

Research Article

Research on the Forecast of the Combined GRU Model Based on the Optimized Bat Algorithm in the Bank Transaction Volume

Jingfeng Rong ^{1,2}, Di Wang,¹ Boya Zhang,¹ and Yonggui Wang¹

¹School of Software, Liaoning Technical University, Huludao, Liaoning, China

²Fuxin Bank, Fuxin, Liaoning, China

Correspondence should be addressed to Jingfeng Rong; rongjingfeng@xs.ustb.edu.cn

Received 8 March 2022; Revised 25 May 2022; Accepted 1 June 2022; Published 24 June 2022

Academic Editor: Ateeq Rehman

Copyright © 2022 Jingfeng Rong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The accurate prediction of the transaction volume of the core accounting system is of great significance to the stable operation of commercial banks. After fully investigating the transaction volume history of the core accounting system and discovering the unique time-series attributes of the data, this study proposes a transaction volume prediction model of the core accounting system of commercial banks based on the improved bat algorithm and the optimized gating loop unit neural network. The chaos algorithm, reverse learning, evolution, and search mechanisms are implemented to improve the search efficiency of solving the global optimization problem, to overcome the shortcomings of the original bat algorithm, such as early maturation and easily falling into the trap of a local optimal solution, and to enhance the algorithm's optimization ability and precision. Moreover, the bat algorithm's optimization ability is fully utilized, and the optimal parameters of the GRU model, such as network layers and neural units, are determined. Finally, the effectiveness of the combination model is evaluated using historical transaction data from the core accounting system stored in a bank data warehouse. The experimental results show that the improved combined GRU model performs well in mean squared error, root mean squared error, and mean absolute error, which is superior to the original GRU model and the traditional time-series forecasting ARMA model. The proposed combined model can be effectively applied to the transaction volume prediction of the core accounting system of commercial banks.

1. Introduction

With the continuous spread of the worldwide epidemic, more and more transactions are moving from offline to online, and the demand for online financial services of commercial banks in various industries has shown a surging trend. According to the statistical data analysis of the China Banking Service Report in 2021 released by the China Banking Association, the banking industry achieves multi-dimensional digital intelligent service transformation through product service iterative process around the development concept of technology-enabled business innovation, assisting enterprises in resuming production as soon as possible, supporting entities' economic development, and preventing financial risks [1].

At present, commercial banks offer various service channels to customers through the combination of online

and offline operation modes. Customers can do business online via Internet banking, mobile banking, and bank APPs, or they can accomplish associated tasks manually at a counter. The development of various service channels will test the transaction capacity of the core accounting systems of commercial banks [2]. To avoid hardware failure due to a rise in transaction volume, the service load is often enhanced through hardware redundancy. This strategy, on the other hand, will make equipment operation and maintenance more complicated and, as a result, will surely prolong the time it takes to maintain the equipment. At the same time, it will make managing the software service interface more complicated, and it may even result in a waste of service resources and computer computing power. Therefore, the research on the accurate prediction of system transaction volume is of great significance in reducing the redundancy of commercial bank systems [3].

Machine learning, as a hot topic in current forecasting research, has the advantages of mining the internal implicit relationship of historical data, analyzing the data hierarchy, and automatically feeding back data features through learning [4]. Because of the advantages of machine learning, more and more machine learning is applied in different fields [5]. Wang [6] used the gated recurrent neural network (GRU) model to analyze the big meteorological data, excavate the correlation between meteorological elements, and estimate each meteorological element. Tong [7] adopted the GRU model based on social, economic, and epidemiological characteristic data to study the current COVID-19 epidemic situation and predict its changing trend. Aiming at the problem of network intrusion security, Liang [8] proposed to dig deep into the network data information by summarizing the characteristics of network activities and used the GRU model to classify and identify the information, providing a brand-new idea for the processing of network intrusion search. In particular, the GRU model has a high prediction accuracy for time-series attribute data. However, its complicated internal structure leads to more complicated super parameters, which has a large influence on the prediction accuracy [9, 10].

Presently, the bat algorithm (BA) has been widely used in wind energy development, model prediction [11], UAV path planning [12], and economic development [13]. Some scholars have made corresponding improvements to BA. Cai et al. [14] introduced Gaussian disturbance into BA and adopted different speed updating strategies to improve the global exploration ability. The authors in [15] combined the tone fine-tuning operator of harmony search to improve the convergence accuracy and speed of BA. Huang et al. [16] proposed an improved binary BA by using neighborhood search and dynamic inertia weight strategy. The authors in [17, 18] used the ergodicity of chaos to study the optimization efficiency of different chaotic mapping for BA. In [19], the chaos of elite individuals is optimized by logistic mapping and dynamically shrinks the search space to avoid BA falling into local optimum prematurely. Many scholars further simulate bat behavior from the biological point of view, combined with habitat selection and adaptive compensation of the Doppler effect, and apply it to practical engineering problems [20–23]. He et al. [24] integrated the simulated annealing algorithm and Gaussian mutation into BA and made use of the advantages of the simulated annealing algorithm, which not only enhances the exploration ability of BA but also accelerates the convergence speed. Although BA has the advantages of rapid convergence, few parameters, and strong robustness, BA also has the disadvantages of easily falling into local optimum and low solution accuracy, like other heuristic optimization algorithms.

Therefore, this study proposes a combined BU-GRU neural network model to predict the transaction volume of the core accounting system of commercial banks. The major contributions of the proposed study are as follows:

- (i) Fuch mapping is introduced into the bat algorithm to produce uniformly distributed individuals.

Improving the initial population quality can improve the search efficiency of solving the global optimization problem.

- (ii) Evolutionary mechanism and search mechanism are introduced to enhance the ability to get rid of local extremum and give full play to the information leading ability of the optimal individual.
- (iii) A GRU combination model based on the bat algorithm is proposed, which can give full play to the optimization ability of the bat algorithm and optimize the super parameters of the GRU neural network.
- (iv) Aiming at the unique time-series attributes of transaction volume data in the core accounting system of commercial banks, it is proposed to optimize GRU neural network model based on an improved bat algorithm, which can overcome the problems of low precision and large error of traditional time-series data prediction.

The rest of the paper is organized as follows: Section 2 provides an overview of the GRU neural network. Section 3 provides a detailed description of the bat algorithm and different optimization techniques. Section 4 describes the proposed ESCBA-GRU model. In Section 5, experimental analysis is presented, and Section 6 is about the conclusion.

2. GRU Network

GRU neural network is a variant neural network model proposed to solve the gradient explosion problem of the recurrent neural network (RNN) model. The core of this algorithm is a combination of a sigmoid function and some relational operations, which together constitute the GRU control and information selection mechanism. By using the sigmoid function, the inherited memory state and the current input data are calculated and mapped between 0 and 1 to simulate the memorizing and discarding state of data. Similar to the mechanism of three control gate units in the long short term memory (LSTM) model, there are two control units in GRU, reset gate and update gate, because the GRU model evolved from the LSTM model [25]. Therefore, the two-gate structure is simplified from the three-gate structure. The input-output structure of the GRU model is shown in Figure 1.

h indicates the information transmitted between each nerve unit, x specifies the input information, and y is the output information of the current nerve unit. However, at every moment or in the final full connection calculation, GRU is constantly updating itself. The update is based on its inheritance and combined with the current alternative answers. Inside the neural unit, this genetic information will be processed and calculated. In this way, such neural units are connected in series to form a network-like model with a complex structure and direct or indirect connection between neural units [26]. The internal structure of the model is shown in Figure 2.

