*Research Article*

# Underwater Internet of Things-Based Solutions for Intelligent Marine Target Recognition

**Ping Hu [iD],[1] Kaina Jiang [iD],[1] Peng Chen [iD],[2] Yingchun Huan [iD],[3] and Ruiyong Yue [iD][1]**

[1]*Dalian Institute of Measurement Ang Control Technology, Dalian, China*
[2]*No. 722 Research Institute, China State Shipbuilding Corporation Limited, Wuhan, China*
[3]*The PLA Unit, Beijing 91977, China*

Correspondence should be addressed to Ping Hu; 15942616861@163.com

In order to solve the problems of poor data processing ability of underwater hardware equipment and low accuracy of classification algorithms in the existing marine target recognition and detection methods based on sensors and transducers, by combining perception technology, underwater Internet of Things technology, and artificial intelligence, multiple devices could communicate with each other to achieve automatic and intelligent high-precision marine target recognition. Compared with existing methods, not only the accuracy rate is improved but also the hardware requirements are lower, and it is easier to deploy in engineering.

## 1. Introduction

Due to the limited computing power of embedded underwater devices and special signal transmission environment, underwater target recognition has always been a difficult engineering problem [1].

First, the sensor nodes need to be deployed in the monitoring sea area. The sensor array deployed in the ocean is difficult to maintain and upgrade, and due to the difficulty of energy supply, the energy consumption cannot be too large, resulting in limited computing power supplied for signal processing [2]. Second, underwater cables or underwater acoustic communication is generally used to send information to ground base stations. While the length of underwater cables is limited, the underwater acoustic signals have the characteristics of fast energy attenuation and narrow bandwidth, making it difficult to transmit or process large amounts of data in real time.

With the development of artificial intelligence and underwater communication technology, the emerge of Underwater Internet of Things (UIoT) technology provides a new way to solve this problem. UIoT can be regarded as an extension of the Internet of Things (IoT) to the marine and ocean environment, constitutes powerful way for solving complex tasks such as signal processing and transmission in the ocean intelligently. A UIoT system with five-layer architecture is presented in [3], with sensing, communicating, networking, fusion, and application layer included. It is also suggested that the UIoT can be combined with cloud computing, edge computing, and other artificial intelligence technologies in the future. In [4], an underwater data acquisition system assisted with unmanned aerial vehicles is proposed to link between vehicles and underwater sensors; multiple receiver nodes are placed on the surface of sea to act as intermediate relays.

There are many ways to manufacture and build UIoT system; for example, it can be constructed by fixed underwater nodes, AUVs, base stations, or satellites. The automatic high-precision target recognition by multiple devices that can communicating with each other could be realized. But there are still many difficulties in establishing underwater Internet of Things for marine target recognition. For example, a mobile task assignment method is designed in [5] to lead the AUVs to the specified positions under rough ocean environment and propose a formation control scheme to escort the target to the destination safely. To extend the

life of UIoT and overcome challenges of ocean network building project, such as high path loss, limited battery power, and available bandwidth, a new balanced energy adaptive routing protocol is designed in [6].

In addition to research on structures, there are transmission and communication technologies, and unmanned task assignment methods. The research of UIoT is carried out from all directions, making underwater intelligent target recognition and identification feasible. For target recognition problems, a wide variety of methods have been developed, such as clustering algorithm-based methods such as partition, hierarchical and network-based classifiers, deep learning methods such as computer vision-based methods.

Among various methods, neural network-based methods are the most attractive due to their excellent performance and generalization ability and have been widely studied and deployed in ground-based and airborne equipment. However, due to the complexity and computational difficulty of existing models, many models with excellent performance are difficult to deploy on low-power underwater processors. On the contrary, using the existing simple classifier directly for underwater target recognition cannot obtain ideal classification results. Therefore, it is necessary to select the classifier in a pertinent manner and to optimize its structure and parameters for a more reliable decision mechanism, so as to obtain a more robust and fast target recognition method.

Recurrent neural networks can analyse time series, the most representative of which is the long short-term memory (LSTM) neural networks with the ability of analysing and learning the temporal dependencies among the data. By storing and retrieving information from the input data, LSTMs can learn between time steps of arbitrary length to accomplish complex tasks such as classifying and predicting time series. It is widely used in various applications, such as handwriting generation, image captioning, question answering, video to text conversion, machine translation, and so on. In [7], an LSTM network-based processing method is designed for forecasting multivariate time sequence data, and an autoencoder method is combined with an SVM (Support Vector Machine) algorithm for detecting the anomalies in trades.

