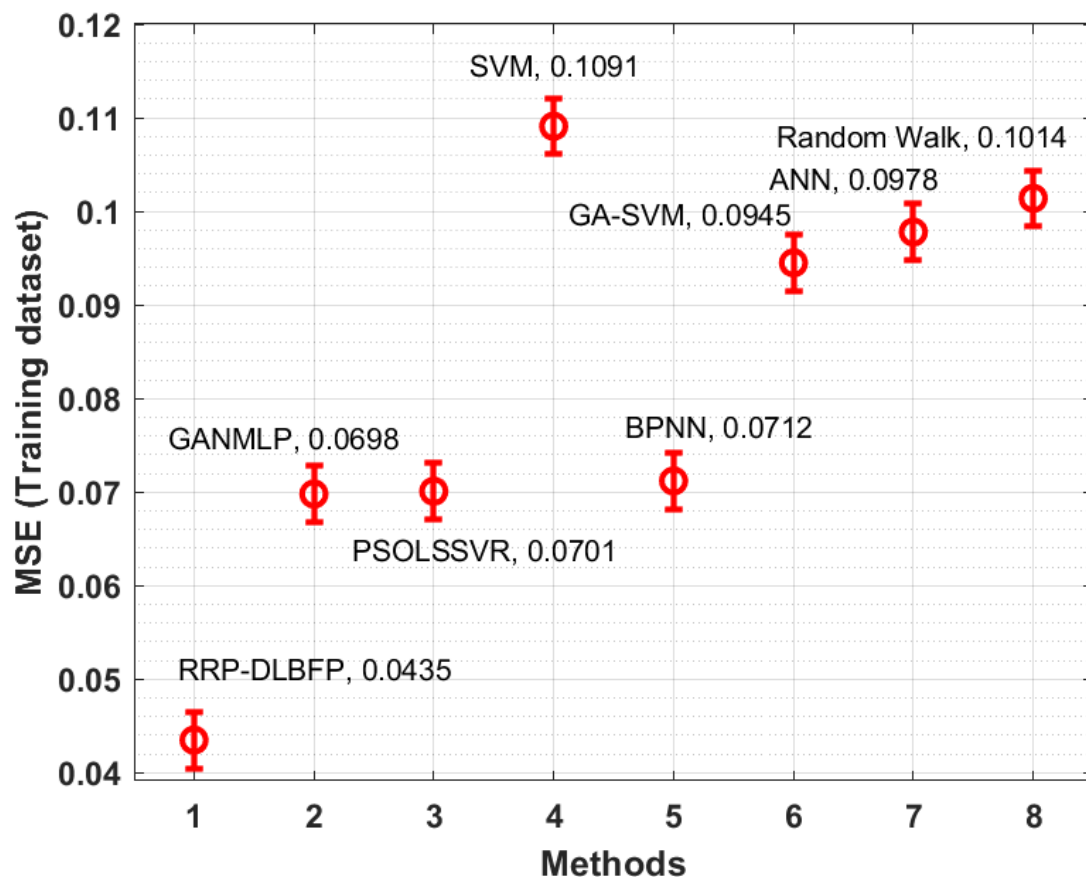
**Figure 5.** MSE analysis of RRP-DLBFP technique on the training set.

Figure 6 illustrates the MAPE analysis of the RRP-DLBFP manner under the applied training set. The figure demonstrated that the SVM, GA-SVM, ANN, and Random walk

approaches have resulted in an unsuccessful outcome with a maximum MAPE of 4.7237, 4.4697, 4.5860, and 4.3146. Likewise, the PSOLSSVR and BPNN techniques have gained somewhat optimum performance with the MAPE of 3.2356 and 3.2126, correspondingly. Similarly, the GANMLP technique has accomplished a moderately reasonable MAPE of 3.1902. Lastly, the RRP-DLBFP methodology has attained an effective outcome with a minimum MAPE of 2.9845.