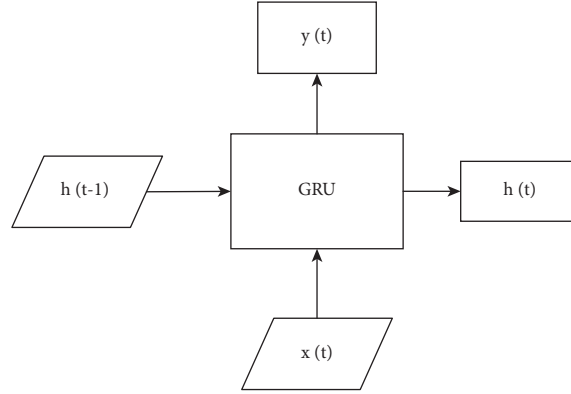


FIGURE 1: The input-output structure of GRU.

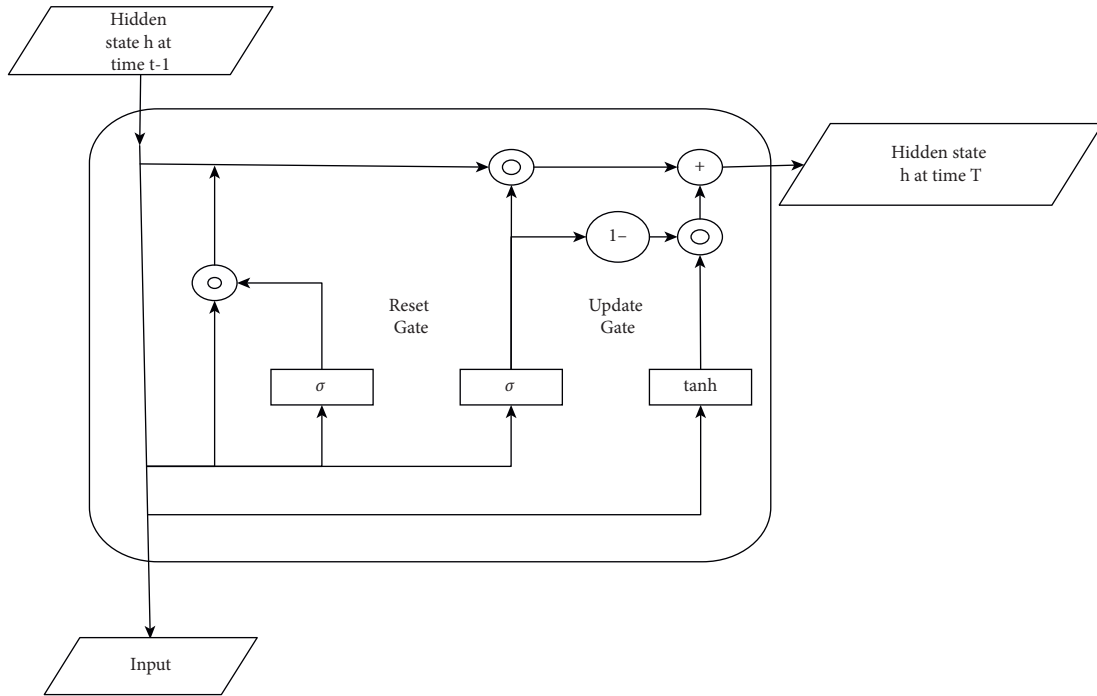


FIGURE 2: GRU internal control gate structure.

2.1. *Update Gate.* Z_t indicates the update gate of GRU at the current time t , which is responsible for determining how much information content is transmitted to the next neural unit. If Z_t it is infinitely close to 1, h_{t-1} will be equal to h_t , which means that the received information will be directly transmitted to the next unit without being processed or forgotten. On the contrary, if Z_t it is infinitely close to 0, h_{t-1} is not equal to h_t , which means that the received information will be forgotten through machining calculation and will not be passed on to the next unit [27]. The process of updating the gate is expressed as follows:

$$z_t = \delta(W_z \bullet [h_{t-1}, x_t]), \quad (1)$$

where δ is used to indicate sigmoid function, W_z is the update gate weight corresponding to h_{t-1} when solving Z_t , and x_t indicates the currently entered information.

2.2. *Reset Gate.* r_t indicates the reset gate at the current time t , and r_t determines the influence of h_{t-1} on h_t . When r_t is approximately equal to 0, h_{t-1} will not pass on information to \tilde{h}_{t-1} . When r_t is approximately equal to 1, h_{t-1} will pass on information to \tilde{h}_{t-1} . The reset gate operation is expressed as

$$r_t = \delta(W_r \bullet [h_{t-1}, x_t]), \quad (2)$$

where W_r indicates the reset gate weight corresponding to h_{t-1} when solving r_t .

2.3. *New Memory.* The update gate Z_t determines the current time, and the information amount of the previous time is included in the condition of result judgment. \tilde{h}_{t-1} indicates the comprehensive calculation result at the current input information and the information transmission h_{t-1} of the hidden layer at the previous moment, which includes the

newly input information and the above information. The new memory process is computed as

$$\bar{h}_t = \tanh(W_{\bar{h}} \bullet [r_t * h_{t-1}, x_t]). \quad (3)$$

2.4. Hidden Layer. The hidden layer state h_{t-1} of GRU at the current moment is calculated based on h_{t-1} , \bar{h} and at the previous moment, and the weight relationship between them is controlled by Z_t . The equation of its information transmission and the calculation process is as follows:

$$h_t = (1 - Z_t) * h_{t-1} + z_t * \bar{h}_t. \quad (4)$$

2.5. Results. After determining h_t at the current moment, the output of this neural unit can be completed, and finally, y_t data can be obtained after calculation. The output process is as follows:

$$y_t = \delta(W_o \bullet h_t). \quad (5)$$

Because of this special result, each unit has new input information and genetic information at the previous moment, which makes the network have the characteristics of dealing with nonlinear feature sequences.

Studies have shown that the GRU's hidden layer number H , model iteration number N , learning rate lr , and learning rate calculation factor decay have a great influence on the prediction results, while the bat optimization algorithm has better performance than other swarm intelligence algorithms in the whole optimization process [28]. Therefore, the bat algorithm is proposed to optimize the relevant super parameters of GRU.

3. Bat Algorithm (BA)

The BA is a new heuristic optimization algorithm based on swarm intelligence. It is developed according to bats' characteristics of using echolocation to search for prey and avoid obstacles.

3.1. Basic Bat Algorithm. In BA, x_i^t and v_i^t , respectively, indicate the position and speed of the i -th bat in the t -th generation. f_i and A_i^t shows the frequency and loudness of the bat, respectively, and r_i^t is the pulse emission rate. The position of the bat indicates the feasible solution. Because bats do not know the specific location of prey at first in the algorithm, they are randomly initialized by the following equation:

$$x_{ij}^t = x_j^{\min} + \text{rand}(0, 1)(x_j^{\max} - x_j^{\min}). \quad (6)$$

After the initial setting of each parameter is completed, the algorithm enters the following three main iterative processes:

- (i) Random flight: the spatial position of bats is updated by

$$\begin{cases} f_i = f_{\min} + (f_{\max} - f_{\min})\beta \\ v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*)f_i \\ x_i^t = x_i^{t-1} + v_i^t \end{cases}, \quad (7)$$

where f_{\min} and f_{\max} are the minimum and maximum values of frequency, $\beta \in [0, 1]$ is a random number that obeys uniform distribution, and x^* indicates the global optimal solution currently searched.

- (ii) Local search: once the optimal solution is generated, a local search is performed around its neighborhood by a random walk of the following equation:

$$x^{\text{new}} = x^* + \varepsilon A^t, \quad (8)$$

where $\varepsilon_j \in [0, 1]$ is a random vector and A^t is the average loudness.

- (iii) Update of control parameters: when bats accept new prey, their loudness and pulse emission rate are updated by the following equation:

$$\begin{cases} A_i^{t+1} = \alpha A_i^t \\ r_i^{t+1} = r_i^0 (1 - \exp(-\gamma t)) \end{cases}, \quad (9)$$

where α and γ are both constants. As the iteration progresses, when the bat approaches the optimal solution, the loudness gradually decreases, and the pulse emission rate gradually increases.

From the characteristics of the BA, we can see that its main problems are summarized as follows.