Due to the above capabilities, LSTM networks are widely used for classification and identification for time series data. Although the LSTM network has achieved great success on many kinds of classification problems, its performance is limited by parameter selection. It is difficult to manually adjust and select the hyperparameters of the LSTM neural network, and inappropriate parameter settings may lead to overfitting or slow convergence and even seriously reduce the classification accuracy.

Therefore, many scholars try to improve the LSTM network and apply it to different engineering problems. The utilization of LSTM networks in [8] enables the detection of complex anomaly and error types while considering both contextual and temporal characteristics. In [9], LSTM neural networks are combined with attention mechanism sentiment representation learning and detection of words and texts, where the attention mechanisms of LSTM architectures are based on diverse abstraction levels.

However, designing the optimal LSTM network architecture with suitable parameters is still a daunting task, not only requiring human supervision but also the searching and regulating process may lead to overly complex models, which may weaken the generalization ability of the network. The use of intelligent optimization technologies such as evolutionary computing to assist deep learning is an emerging technical means, which enables the system to adaptively adjust parameters according to the actual circumstances, and trade-off between accuracy and calculation volume, thereby improving the recognition accuracy and antiresistance ability of the LSTM networks.

This article combines the UIoT with the evolutionary LSTM networks and designs an improved intelligent optimization algorithm called EEC-DE (Exploration and Exploitation Control-Differential Evolution) to improve the network classification accuracy, while reducing its complexity. By implementing fast and low-burden edge computing on underwater sensor arrays and surface nodes, complex and time-consuming training and optimization processes can be moved to land or ships, the computing process is done using cloud computing or fog computing, and the entire system can be easily deployed on the existing hardware platform, the accuracy and reliability are higher, and it is more practical.

The rest of this article is organized as follows: first, close works that are related to evolutionary algorithm-based classification and intelligent recognition methods are reviewed. Then, the novel EEC-DE algorithm is elucidated in detail. After that, the extensive experiments are conducted to verify the effectiveness of the proposed method. Finally, conclusion of this article and some future directions are presented in the last section.

## 2. Related Work

Comparing with the traditional target recognition method that combines artificial feature selection and pattern recognition, the neural network, the automatic feature extraction inherent in convolutional neural networks, makes it the most promising method, and the evolutionary computing is a prospective approach to improve the performance of neural networks on difficult and complicated tasks. Since this article mainly focuses on designing networks for marine target recognition, this section only reviews closely related works on evolutionary algorithm-based evolving neural networks.

Since the LSTM networks can be used for time-series classification and regression, the marine target identification and recognition problem studied in this paper belong to the sequence-to-label classification. An LSTM layer could acquire the long-term dependencies between time steps in time series or sequence data, but it is sensitive to the details and characteristics of the input data. To alleviate this problem, many studies on methods of evolutionary classes were born. In [10], a two-stage algorithm is designed, where the first stage is designed to get the best performing LSTM structure automatically. During the evolution process, the connection weight inheritance method is used to improve the efficiency,

and the second stage designs the ensemble system by choosing a suitable LSTM. An ensemble evolutionary deep network model is proposed in [11], including a convolutional neural network (CNN) and a bidirectional long short-term memory (BLSTM) network for human action recognition. A guided general search process is incorporated into the algorithm to overcome stagnation caused by insufficient search capacity.

Due to the significant impact of hyperparameters in LSTMs on classification performance and the complexity of the interactions between parameters, manual selection of these parameters requires specific prior knowledge, and traditional trial and error tuning procedures often only find suboptimal solutions. The search for optimal parameters is an NP-hard problem, which can be solved by evolutionary computation [12]. In order to make the LSTM network achieve better performance when dealing with difficult classification problems, more innovative and efficient evolutionary computing methods need to be devised [13]. This paper designs a novel differential evolution-class algorithm, which can balance between the explorations and exploitations. The correlation learned by LSTM networks will be enhanced; thus, the usability of the proposed marine target recognition method could be improved.

## 3. Proposed Algorithm

To improve network performance and reduce reliance on expert experience in parameter setting, this article proposed the framework in the case where the network depth is specified as a fixed value; hyperparameters within a preset search range will be optimized automatically and simultaneously according to the objective function.