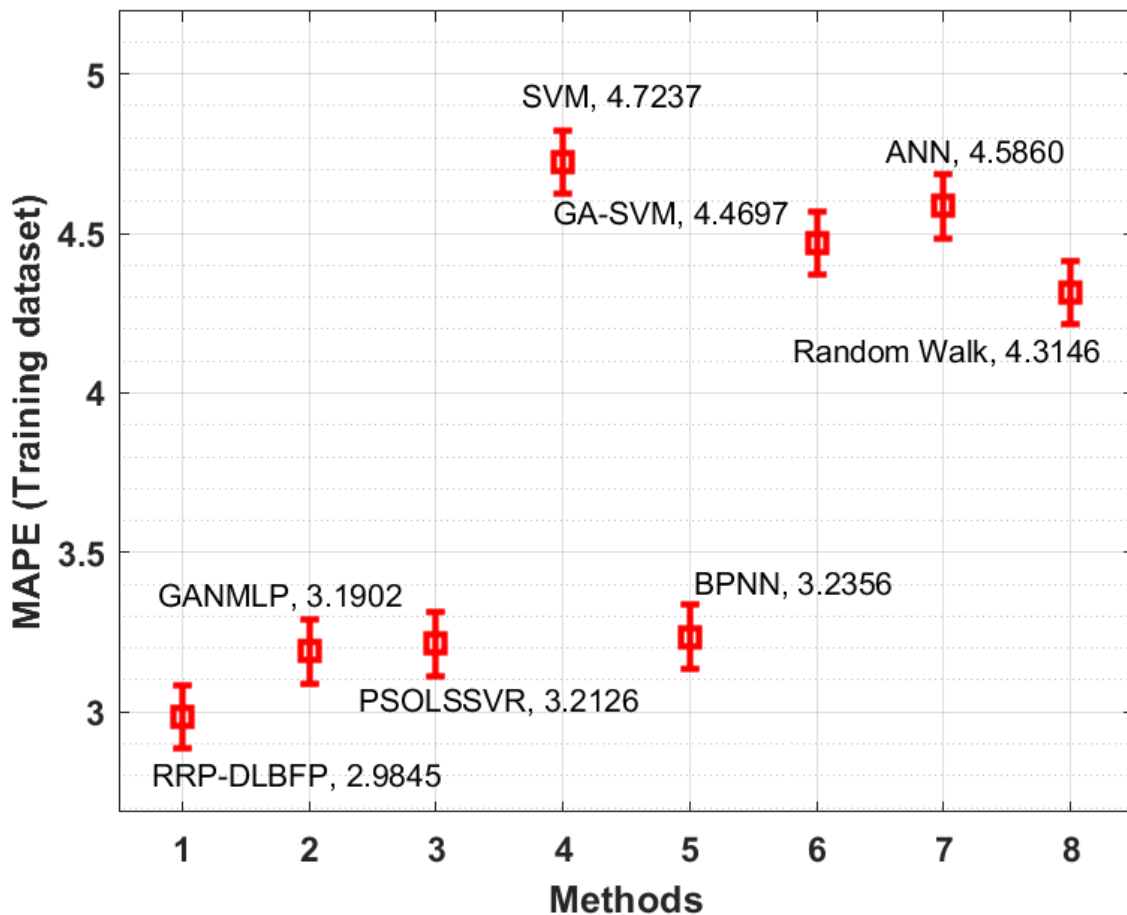


Figure 6. MAE analysis of RRP-DLBFP technique on the training set.

Figure 7 depicts the MSE analysis of the RRP-DLBFP approach under the applied testing set. The figure exhibited that the SVM, GA-SVM, ANN, and Random walk techniques have resulted in a worse outcome with the superior MSE of 0.1132, 0.1032, 0.1076, and 0.1034, respectively. The BPNN and PSOLSSVR techniques have achieved somewhat better performance with the MSE of 0.1021 and 0.0973, correspondingly. In addition, the GANMLP algorithm has accomplished a moderately reasonable MSE of 0.0962; but, the RRP-DLBFP methodology reached an effective outcome with a lower MSE of 0.0655.

Figure 8 demonstrates the MAPE analysis of the RRP-DLBFP manner under the applied testing set. The results outperformed that the SVM, BPNN, GA-SVM, and ANN approaches resulting in ineffective outcomes with a higher MAPE of 4.5721, 4.7372, 4.6938, and 4.7139. Along with that, the Random walk and PSOLSSVR techniques have obtained slightly optimum performance with the MAPE of 4.3154 and 4.3531, correspondingly. Next, the GANMLP methodology accomplished a moderately reasonable MAPE of 4.2890. Finally, the RRP-DLBFP approach has obtained effective results with a lesser MAPE of 3.9856. Findings for all models in training and testing are shown in Figures 9 and 10, which combine all MSE and MAPE results into one graph.