The BA generates the initial population solution through random initialization as given in equation (6). If the random solution is close to the optimal solution, it will converge at a faster speed. Once the random solution is far from the optimal solution, it may converge slowly, which has a great influence on algorithm optimization [29].

The algorithm is performing global exploration (equation (7)). At this time, bats fly towards the best individual of the population. If the best individual is far from the global optimum, the population will easily fall into the local optimum. Equation (8) means local development. With the iteration, r_i increases, and the possibility of $\text{rand} > r_i$ decreases. So, the algorithm performs development in the early stage and performs exploration in the later stage. Too much development and too little exploration will lead to premature convergence, and too much exploration and too little development will make it difficult to converge to the optimal solution, which will reduce the overall search performance.

No matter the exploration or development, the search process is completely dependent on a random walk, and the method is single, which makes bat individuals lack flexible mutation ability in iteration. There is no effective mechanism to jump out of the local extremum, which cannot guarantee rapid convergence to the optimal solution.

Therefore, the improvement of the BA in this study is embodied in three aspects: generating a suitable initial population, having the ability of continuous evolution and getting rid of early maturing, and updating adjustment parameters.

3.2. Fuch Chaotic Mapping. Chaos is a characteristic of a nonlinear system, which is mathematically defined as a random number generated by a simple deterministic function. Integrating chaos into a metaheuristic optimization algorithm, the core is to use the randomness, ergodicity, and irregularity of chaotic variables to replace random variables that obey the standard probability distribution for optimization search, to search the whole solution space, generate evenly distributed individuals, and improve the initial population quality [30].

At present, the chaotic mapping widely used in literature is logistic mapping and tent mapping. Among them, logistic mapping is sensitive to initial values. Only when the control parameter is 4 can it completely enter the chaotic state, with a high probability of taking values at both ends of the interval and uneven traversal. Tent mapping has better ergodic uniformity. But it has a small period and fixed point, which makes it easy to fall into local circulation and reduces the efficiency of chaos optimization. The Fuch map proposed has stronger chaotic characteristics than logistic mapping and tent mapping [31]. This mapping has the advantages of insensitivity to initial value, ergodic equilibrium, and faster convergence and can generate chaos even when the initial value is not zero. Because of the above analysis, this paper uses Fuch mapping as the mapping function of chaotic search, which is specifically defined as follows:

$$c_{i+1} = \cos \frac{1}{c_t}, \quad (10)$$

where $c_i \neq 0, i \in Z^+$.

Since the value range of variables generated by logistic mapping and tent mapping is $[0, 1]$ when mapping the decision variable x_i into a chaotic vector for chaotic search, it is necessary to first limit the chaotic initial value of each dimension x_i to $[0, 1]$ by the following equation:

$$l_{id} = \frac{x_{id} - x_{id}^{\min}}{x_{id}^{\max} - x_{id}^{\min}}. \quad (11)$$

Then, carrier operation is carried out on the generated chaotic initial vector l_i through logistic mapping or tent mapping function to generate chaotic vector c_i . Finally, it is transformed into the spatial vector of the original solution by the following equation:

$$x_{id}' = c_{id}(x_{id}^{\max} - x_{id}^{\min}) + x_{id}^{\min}. \quad (12)$$

Fuch mapping does not depend on the initial value; the chaotic sequences generated when the initial value changes slightly are completely different and irregular [32]. Therefore, when searching for chaos, it is not necessary to first transform the decision variable x_i into the chaotic variable interval by equation (11). The chaotic vector can be directly generated by equation (10). After carrier operation, it is restored to the original solution space.

3.3. Fuch Chaotic Search. Assuming that $x^i = (x_1^i, x_2^i, \dots, x_D^i)$ indicates the solution vector to be optimized, with D indicating the number of dimensions $x_k^i =$

$[x_k^{\min}, x_k^{\max}]$, where x_k^{\min} and x_k^{\max} indicate the minimum and maximum values of the k -th dimension of the solution vector, respectively, the maximum iteration number of chaos searches is MaxCt, and $F(\square)$ indicates the objective function, the steps of Fuch chaotic search proposed in this paper are as follows:

Step 1. Initialize the initial value $c_k^i \neq 0$ of chaotic vector iteration, where $i = 1, 2, \dots, \text{MaxCt}, k = 1, 2, \dots, d$.

Step 2. Substitute into equation (6) to perform Fuch mapping iteration to generate chaotic vector.

Step 3. Carry the chaotic vector c_k^{i+1} to the original solution space to generate a new solution by using the following equation:

$$x_{\text{new}}^i = x_k^{\min} + (x_k^{\max} - x_k^{\min}) \frac{(c_k^{i+1} + 1)}{2}. \quad (13)$$

Step 4. Repeat steps 2–4 until the maximum number of chaotic searches MaxCt is reached, and keep the solution x_{new}^i with the optimal fitness value.

Step 5. Compare the new solution x_{new}^i with the solution x^i to be optimized, and keep the optimal solution.

3.4. Fuch Chaotic Reverse Initialization Strategy. To overcome the defects caused by random population initialization in the BA, this study combines Fuch chaotic initialization with reverse learning initialization and proposes Fuch chaotic reverse learning initialization strategy. In fact, according to the probability theorem, the probability that the current solution is farther away from the optimal solution than its corresponding inverse solution is 50%, so searching the current solution and the inverse solution at the same time and keeping the optimal solution can make the population distribute as evenly as possible in the search space.

In this study, it is found that the upper and lower bounds of the optimization variable are symmetrical, and the benchmark function has a square term or an absolute term; if the general reverse learning strategy is adopted, the initial solution and the reverse solution are only opposite in sign, but the corresponding fitness values are the same. Therefore, when selecting the initial population, the same fitness value corresponds to multiple solutions, which reduces the diversity of population fitness values and often produces larger average fitness values. To reduce this phenomenon, this study proposes a new equation for generating the inverse solution:

$$\text{op}_j^i = \begin{cases} -x_j^i + \frac{(x_j^{\max} - x_j^{\min})}{2}, & x_j^i \geq 0, \\ -x_j^i + \frac{(x_j^{\max} - x_j^{\min})}{2}, & x_j^i < 0. \end{cases} \quad (14)$$

Each dimension of the initial solution is mapped to the vicinity of the interval center value. Moreover, each dimension of the generated inverse solution is no longer just the inverse number of the corresponding initial solutions, so

the inverse solution has different fitness values, which further improve the diversity of population fitness values and the quality of the initial solution.

If NP is the number of bat individuals, the initialization steps are as follows:

Step 1. Iterate by equation (10) to generate a chaotic vector sequence:

$$c^i = (c_1^i, c_2^i, \dots, c_D^i), \quad i = 1, 2, \dots, \text{NP}. \quad (15)$$

Step 2. Carry the chaotic vector to the original solution space by equation (13) to generate the initial solution x^i .

Step 3. Get the corresponding inverse solution from the initial solution x^i according to equation (14).

Step 4. The initial solution x^i and the reverse solution op^i are sorted according to the fitness value, and the top NP solutions are selected as the initial solutions of the population.

3.5. Local Search for the Best Individual Evolution Guidance.

In the BA, local search relies on the random walk in the neighborhood of the current optimal solution, which leads to the lack of mutation mechanism in the population. Local search adopts the method of mutating randomly selected individuals, which ignores the population optimal individuals and cannot make full use of the location information left by them [32]. Therefore, we learn from the differential evolution operator with strong local development ability, take the best individual of the population as the reference position, and introduce mutation and crossover operation to guide the movement of the population. The mutation operator is as follows:

$$v_{i,j}^{t+1} = x_j^* + \varphi_{i,j}(x_{r_1,j}^t - x_{r_2,j}^t), \quad (16)$$

where $\varphi_{i,j} \sim U[0, 1]$, $j \in \{1, 2, \dots, d\}$.