*3.1. Evolutionary Processing.* The whole evolutionary process includes four main steps, namely initialization of population, fitness evaluation on the LSTM networks, offspring generation, and exploration and exploitation control.

First, the population is initialized with the given population parameters as following: population size $N_p$, the dimension size $L$, which is related to the number of hyperparameters to be optimized. The individual in the population can be described as a vector $x_p^g = [x_{p1}^g, x_{p2}^g, \ldots x_{pL}^g]$, where the subscript $g$ indicates the iteration at the present. In the initialization process, every dimension of each individual is uniformly sampled from zero to one. Every individual map a candidate choice of the value of hyperparameters, and it is a candidate solution of the optimization problem at the meanwhile.

During the evolutionary process, every individual will update itself under the guidance of evolutionary formula. The main steps in the offspring generation of DE algorithm are mutation and crossover. The researchers have developed a variety of mutation operators to achieve the mutation process such as [14, 15]. Among all of them, the mutation operators shown in Table 1 are used most commonly.

Here, the indicator $r_1$ to $r_5$ are all exclusive integers randomly sampled from the range $[1, N_p]$, $x_b^g$ indicates the best individual in the present generation $g$, and parameter $F$

TABLE 1: Some commonly used mutation operators.

| Type | Mutation formula |
|---|---|
| DE-rand-1 | $v_p^g = x_{r1}^g + F(x_{r2}^g - x_{r3}^g)$ |
| DE-rand-2 | $v_p^g = x_{r1}^g + F(x_{r2}^g - x_{r3}^g + x_{r4}^g - x_{r5}^g)$ |
| DE-best-1 | $v_p^g = x_b^g + F(x_{r1}^g - x_{r2}^g)$ |
| DE-current to rand-1 | $v_p^g = x_p^g + F(x_p^g - x_{r1}^g)$ |

is the scale factor that has a huge impact on the mutation process, which influence the solution of the algorithm eventually. However, setting the scale factor as a constant cannot balance between exploration and exploitation. Many researchers believe that the scale factor should change during the iteration process or adjust adaptively.

Since the kernel density estimation method can be used to reckon the probability density function of any random variables in a nonparametric manner, it motivates us to design a new scheme called EEC to balance the exploration and exploitation by estimating the state of individuals instead of change the scale factor or the mutation operator.

Except for the first generation after initialization, the newly generated offspring $x_p^g$ needs to perform fitness calculation $f(x_p^g)$ in each iteration. The fitness calculation could be obtained according to the optimized objective function, which will be introduced in the next subsection specifically.

*3.2. Objective Function Design.* When calculating the fitness value of each offspring during the optimization, the network embedded in the algorithm needs to be trained and verified every time, and the recognition accuracy could be obtained after the pair of training and testing operations are done. In order to complete the classification task, the network is required not only to correctly cluster the data but also to ensure that the data match its corresponding labels.

During training, testing, and validating, the data first enter a bidirectional LSTM layer with $n$ hidden nodes, and the last element of the sequence will be taken, in which the activation function uses the hyperbolic tangent function to realize automatic mining and learning of data internal dependencies through forgetting, input, and output gates. Additive interactions are applied at each layer, which helps improve gradient flow over long sequences during training and testing process [16]. This prompts the bidirectional LSTM layer to map the input sequences to $n$ different features and then generate the output for the fully connected layer. Finally, recognition is accomplished by a fully connected layer, a softmax layer, and a classification layer.

On the premise of obtaining the highest possible recognition accuracy, obtaining a sensible and feasible parameter setting method is a multiobjective problem, where the Lagrange multiplier method is used to design the objective function, which is designed as a sum of three terms as follows:

$$f\left(x_p^g\right) = A\left(x_p^g\right) - x_{p1}^g \times c_1 + x_{p2}^g \times c_2, \quad (1)$$

where $x_{p1}^g$ is the number of hidden nodes in the bidirectional LSTM layer, $x_{p2}^g$ is the batch size, which represents the number of signals that the network learn each time, and

$A(x_p^g)$ represents the percentage accuracy of the LSTM neural network trained with the hyperparameters corresponding to the current individual on the validation set, where $c_1$ and $c_2$ are constants, which represent the importance of the corresponding quantity in the objective function. This Lagrange multiplier style objective function engages the algorithm to find the stationary point automatically in an iterative way.