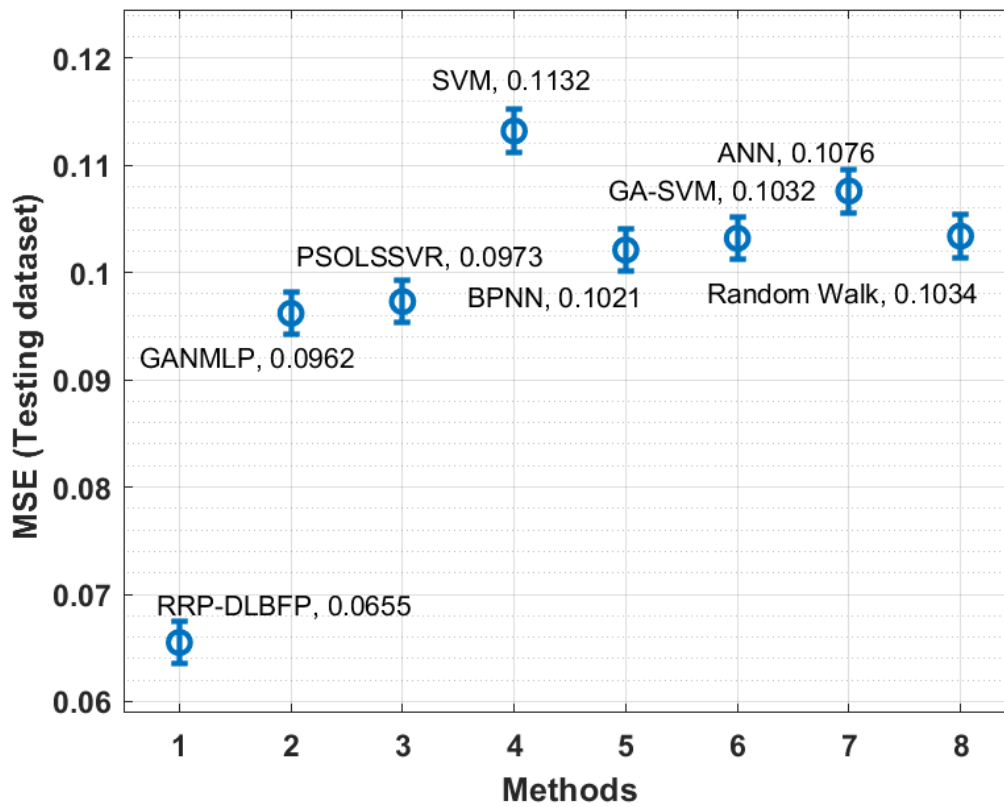


Figure 7. MSE analysis of RRP-DLBFP technique on the testing set.