The difference between $x_{r_1,j}^t$ and $x_{r_2,j}^t$ indicates that when performing difference disturbance on the optimal solution x^* , the smaller the difference vector, the smaller the disturbance. At the initial stage of iteration, the larger the difference between different individuals, the greater the disturbance. So the search is performed around the optimal solution in a wide range. As the individual gradually approaches the optimal solution in the later stage of iteration, the disturbance is small. The search range near the optimal solution is reduced to enhance the local search ability and improve the convergence performance. In equation (16), the new solution is to exploit a new search area near the optimal solution obtained in the last iteration and randomly select two individuals to make a difference to guide bats to search the optimal area as much as possible to improve their development ability. Cross-operation of the individuals generated by variation is performed according to

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & \text{rand}(0, 1) \leq \text{Crorj} = j_{\text{rand}} \\ x_j^*, & \text{otherwise} \end{cases}, \quad (17)$$

where Cr is the crossover probability, with a value within $[0, 1]$, and j_{rand} is an integer randomly selected from $[1, D]$. Individual crossover ensures that the value of at least one dimension can be obtained from the mutant individual, thus ensuring that the crossover individual is different from the current optimal individual and mutant individual and avoiding invalid crossover [25]. Otherwise, no new individuals will be produced, resulting in no evolution of the population. The search will also come to a halt.

3.6. Introducing the Chaotic Global Search of Investigation.

In the artificial bee colony algorithm, if the honey source has not been improved for L consecutive iterations, the corresponding honey collecting bees will give up the honey source and change to scout bees to search for new honey sources, which is beneficial to get rid of the local optimum and enhance the global optimization performance. Inspired by the idea of scout bees, bats trapped in local optimum are defined as scout bats in the bat algorithm, and new positions are searched in the whole search space to replace the old ones [25]. However, the scout bats' research for the location is not just like the scout bees' giving up the old location and searching randomly. They search for the new location with two chaotic search strategies, respectively. For each scout bat, based on the current position and random position, by using Fuch mapping with good ergodic equilibrium, an intermediate solution is generated by a chaotic search strategy of a certain number of times. After comparing with the original position, the optimal solution is kept as the current position. In this way, the bats constantly jump out of the local extremum.

For the scout bats x_i^t and c_i^t in the T -th iteration, they indicate the number of Fuch chaotic sequences generated according to equation (10), and their chaotic search based on the current position is as follows:

$$x_{i,k}^{t+1} = x_{i,k}^t + (2c_i^t - 1)(x_{i,k}^t - x_{j,k}^t), \quad (18)$$

where $k \in \{1, 2, \dots, d\}$, $I, j \in \{1, 2, \dots, \text{NP}\}$, and $i \neq j$. Chaos search based on random position is as follows:

$$x_{i,k}^{t+1} = x_{i,k}^{\min} + c_i^t(x_k^{\max} - x_k^{\min}), \quad (19)$$

where x_k^{\min} and x_k^{\max} indicate the minimum and maximum values of the K -th dimension of the solution vector, respectively. At this time, the two intermediate solutions are searched by chaos iteration for a certain number of times according to the described steps, and finally, the better solution is selected.

For the above equation, the exploration ability is certainly improved by flying from the current position to other randomly selected positions, which, however, may not be a better position. This may lead the flight to a worse position. This method limits the search to the neighborhood of the current position [33]. Although the nearby areas are effectively exploited, the effect of better searching the whole solution space cannot be achieved. By further expanding the search scope and combining it with chaotic mapping, the whole solution space can be traversed effectively, but the

subtle search cannot be carried out, and the information around the current individual is ignored. Therefore, the two search strategies are used as independent ways to generate intermediate solutions. Their respective advantages are combined while avoiding their shortcomings so that individuals trapped in local optimum can independently choose a better new position to replace the original position. The combination of these two strategies can constantly get rid of the constraints of the local extremum and enhance the global exploration ability.

3.7. Adaptive Chaotic Updating of Parameters. In the basic BA, the pulse emittance r controls the mutual conversion between global search and local search, and the loudness A determines whether to receive the new solution, which affects the optimization efficiency to some extent. Therefore, this study proposes a strategy of adaptive updating of parameters with the change of iteration times and chaotic sequence, as follows:

$$\begin{cases} A^{t+1} = \left[\left(\frac{A_i - A_f}{1 - t^{\max}} \right) (t - t^{\max}) + A_f \right] |c^t| \\ r^{t+1} = \left[\left(\frac{r_i - r_f}{1 - t^{\max}} \right) (t - t^{\max}) + r_f \right] |c^t| \end{cases}, \quad (20)$$

where A_i and A_f are the initial and final values of loudness A , r_i and r_f are the initial and final values of pulse emissivity, t is the current iteration number, t^{\max} shows the maximum iteration number, and c^t is the chaotic sequence number. The renewal equation of A is the linear decreasing function multiplied by a chaotic number, and the renewal equation of R is the linear increasing function multiplied by the chaotic number. The linear function accelerates the updating speed of parameters, and the chaotic sequence ensures that the values can traverse the whole interval, thus controlling the balance between development capability and exploration capability.

3.8. Chaotic Bat Algorithm with Evolution and Search Mechanism (ESCBA) Flow. To sum up, the implementation process of the chaotic bat algorithm with evolution and search mechanism (ESCBA) is as follows:

Step 1. Initialize algorithm-related parameters, including the number of individual bats NP, variable dimension number D , maximum iteration number MaxIt, search threshold L , chaotic search number MaxCt, and individual update flag bit T.

Step 2. According to the description, initialize the population with a chaotic reverse learning strategy, calculate its fitness value, and find the optimal individual x^* .

Step 3. Traverse all individuals, update the frequency, speed, and position information according to equation (7), and generate a new individual x^{new} .

Step 4. Generate a random number R_1 . If R_1 is larger than r_i , perform mutation and crossover operation

according to equations (15) and (16), and reassign the individual after evolution to x^{new} .

Step 5. Generate a random number R_2 . If the fitness x^{new} is better than that of the current individual and R_2 is less than P4, accept the new individual, reassign T to 0, update A and r according to equation (19), and go to step 7; otherwise, go to step 6.

Step 6. Increase the value of T by 1. If T is greater than or equal to L , then carry out MaxCt subchaotic search according to equations (17) and (18), respectively, to generate intermediate individuals med_1 and med_2 . Select the better intermediate value and compare it with the current individual, and keep the optimal value as the new position of the current individual; if T is less than L , go to step 7.

Step 7. Compare the obtained new individual with the optimal individuals in the population. Update the optimal individual, and judge whether the iteration times reach MaxIt. If so, output the searched optimal solution and the corresponding fitness value and exit; otherwise, go to step 3.

4. ESCBA-GRU Combined Model

4.1. GRU Super Parameter Optimization Strategy. Since the parameter adjustment process of the GRU neural network is blind and random, the bat algorithm is introduced to complete the optimization work [34]. Firstly, the number m of parameters that the GRU neural network needs to adjust is determined, and it is regarded as an m -dimensional vector that needs to be processed by the bat algorithm. Secondly, determine the value range of this m -dimensional vector; that is, lock the vector space of this m -dimensional vector that needs to be optimized by the bat algorithm. Then, regard GRU neural network as a function, which is called and executed by the bat algorithm. Every time, this m -dimensional vector is used as the super parameter of GRU to train the data model. Finally, the trained network model is used to predict the test data, and the error value is obtained by comparing the predicted value with the real value. The error value is returned to the bat algorithm as the evaluation index of optimization. To clearly show the error between the predicted data and the true value, the mean square error (MSE) function and LOSS function are introduced to express the difference between the predicted value and the original value. The language description of this function is to calculate the sum of squares of distances at corresponding points between predicted data and original data. The evaluation function is expressed as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2, \quad (21)$$

$$\text{LOSS}(\hat{y}, y) = \sum_{i=0}^m (y^{(i)} - \hat{y})^2,$$

where parameter N indicates the number of samples, y_i is the true value of the samples at the i th position, and \hat{y} indicates the predicted value of the i th position predicted by the

TABLE 1: Reference function.

Function	Name	Search range	Minimum value	Characteristics
f1	Ackley	[-32, 32]	0	Multimodal
f2	Griewank	[-600, 600]	0	Multimodal
f3	Rastrigin	[-5.12, 5.12]	0	Multimodal
f4	Sphere	[-100, 100]	0	Unimodal
f5	Schwefel	[-10, 10]	0	Unimodal
f6	Shifted Rastrigin	[-5, 5]	-330	Multimodal
f7	Shifted Rosenbrock	[-100, 100]	390	Multimodal
f8	Shifted Schwefel	[-100, 100]	-450	Unimodal
f9	Shifted Sphere	[-100, 100]	-450	Unimodal

model. The geometric meaning $(y_i - \bar{y})^2$ shows the distance between the true value and the predicted value at the same point, which is used here to measure the gap between the predicted value and the true value. When the gap between the predicted value and the true value is large, whether it is positive or negative, it will get a large value. That is, it is far from the true value. In this algorithm, the result value of MSE will be used as the evaluation function, and the numerical value will be fed back to the improved bat algorithm for the evaluation and judgment of the current optimization result value. The optimization algorithm will judge whether the current position is better than the historical point through the evaluation result to decide whether to update the current point.

4.2. Execution Process of the Combined Model. The purpose of constructing the combined GRU model based on the optimized bat algorithm is to improve the results of data prediction by the neural network and reduce the error between the predicted value and the true value. According to the description of the combined neural network model in the previous section, it can be concluded that the steps of the model in the process of execution are as follows:

Step 1. Initialize parameters, including the population size N of the algorithm, the dimension D of the problem variable, and the maximum iteration times of the algorithm.

Step 2. Determine the adaptive function and the LOSS function after GRU training, and calculate the mean square error to get the evaluation index.

Step 3. Search for the best GRU parameter in the parameter vector space.

Step 4. Complete the optimization process of the ESCBA and obtain the optimal solution. The results are imported into GRU neural network as parameters to complete the data prediction.

Step 5. According to the prediction result, calculate the error value, and return the error value to ESCBA as the evaluation index. Finally, update the optimal solution position according to the evaluation index.

Step 6. Iterate step 3, step 4, and step 5 until the iteration requirements are met or the maximum number of iterations is reached.

Step 7. Get the network parameters of the optimal solution, import them into the neural network, and complete the model training.

Step 8. Use the model to predict and analyze the test data.

Step 9. Get the prediction result value.

Step 10. End

5. Experimental Analysis

5.1. Experimental Data. The experimental data of this study was from a bank's data warehouse, and the target data collected were the transaction volume data of the bank's core system. The data collection time was from July 20, 2018, to July 1, 2020. The data were summarized in minutes, with a total of 1,026,721 samples of data. Among them, the data from July 20, 2018, to January 30, 2020, were selected as model training data, including 763,349 samples of data. The data from January 30, 2020, to July 1, 2020, were selected as test data to test the fitting effect of the model, including 263,372 samples of data.

5.2. Data Processing. For the abnormalities of the collected data, according to the convention of cleaning data with big data, the missing part of the collected data was filled with the average value of the data in the same period of this point. After cleaning, the experimental data were processed and normalized in the range of [0, 1]. After cleaning and processing, the data had a good comprehensive contrast property and were all at the same data level.

5.3. Experimental Analysis of Bat Algorithm. Nine benchmark functions with different characteristics were selected to test the performance of the bat algorithm and make a full investigation of the optimization ability of the algorithm. The function name, value range, global minimum, and characteristics are shown in Table 1. Among them, f4, f5, f8, and f9 are unimodal functions, mainly testing the optimization accuracy and convergence speed of the algorithm. f1, f2, f3, f6, and f7 are multimodal functions, mainly testing the global search of the algorithm and its ability to jump out of the local extremum to avoid premature convergence.

TABLE 2: Performance comparison of different initialization strategies.

Function	Random initialization		General reverse learning		Fuch chaos inversion	
	Fit	Driver	Fit	Driver	Fit	Driver
f1	20.7497	18.1645	20.7483	18.4191	20.5793	19.8864
f2	1.12E + 03	341.662	1.06E + 03	343.9945	960.117	372.0474
f3	3.26E + 03	8.5075	3.23E + 03	8.6192	2.77E + 03	9.3072
f4	1.21E + 05	56.8516	1.20E + 05	57.3976	1.08E + 05	62.1469
f5	3.85E + 23	5.674	5.80E + 22	5.7708	3.84E + 11	6.212
f6	469.0284	2.842	450.6013	2.8784	368.8445	3.1041
f7	6.10E + 10	56.93	5.62E + 10	57.5255	4.65E + 10	62.2366
f8	1.33E + 06	56.9289	7.37E + 05	57.6686	3.87E + 05	62.0767
f9	1.23E + 05	56.9036	1.21E + 05	57.5548	9.81E + 04	62.1608

TABLE 3: Comparison of optimization accuracy in 30 dimensions among all algorithms.

Function		BA	HSBA	BAGW	dBA	SABA	ESCBA
f1	Best	17.7596	4.2553	0.9103	3.251	1.30E - 10	1.33E - 14
	Worst	19.9668	13.8567	19.9652	19.9668	5.09E - 10	2.45E - 10
	Mean	19.7401	9.8034	19.004	4.5794	2.87E - 10	3.53E - 11
	Std	0.5893	2.3872	3.8096	3.2719	7.32E - 11	5.53E - 11
f2	Best	70.93	5.71E - 03	1.30E - 04	4.93E - 10	4.44E - 16	0
	Worst	299.824	0.0918	0.0224	0.0221	0.0295	1.11E - 15
	Mean	167.9349	0.033	5.68E - 03	7.68E - 03	0.0037	3.91E - 16
	Std	56.9986	0.0238	7.52E - 03	7.92E - 03	0.007	3.89E - 16
f3	Best	317.0197	169.1424	49.2816	115.9538	8.88E - 15	0
	Worst	806.1842	406.9305	236.1797	288.302	0.995	2.27E - 13
	Mean	526.4121	278.6654	139.8219	218.8277	0.0404	2.96E - 14
	Std	133.5952	61.605	39.6879	50.9537	0.1969	5.94E - 01
f4	Best	0.0101	2.05E - 04	4.75E - 04	7.90E - 09	1.70E - 19	1.90E - 27
	Worst	0.194	0.0104	4.65E - 03	2.30E - 05	5.17E - 18	3.39E - 20
	Mean	0.0543	2.39E - 03	2.27E - 03	1.61E - 06	1.31E - 18	1.36E - 21
	Std	0.0465	2.56E - 03	8.60E - 04	4.65E - 06	8.66E - 19	6.78E - 02
f5	Best	1.0537	0.1848	0.2578	1.30E - 03	5.40E - 10	7.32E - 14
	Worst	5.82E + 05	21.1952	258.5819	0.9999	2.91E - 04	8.23E - 06
	Mean	2.41E + 04	4.1951	24.0055	0.0728	1.70E - 05	3.73E - 07
	Std	1.16E + 05	4.8816	60.459	0.1988	5.52E - 05	1.64E - 06
f6	Best	-78.6817	-276.2715	-261.0051	-290.2633	-330	-330
	Worst	63.6291	-196.6759	-149.453	-162.8861	-329.005	-330
	Mean	-2.8249	-224.5311	-198.5203	-237.9454	-329.9801	-330
	Std	33.7651	20.2644	37.5409	26.9607	0.1407	6.46E - 14
f7	Best	414.8215	414.4444	413.0764	407.6861	390.0006	390
	Worst	2.36E + 03	1.19E + 03	9.54E + 03	706.0761	467.0134	390
	Mean	660.7911	524.9373	1.40E + 03	468.4775	408.4709	390
	Std	438.4634	175.1471	2.12E + 03	101.6758	15.6599	2.54E - 08
f8	Best	-442.9437	-333.7772	-449.6903	-449.7759	-450	-450
	Worst	3.46E + 03	2.90E + 03	-448.4075	-449.0023	-449.5068	-449.9856
	Mean	571.2265	496.6318	-449.1955	-449.4091	-449.9609	-449.9969
	Std	1.04E + 03	915.4484	0.3811	0.2215	0.0791	4.77E - 03
f9	Best	-449.986	-449.9996	-449.9992	-450	-450	-450
	Worst	-449.8731	-449.9402	-449.9953	-450	-450	-450
	Mean	-449.9543	-449.9941	-449.9977	-450	-450	-450
	Std	0.0303	0.0121	8.61E - 04	6.21E - 06	1.20E - 13	4.64E - 14