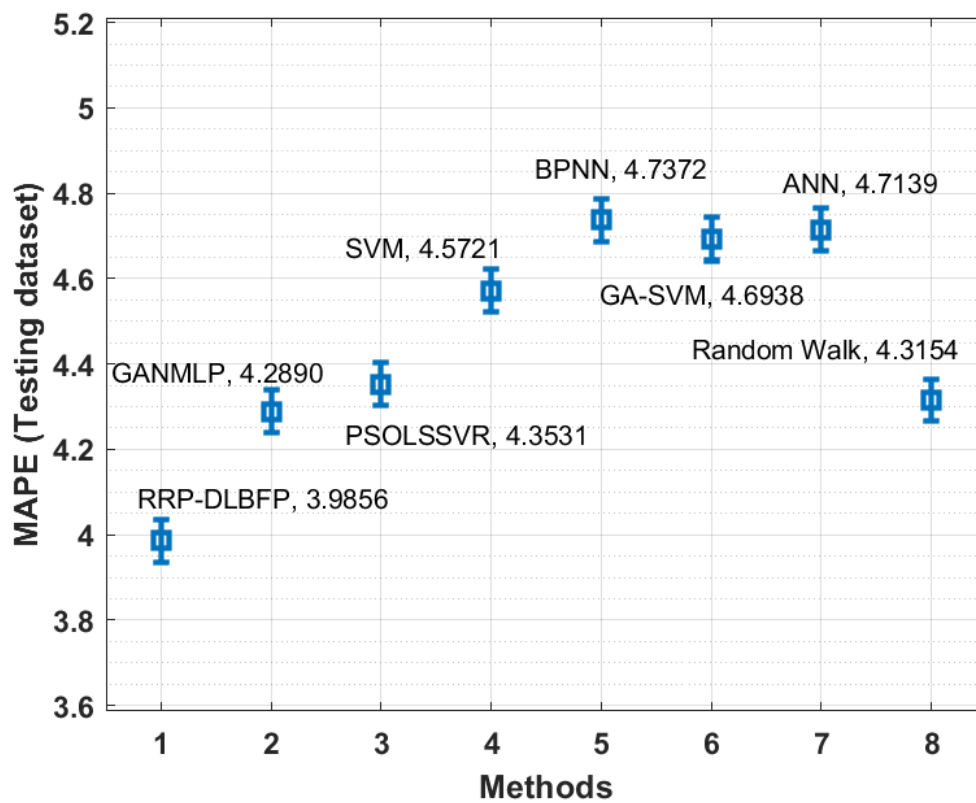


Figure 8. MAPE analysis of RRP-DLBFP technique on the testing set.

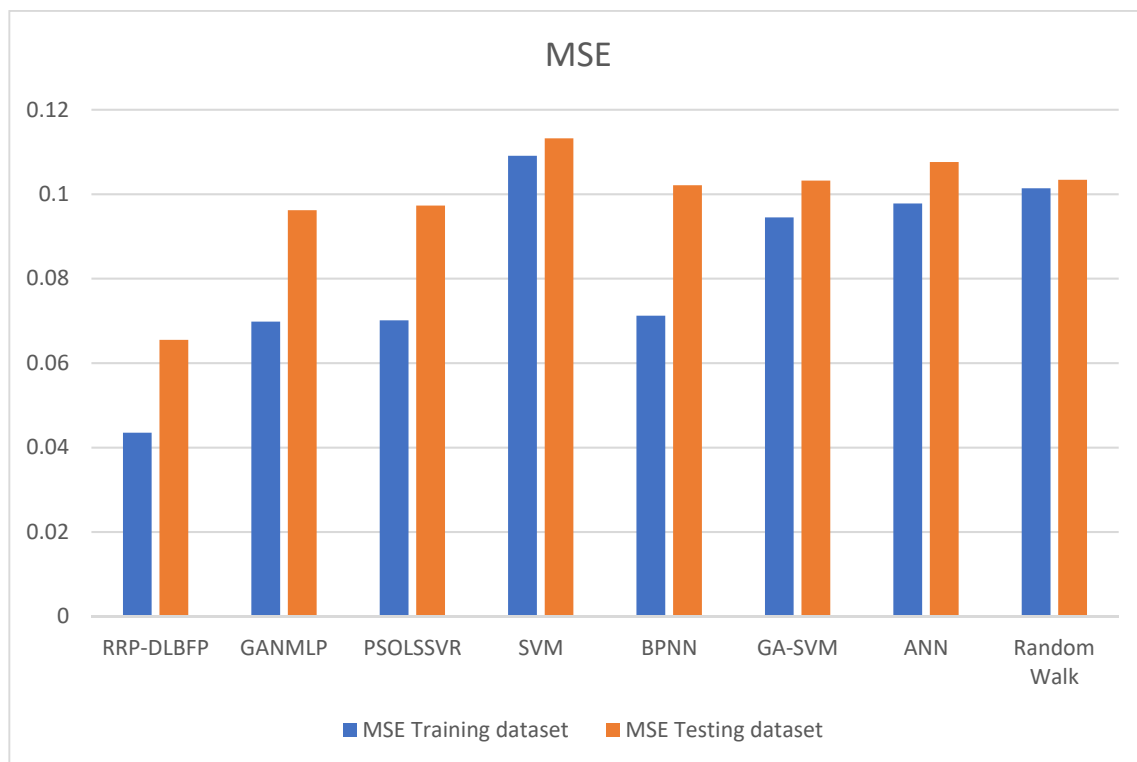


Figure 9. MSE scores for all models are shown below.