5.4. ESCBA Initialization Performance Analysis. In this paper, random initialization, general reverse learning initialization, and Fuch chaotic reverse initialization are used, respectively. By calculating the mean value of initial population fitness (Fit) and diversity (Diver), the performance of different initialization strategies is compared. The population diversity is computed as follows:

$$\text{Diver} = \frac{1}{NP} \sum_{i=1}^{NP} \sqrt{\frac{1}{D} \sum_{d=1}^D (x_{i,d} - m_d)^2}, \quad (22)$$

where NP indicates the population number and m indicates the average initial solution. The greater the diversity, the

TABLE 4: Comparison of optimization accuracy in 50 dimensions among all algorithms.

Function		BA	HSBA	BAGW	dBA	SABA	ESCBA
f1	Best	19.9643	10.0628	19.9628	4.4953	2.28E-09	1.80E-11
	Worst	19.9668	15.4746	19.9648	19.9668	1.89E-07	1.16E-08
	Mean	19.9652	13.4656	19.9645	8.0741	3.78E-08	2.01E-09
	Std	5.46E-04	1.3904	4.94E-04	4.5963	4.56E-08	2.59E-09
f2	Best	172.5732	0.4679	1.08E-03	4.47E-05	9.37E-14	1.55E-15
	Worst	530.4195	1.145	4.6493	0.0175	0.0344	4.10E-10
	Mean	320.3457	0.9123	0.3484	6.57E-03	0.003	3.59E-11
	Std	96.8166	0.1757	0.9989	7.27E-03	0.0071	9.07E-01
f3	Best	605.8356	360.5412	177.8822	297.5844	2.06E-13	2.84E-13
	Worst	1.36E+03	685.127	526.9451	538.5685	2.0008	8.29E-10
	Mean	1.04E+03	507.1378	334.6462	430.5927	0.242	1.59E-10
	Std	175.8792	83.0607	86.3076	66.059	0.6215	2.74E-10
f4	Best	0.044	0.5955	7.96E-03	9.78E-04	1.38E-16	6.87E-32
	Worst	3.8593	24.8722	0.0268	9.77E-03	3.45E-12	3.03E-13
	Mean	0.2882	3.9364	0.0158	4.41E-03	1.18E-13	1.23E-14
	Std	0.7518	5.2077	4.91E-03	2.61E-03	5.17E-13	6.05E-14
f5	Best	2.0848	5.8794	1.529	0.4796	1.25E-05	1.09E-12
	Worst	7.66E+08	34.682	9.62E+09	3.9149	0.1548	1.79E-05
	Mean	4.29E+07	15.5104	4.00E+08	1.1988	0.0124	3.48E-06
	Std	1.57E+08	8.0402	1.92E+09	0.7185	0.0301	5.54E-06
f6	Best	174.4835	-201.8228	-178.8785	-212.3889	-330	-330
	Worst	359.0934	-31.9945	64.8509	-91.7885	-328.0098	-330
	Mean	278.8857	-128.393	-60.2807	-148.2952	-329.8386	-330
	Std	46.8769	43.0127	5.53E+01	29.8312	0.4647	2.61E-09
f7	Best	447.9727	856.3206	435.6838	434.0916	390	390
	Worst	8.90E+04	1.10E+04	6.03E+03	1.47E+03	541.1077	390
	Mean	7.10E+03	2.69E+03	1.12E+03	574.5079	431.3087	390
	Std	2.09E+04	2.32E+03	1.54E+03	206.452	29.0897	5.77E-06
f8	Best	1.17E+03	2.92E+03	-382.2297	-419.309	-448.3683	-450
	Worst	4.81E+04	2.27E+04	-261.4164	-383.9354	-344.2566	-449.1829
	Mean	1.56E+04	1.19E+04	-347.715	-406.4232	-425.8122	-449.937
	Std	1.04E+04	6.16E+03	27.0176	9.0109	20.4383	0.1762
f9	Best	-449.9699	-449.2918	-449.993	-449.9995	-450	-450
	Worst	1.03E+03	-355.2089	-449.9705	-449.9902	-450	-450
	Mean	-386.1564	-439.1937	-449.9851	-449.9964	-450	-450
	Std	295.4655	18.5233	5.14E-03	2.28E-03	1.20E-12	2.88E-08

more dispersed the population distribution; otherwise, it is dense. The smaller the fitness, the higher the quality of the initial population solution.

In the experiment, the nine benchmark functions all take 50 dimensions. The maximum number of iterations is 500, and the population size is 40. All of them run independently 50 times. It can be seen from Table 2 that, in the test of each function, the average fitness and average diversity of the population initialized by the Fuch chaotic reverse learning strategy are better than the other two initialization strategies, showing that the Fuch chaotic reverse strategy of equation (9) can expand the diversity of the population to a certain extent and improve the quality of the initial solution.

5.5. ESCBA Optimization Accuracy Analysis. To verify the performance of the ESCBA, this study selected the basic BA [1], BAGW [6], HSABA [17], dBA [18], and SABA [19] for the comparative experiment. In the experiment, the bat

number NP of all algorithms was set to 40, and the maximum iteration (Imax) was 500. Except that, the parameter settings of other comparison algorithms were consistent with their respective literature. The specific settings of each parameter in the ESCBA were as follows: the maximum pulse frequency fmax=2; the minimum pulse frequency fmin=0; the crossover probability Cr=0.75; the search threshold L=25; the range of loudness A [0.6, 0.9]; the range of pulse emittance R [0.1, 0.7]; the number of chaotic searches Cmax=100. After each algorithm runs independently 50 times for 9 benchmark functions in 30 dimensions, 50 dimensions, and 100 dimensions, the best, the worst, the mean, and the standard deviation (Std) were calculated for the optimization results, respectively, as shown in Tables 3–5.

From the results of Tables 3–5, with the increase of dimension, the accuracy of each algorithm decreases. On the whole, the optimization effect of ESCBA and SABA is better than other control algorithms. Among them, for f2, the mean and variance of SABA are not as good as BAGW, dBA,

TABLE 5: Comparison of optimization accuracy in 100 dimensions among all algorithms.

Function		BA	HSBA	BAGW	dBA	SABA	ESCBA
f1	Best	19.9657	13.5022	19.9647	8.4357	4.30E-04	9.56E-09
	Worst	19.9668	19.9646	19.9661	19.9668	0.0101	1.67E-04
	Mean	19.9661	15.0036	19.9657	14.3058	0.0029	4.21E-05
	Std	2.49E-04	1.2718	2.32E-04	4.6474	0.0023	5.08E-05
f2	Best	463.8012	3.8326	115.6962	4.88E-03	2.23E-05	2.88E-14
	Worst	1.38E+03	34.1334	256.5072	0.0262	0.0492	3.31E-06
	Mean	803.5183	11.0961	212.4665	0.0136	0.0028	2.23E-07
	Std	234.716	6.055	33.4015	6.15E-03	0.01	6.65E-07
f3	Best	1.88E+03	759.4136	834.4371	872.5314	3.55E-05	7.78E-11
	Worst	4.36E+03	1.53E+03	2.16E+03	1.25E+03	1.066	3.77E-05
	Mean	2.95E+03	1.22E+03	1.10E+03	1.08E+03	0.1547	3.14E-06
	Std	573.184	199.7152	267.2813	95.9341	0.3519	7.72E-06
f4	Best	0.7806	242.2646	0.1839	0.1529	3.88E-06	1.21E-31
	Worst	2.21E+04	2.99E+03	0.4049	0.3586	1.38E-04	4.19E-08
	Mean	5.02E+03	1.36E+03	0.2709	0.2468	2.25E-05	2.16E-09
	Std	6.34E+03	712.0796	0.0544	0.0457	2.16E-05	8.48E-09
f5	Best	29.5942	55.9458	318.6646	38.7299	0.0048	6.78E-07
	Worst	2.05E+32	198.8244	1.02E+27	140.6062	0.2658	1.01E-03
	Mean	8.21E+30	91.4364	4.92E+25	54.7943	0.0799	2.43E-04
	Std	4.11E+31	29.3236	2.07E+26	18.7261	0.0644	2.95E-04
f6	Best	167.2032	-10.1364	267.0111	158.6915	-329.9999	-330
	Worst	343.7372	24.6121	587.555	356.279	-327.985	-330
	Mean	280.671	100.3726	365.1652	234.9092	-329.8393	-330
	Std	46.23	67.0494	93.3562	52.4314	0.3965	1.53E-05
f7	Best	1.72E+03	1.09E+06	567.6221	608.3885	390.1261	390
	Worst	2.00E+08	2.57E+07	7.98E+03	7.07E+03	776.6743	390.0004
	Mean	2.37E+07	4.55E+06	1.65E+03	2.09E+03	509.5479	390
	Std	4.84E+07	4.79E+06	2.23E+03	2.08E+03	96.7224	7.87E-05
f8	Best	5.80E+04	7.29E+04	8.80E+03	5.51E+03	-191.8983	-450
	Worst	2.64E+05	2.27E+05	1.92E+04	1.09E+04	2.72E+03	-448.2102
	Mean	1.21E+05	1.18E+05	1.26E+04	7.29E+03	882.4828	-449.8668
	Std	4.78E+04	3.79E+04	2.20E+03	1.33E+03	878.9749	0.3833
f9	Best	-449.2562	87.1221	-449.834	-449.8136	-450	-450
	Worst	2.46E+04	2.66E+03	-449.6486	-449.6389	-449.9999	-449.9999
	Mean	3.87E+03	726.3181	-449.7498	-449.7574	-450	-450
	Std	6.99E+03	571.7735	0.0535	0.044	1.88E-05	2.29E-05

and ESCBA in 30 dimensions, and it is not as good as dBA and ESCBA in 50 dimensions. Except for f2 and f3 in 30 dimensions and 50 dimensions and except for f6 and f7 in 100 dimensions, dBA's solution performance is better than that of HSABA, BAGW, and BA, and BA has poor optimization performance. Whether for unimodal function or multimodal function, ESCBA has better performance than other algorithms in different dimensions, among which ESCBA has more obvious advantages in 100-dimensional function. For multimodal functions f2, f3, f6, and f7, the accuracy of ESCBA in each dimension is improved compared with SAB. For f6, f7, f8, and f9, ESCBA can find the theoretical optimal value in different dimensions, which indicates that ESCBA has a stronger global searching ability. Moreover, for f8, with the increase of dimension, the optimization value of SABA has a large deviation, and the variance further increases, while ESCBA still keeps a good optimization result. In addition, for f9 in 30 dimensions and 50 dimensions, ESCBA and SABA both reach the theoretical optimal value. For f3 in 50 dimensions, SABA gets the best

TABLE 6: Errors of data prediction results of GRU combined neural network.

Algorithm	MSE	RMSE	MAE
ARMA	138022.90	371.51	282.92
GRU	29297.17	171.16	134.30
ESCBA-GRU	24539.69	156.65	111.85

solution. But its mean and variance are not as good as ESCBA, and the difference is obvious. Except for f9 in 50 and 100 dimensions, ESCBA has the smallest standard deviation in different dimensions of other functions, which shows that ESCBA has good stability and robustness as a whole.

5.6. GRU Combination Model Trading Volume Prediction Experiment. To verify the ability of the GRU combination model to predict the transaction volume data of the core accounting system of commercial banks, the GRU model was trained based on the training data set to generate a data

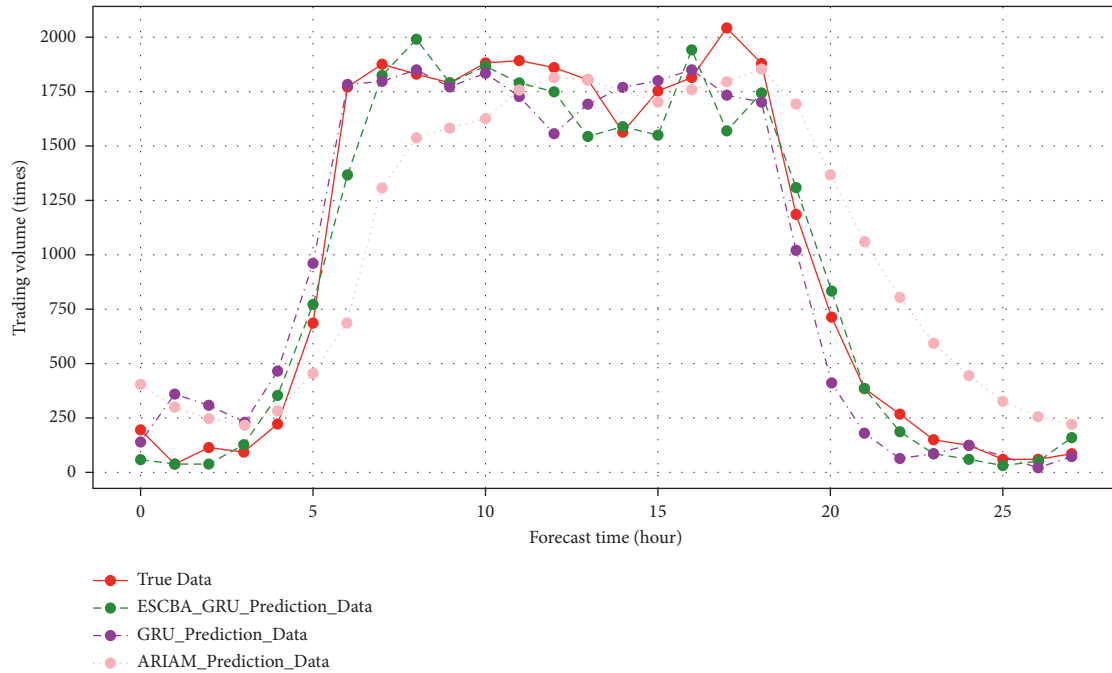


FIGURE 3: Data prediction of GRU combined model and the curve of true values.

model. Finally, test data are used to test the predictive ability of the model. In the experiment, to verify the predictive ability of the GRU combined model, the original GRU model and ARAM model are introduced to predict the experimental data. The experimental data are shown in Table 6, and the curve change of the predicted data is shown in Figure 3.

It can be seen from the experimental data table that the prediction error of the improved combined neural network model is 17% lower than that of the standard GRU neural network model. Besides, the RMSE evaluation index is 8% lower, and the MAE evaluation index is 17% lower, respectively. The prediction effect of the combined neural network on the result value is better than that of the standard GRU neural network.

Figure 3 shows that the curve change of the predicted value of the GRU combined neural network is closer to the true solution curve, and many points coincide with the true value. This fitting effect has a certain guiding significance for data decision-making.