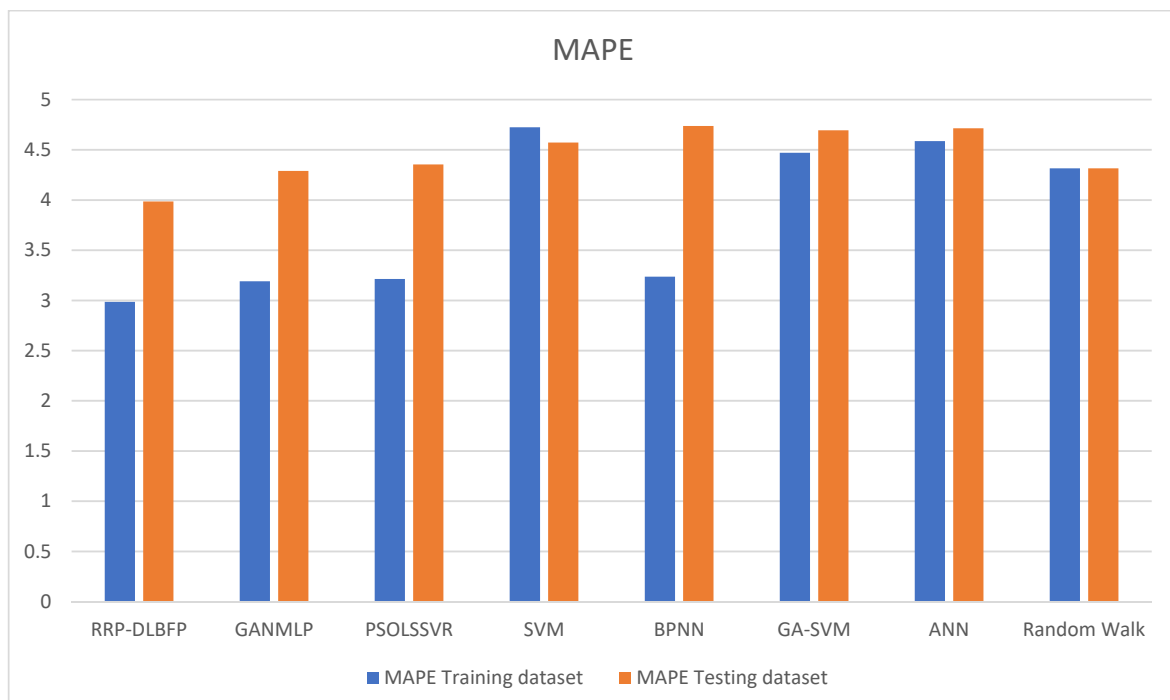


Figure 10. All of the MAPE models' results.

6. Conclusions

This study has developed an RRP-DLBFP technique to predict the return rate of blockchain financial products. The proposed RRP-DLBFP technique involves the design of the LSTM model for the predictive analysis of return rate. In addition, Adam optimizer and the OGSO algorithm are applied to adjust the hyperparameters of the LSTM model

optimally, consequently increasing the predictive performance. The ETH return rate is preferred to ensure the supreme performance of the RRP-DLBFP technique, and the simulation results are investigated in terms of different measures. The simulation outcomes highlighted the supremacy of the RRP-DLBFP technique over the current state-of-the-art techniques in terms of diverse evaluation parameters. In the future, metaheuristic-based hyperparameter tuning models can be devised to boost the predictive outcome further.

Author Contributions: Data curation, M.I.A.; Formal analysis, N.M.; Project administration, I.M.E.-H.; Supervision, I.M.E.-H. and M.E.; Writing—original draft, N.M. and M.I.A.; Writing—review & editing, I.M.E.-H. and M.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