6. Conclusion

Commercial banks are extremely important for social and economic growth. As a result, predicting the transaction volume of the core accounting system is of both theoretical and practical significance. In this study, the theoretical basis and optimization process of the bat algorithm is systematically studied, and the characteristics and shortcomings of this algorithm are deeply analyzed. Based on keeping the bat algorithm to simulate the hunting behavior of the bat population, a chaotic bat algorithm with evolution and search mechanism (ESCBA) is proposed, which introduces Fuch chaotic reverse learning strategy to improve the initial

population diversity. The evolutionary mechanism is introduced to ensure that the population can make full use of the guidance of optimal individual information; the search mechanism further enhances the ability to get rid of the local extremum. The control parameters A and R are adaptively updated so that the algorithm tends to be stable and convergent and keeps the balanced exploration and development ability. Finally, based on ESCBA, a model combined with GRU neural network is proposed, and the super parameters of the GRU neural network are optimized by giving full play to the advantages of the optimization algorithm. Results showed that the improved combined GRU model performs well in MSE, RMSE, and MAE, which are superior to the original GRU model and the traditional time-series forecasting ARMA model. Future work will focus on further exploring the function of the GRU combined model and the realization of automatic deployment and startup management of application systems according to the prediction data and building intelligent and dynamic application management and early warning system.

Data Availability

The data used in this research are confidential and not open to the public.

Disclosure

This article was originally published in The International Conference on Natural Computation Fuzzy Systems and Knowledge Discovery.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] Y. Quan and Y. Zonghan, "Markov particle filter traffic flow prediction model," *Journal of Transportation Systems Engineering and Information Technology*, vol. 19, no. 2, pp. 209–215, 2015.
- [2] Y. Huo and M. Xiaoxuan, "Research on mine image denoising algorithm based on Kalman filter and genetic algorithm," *Computer applications and software*, vol. 39, no. 02, pp. 220–227, 2022.
- [3] P. Jiao, T. Sun, T. Z. Hou, and A. au, "Three revised kalman filtering models for short-term rail transit passenger flow prediction," *Mathematical Problems in Engineering*, vol. 2016, Article ID 9717582, 10 pages, 2016.
- [4] W. Shiqiang, T. Bin, L. Bing, and G. Youcheng, "Power frequency magnetic anomaly search under complex observation conditions based on deep learning," *Automation and Instrumentation*, vol. 37, no. 02, pp. 50–53+58, 2022.
- [5] D. Zhongxing, W. Qian, Z. Meng, S. Jiefei, and W. Jian, "Short-term passenger flow forecast based on improved bat algorithm to optimize LSTM network," *Journal of Railway Science and Engineering*, vol. 18, no. 11, pp. 2833–2840, 2021.
- [6] Y. Wang, "Prediction algorithm of meteorological elements based on GRU network," *Computer Simulation*, vol. 38, no. 07, pp. 419–423, 2021.
- [7] Y. Tong, "A Deep Learning Approach for Covid-19 Trend Prediction," <https://arxiv.org/pdf/2008.05644.pdf>.
- [8] Y. Liang, "Network intrusion search based on GRU and feature embedding," *Journal of Applied Sciences*, vol. 39, no. 04, pp. 559–568, 2021.
- [9] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO2010)*, Springer Publications, Berlin Heidelberg, pp. 65–74, 2010.
- [10] L. Xiao, F. Qian, and W. Shao, "Multi-step wind speed forecasting based on a hybrid forecasting architecture and an improved bat algorithm," *Energy Conversion and Management*, vol. 143, pp. 410–430, 2017.
- [11] A. Gholami and H. R. Ansari, "Estimation of porosity from seismic attributes using a committee model with bat-inspired optimization algorithm," *Journal of Petroleum Science and Engineering*, vol. 152, pp. 238–249, 2017.
- [12] G. G. Wang, H. E. Chu, and S. Mirjalili, "Three-dimensional path planning for UCAV using an improved bat algorithm," *Aerospace Science and Technology*, vol. 49, pp. 231–238, 2016.
- [13] B. R. Adarsh, T. Raghunathan, T. Jayabarathi, and X.-S. Yang, "Economic dispatch using chaotic bat algorithm," *Energy*, vol. 96, pp. 666–675, 2016.
- [14] X. Cai, L. Wang, and Q. Q. Kang, "Bat algorithm with Gaussian walk," *International Journal of Bio-Inspired Computation*, vol. 6, no. 3, pp. 166–174, 2014.
- [15] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 696491, 21 pages, 2013.
- [16] X. Huang, X. Zeng, and R. Han, "Dynamic inertia weight binary bat algorithm with neighborhood search," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 3235720, 15 pages, 2017.
- [17] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, 2014.
- [18] A. R. Jordehi, "Chaotic bat swarm optimization," *Applied Soft Computing*, vol. 26, pp. 523–530, 2015.
- [19] C. P. Liu and C. M. Ye, "Bat algorithm with chaotic search strategy and analysis of its property," *Journal of System Simulation*, vol. 25, no. 6, pp. 1183–1188, 2013.
- [20] X.-B. Meng, X. Z. Gao, Y. Liu, and H. Zhang, "A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6350–6364, 2015.
- [21] F. J. Kuang, Z. Jin, and W. H. Xu, "Hybridization algorithm of Tentchao sartificial bee colony and particle swarm optimization," *Control and Decision*, vol. 30, no. 5, pp. 839–847, 2015.
- [22] S. J. Zhao, L. F. Gao, and D. M. Yu, "Antlion optimizer with chaotic investigation mechanism for optimizing SVM parameters," *Journal of Frontiers of Computer Science and Technology*, vol. 10, no. 5, pp. 722–731, 2016.
- [23] W. F. Fu and Codling, "An adaptive iterative chaos optimization method," *Journal of Xi'anJiaotongUniversity*, vol. 47, no. 2, pp. 33–38, 2013.
- [24] X.-s. He, W.-J. Ding, and X.-S. Yang, "Bat algorithm based on simulated annealing and Gaussian perturbations," *Neural Computing & Applications*, vol. 25, no. 2, pp. 459–468, 2014.
- [25] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.
- [26] M. Nazari-Heris, B. Mohammadi-Ivatloo, and A. Haghrah, "Optimal short-term generation scheduling of hydrothermal systems by implementation of real-coded genetic algorithm based on improved Mühlenbein mutation," *Energy*, vol. 128, pp. 77–85, 2017.
- [27] M. Bahiraei, S. Nazari, H. Moayedi, and H. Safarzadeh, "Using neural network optimized by imperialist competition method and genetic algorithm to predict water productivity of a nanofluid-based solar still equipped with thermoelectric modules," *Powder Technology*, vol. 366, pp. 571–586, 2020.
- [28] T. Xiang, X. Liao, and K.-W. Wong, "An improved particle swarm optimization algorithm combined with piecewise linear chaotic map," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1637–1645, 2007.
- [29] S. Garcia and F. Herrera, "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.
- [30] A. Deb, J. S. Roy, and B. Gupta, "Performance comparison of differential evolution, particle swarm optimization and genetic algorithm in the design of circularly polarized microstrip antennas," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 8, pp. 3920–3928, 2014.
- [31] J. Vesterstrøm and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," vol. 2, pp. 1980–1987, in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 2, IEEE, Portland, OR, USA, June 2004, <https://doi.org/10.1109/CEC.2004.1331139>.
- [32] Y. Liu, X. Zhu, and J. Yang, "Fault diagnosis of PV array based on optimised BP neural network by Improved adaptive genetic algorithm," *Journal of Engineering*, vol. 2017, no. 13, pp. 1427–1431, 2017.
- [33] S. Xie, Z. Yu, and Z. Lv, "Multi-disease prediction based on deep learning: a survey," *Computer Modeling in Engineering and Sciences*, vol. 128, no. 2, pp. 489–522, 2021.
- [34] S. Vaidyanathan, B. A. Idowu, and A. T. Azar, "Backstepping controller design for the global chaos synchronization of spott's jerk systems," *Chaos Modeling and Control Systems Design*, vol. 581, pp. 39–58, 2015.