- Saadah, S.; Whafa, A.A. Monitoring Financial Stability Based on Prediction of Cryptocurrencies Price Using Intelligent Algorithm. In Proceedings of the 2020 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 5–6 August 2020; pp. 1–10.
- Subramanian, H. Decentralized blockchain-based electronic marketplaces. *Commun. ACM* **2017**, *61*, 78–84. [[CrossRef](#)]
- Yilmaz, N.K.; Hazar, H.B. Predicting future cryptocurrency investment trends by conjoint analysis. *J. Econ. Financ. Account.* **2018**, *5*, 321–330. [[CrossRef](#)]
- Lee, R.S. Time series chaotic neural oscillatory networks for financial prediction. In *Quantum Finance*; Springer: Singapore, 2020; pp. 301–337.
- Lee, R.S. Chaotic interval type-2 fuzzy neuro-oscillatory network (CIT2-FNON) for Worldwide 129 financial products prediction. *Int. J. Fuzzy Syst.* **2019**, *21*, 2223–2244. [[CrossRef](#)]
- Sigova, M.V.; Klioutchnikov, I.K.; Zatevakhina, A.V.; Klioutchnikov, O.I. Approaches to evaluating the function of prediction of decentralized applications. In Proceedings of the 2018 International Conference on Artificial Intelligence Applications and Innovations (IC-AIAI), Nicosia, Cyprus, 31 October–2 November 2018; pp. 1–6.
- Schlegel, M.; Zavolokina, L.; Schwabe, G. Blockchain technologies from the consumers' perspective: What is there and why should who care? In Proceedings of the 51st Hawaii International Conference on System Sciences, Hilton Waikoloa Village, HI, USA, 3–6 January 2018.
- Saracevic, M.; Wang, N.; Zukorlic, E.E.; Becirovic, S. New Model of Sustainable Supply Chain Finance Based on Blockchain Technology. *Am. J. Bus. Oper. Res.* **2021**, *3*, 61–76. [[CrossRef](#)]
- Salah, K.; Rehman, M.H.; Nizamuddin, N.; Al-Fuqaha, A. Blockchain for AI: Review and open research challenges. *IEEE Access* **2017**, *7*, 10127–10149. [[CrossRef](#)]
- Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [[CrossRef](#)]
- Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
- Kiktenko, E.O.; Pozhar, N.O.; Anufriev, M.N.; Trushechkin, A.S.; Yunusov, R.R.; Kurochkin, Y.V.; Lvovsky, A.I.; Fedorov, A.K. Quantum-secured blockchain. *Quantum Sci. Technol.* **2018**, *3*, 035004. [[CrossRef](#)]
- Hussein, A.F.; Arunkumar, N.; Ramirez-Gonzalez, G.; Abdulhay, E.; Tavares, J.M.; de Albuquerque, V.H. A medical records managing and securing Blockchain based system supported by a genetic algorithm and discrete wavelet transform. *Cogn. Syst. Res.* **2018**, *52*, 1–11. [[CrossRef](#)]
- Huang, W.; Nakamori, Y.; Wang, S.Y. Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.* **2005**, *32*, 2513–2522. [[CrossRef](#)]
- Indera, N.I.; Yassin, I.M.; Zabidi, A.; Rizman, Z.I. Non-linear autoregressive with exogeneous input (NARX) Bitcoin price prediction model using PSO-optimized parameters and moving average technical indicators. *J. Fundam. Appl. Sci.* **2017**, *9*, 791–808. [[CrossRef](#)]
- Kaur, S.; Singh, K.D.; Singh, P.; Kaur, R. Ensemble Model to Predict Credit Card Fraud Detection Using Random Forest and Generative Adversarial Networks. In *Emerging Technologies in Data Mining and Information Security*; Springer: Singapore, 2021; pp. 87–97.
- Poongodi, M.; Sharma, A.; Vijayakumar, V.; Bhardwaj, V.; Sharma, A.P.; Iqbal, R.; Kumar, R. Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system. *Comput. Electr. Eng.* **2020**, *81*, 106527.

18. Sivaram, M.; Lydia, E.L.; Pustokhina, I.V.; Pustokhin, D.A.; Elhoseny, M.; Joshi, G.P.; Shankar, K. An optimal least square support vector machine based earnings prediction of blockchain financial products. *IEEE Access* **2020**, *8*, 120321–120330. [CrossRef]
19. Celeste, V.; Corbet, S.; Gurdgiev, C. Fractal dynamics and wavelet analysis: Deep volatility and return properties of Bitcoin, Ethereum and Ripple. *Q. Rev. Econ. Financ.* **2020**, *76*, 310–324. [CrossRef]
20. Sifat, I.M.; Mohamad, A.; Shariff, M.S. Lead-lag relationship between bitcoin and ethereum: Evidence from hourly and daily data. *Res. Int. Bus. Financ.* **2019**, *50*, 306–321. [CrossRef]
21. Shah, D.; Zhang, K. Bayesian regression and Bitcoin. In Proceedings of the 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 30 September–3 October 2014; pp. 409–414.
22. Matta, M.; Lunesu, I.; Marchesi, M. Bitcoin Spread Prediction Using Social and Web Search Media. In Proceedings of the UMAP Workshops, 23rd Conference on User Modeling, Adaptation and Personalization, Dublin, Ireland, 29 June–3 July 2015; pp. 1–10.
23. Matta, M.; Lunesu, I.; Marchesi, M. The predictor impact of Web search media on Bitcoin trading volumes. In Proceedings of the 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), Lisbon, Portugal, 12–14 November 2015; Volume 1, pp. 620–626.
24. Gu, B.; Konana, P.; Liu, A.; Rajagopalan, B.; Ghosh, J. Identifying Information in Stock Message Boards and Its Implications for Stock Market Efficiency. 2006. Available online: <http://www.ideal.ece.utexas.edu/pdfs/151.pdf> (accessed on 5 September 2021).
25. Greaves, A.; Au, B. Using the bitcoin transaction graph to predict the price of bitcoin. *Comput. Sci.* **2015**, 1–8.
26. Madan, I.; Saluja, S.; Zhao, A. Automated Bitcoin Trading via Machine Learning Algorithms. Volume 20. 2015. Available online: <http://cs229.stanford.edu/proj2014/Isaac%20Madan,%20Shaurya%20Saluja,%20Aojia%20Zhao,Automated%20Bitcoin%20Trading%20via%20Machine%20Learning%20Algorithms.pdf> (accessed on 5 September 2021).
27. Delfin-Vidal, R.; Romero-Meléndez, G. The fractal nature of bitcoin: Evidence from wavelet power spectra. In *Trends in Mathematical Economics*; Pinto, A., Accinelli Gamba, E., Yannacopoulos, A., Hervés-Beloso, C., Eds.; Springer: Cham, Switzerland, 2016; pp. 73–98.
28. Kristoufek, L. What are the main drivers of the Bitcoin price? Evidence from wavelet coherence analysis. *PLoS ONE* **2015**, *10*, e0123923. [CrossRef] [PubMed]
29. White, H. Economic prediction using neural networks: The case of IBM daily stock returns. *ICNN* **1988**, *2*, 451–458.
30. Koskela, T.; Lehtokangas, M.; Saarinen, J.; Kaski, K. Time series prediction with multilayer perceptron, FIR and Elman neural networks. In Proceedings of the World Congress on Neural Networks, Bochum, Germany, 16–19 July 1996; pp. 491–496.
31. Giles, C.L.; Lawrence, S.; Tsoi, A.C. Noisy time series prediction using recurrent neural networks and grammatical inference. *Mach. Learn.* **2001**, *44*, 161–183. [CrossRef]
32. Rather, A.M.; Agarwal, A.; Sastry, V.N. Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Syst. Appl.* **2015**, *42*, 3234–3241. [CrossRef]
33. McNally, S.; Roche, J.; Caton, S. Predicting the price of bitcoin using machine learning. In Proceedings of the 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Cambridge, UK, 21–23 March 2018; pp. 339–343.
34. Catanzaro, B.; Sundaram, N.; Keutzer, K. Fast support vector machine training and classification on graphics processors. In Proceedings of the Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 104–111.
35. Cireşan, D.C.; Meier, U.; Gambardella, L.M.; Schmidhuber, J. Deep, big, simple neural nets for handwritten digit recognition. *Neural Comput.* **2010**, *22*, 3207–3220. [CrossRef] [PubMed]
36. Mehdizadeh, M. Integrating ABC analysis and rough set theory to control the inventories of distributor in the supply chain of auto spare parts. *Comput. Ind. Eng.* **2020**, *139*, 105673. [CrossRef]
37. Liu, Y.; Zhang, Q.; Fan, Z.P.; You, T.H.; Wang, L.X. Maintenance spare parts demand forecasting for automobile 4S shop considering weather data. *IEEE Trans. Fuzzy Syst.* **2018**, *27*, 943–955. [CrossRef]
38. Chang, Z.; Zhang, Y.; Chen, W. Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. *Energy* **2019**, *187*, 115804. [CrossRef]
39. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
40. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. *Futur. Gener. Comput. Syst.* **2020**, *107*, 841–853. [CrossRef]
41. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
42. Bello, I.; Zoph, B.; Vasudevan, V.; Le, Q.V. Neural optimizer search with reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, Australia, 6–11 August 2017; pp. 459–468.
43. Jayakumar, D.N.; Venkatesh, P. Glowworm swarm optimization algorithm with topsis for solving multiple objective environmental economic dispatch problem. *Appl. Soft Comput.* **2014**, *23*, 375–386. [CrossRef]