In the offline training stage, the algorithm can adjust and optimize the hyperparameters of the network adaptively by solving this constrained optimization problem, and it can obtain a higher recognition rate with less computational effort in practical applications.

### 3.3. Exploration and Exploitation Control.
In order to determine whether to explore or exploit, a variation of kernel density estimation is designed to check the state of the population. Since the individual distribution in DE algorithms tends to fit the Gaussian mixture model when the number of individuals is large enough, the Gaussian kernel function is used to design the decision condition. The discrimination flags are as follows:

$$\rho^g = \frac{1}{N_p} \sum_{p=1}^{N_p} f(x_p^g) \kappa(x_b^g - x_p^g). \tag{2}$$

It can be interpreted as the sum of the objective function values weighted by the corresponding standard Gaussian kernel [17]:

$$\kappa(x) = \frac{1}{\sqrt{2\pi}} e^{-\left(\|x\|^2/2\right)}. \tag{3}$$

Then, compare it with the optimal fitness value weighted by a constant value. If $\rho^g \geq \eta f(x_b^g)$, it means that the algorithm has not fully converged and needs to continue to evolve according to the exploration mode. The multiplier parameter $\eta$ is called the adjustment factor that can adjust the threshold on the right side of the inequality sign. On the contrary, if the value of $\rho^g$ is smaller, it means that the individuals in the algorithm are close enough to the global optimal point; then it will switch to the exploitation mode, search for the optimization near the optimal value, and get a higher precision solution. Simultaneously, a pool of alternative strategies will be created, which contains two alternative evolution operators; the evolution formula of exploration mode is as follows:

$$v_p^g = x_p^g + F\left(x_{r1}^g - x_{r2}^g + x_{r3}^g - x_{r4}^g\right). \tag{4}$$

This means that four individuals are randomly selected from the population, and the difference operation is performed with each other. As for the exploitation mode, the mutation formula is as follows:

$$v_p^g = x_p^g + F\left(x_{r1}^g - x_{r2}^g\right) + \mu^{(-g/G)}, \tag{5}$$

where $\mu$ is the zoom factor with the same dimension as the individual, the last term in the formula will be exponentially decayed with the number of iterations grows.

When the mutation is completed, the algorithm enters the crossover step. Calculate the fitness of the mutated individuals, and then apply the greedy selection strategy to crossover the individuals in the population with the mutated offspring:

$$x_p^{g+1} = \begin{cases} v_p^g, & f(v_p^g) > f(x_p^g), \\ x_p^g, & f(v_p^g) \leq f(x_p^g). \end{cases} \tag{6}$$

Then all the populations are merged to generate the next generation population. As the number of iterations increases, the population in exploration mode will implement global search constantly, while in exploitation mode, the algorithm improves the accuracy of the algorithm around the optimal solution. Until the maximum number of iterations is reached, the optimal individual is linearly mapped to the parameter space, so the output will be the hyperparameter optimization result.

Since there is no fixed or closed form functional relationship between neural network recognition accuracy and hyperparameters, but rather a complex and random correspondence, and the objective function is both multidimensional and multimodal. The proposed EEC-DE algorithm is especially suitable for evolutionary neural networks.

### 3.4. Overall Framework.
The overall architecture of the underwater Internet of Things can be categorized by geographic location and application layer. To identify underwater targets, during the offline training phase, the simulated signals evolve the LSTM networks on the server located in the seashore computing centre or complicated by cloud computing, and the output optimal hyperparameters are transferred to the edge devices. When the system is powered on and working, underwater magnetic signals are collected by magnetic sensors deployed on the seabed or AUVs and then transmitted to the sink nodes or surface platforms by underwater cables or underwater acoustic communication [18], where the recognition by the evolved network is accomplished and finally sent to the ground control centre via satellite or other RF links for subsequent decisions.

According to the above content, the overall framework of intelligent marine target recognition and specific process of evolving the LSTM network with EEC-DE algorithm is as follows, and the schematic diagram is shown in Figure 1:

Step 1: preprocess the data, including feature extraction, shuffling and normalization, to obtain training set, validation set, and test set.

Step 2: randomly generate an initial population, each individual representing a candidate hyperparameter setting method of LSTM.

Step 3: decode the individuals and train the corresponding LSTM network on the training set and then use the validation set to evaluate the trained network to obtain the fitness value for each offspring individual.
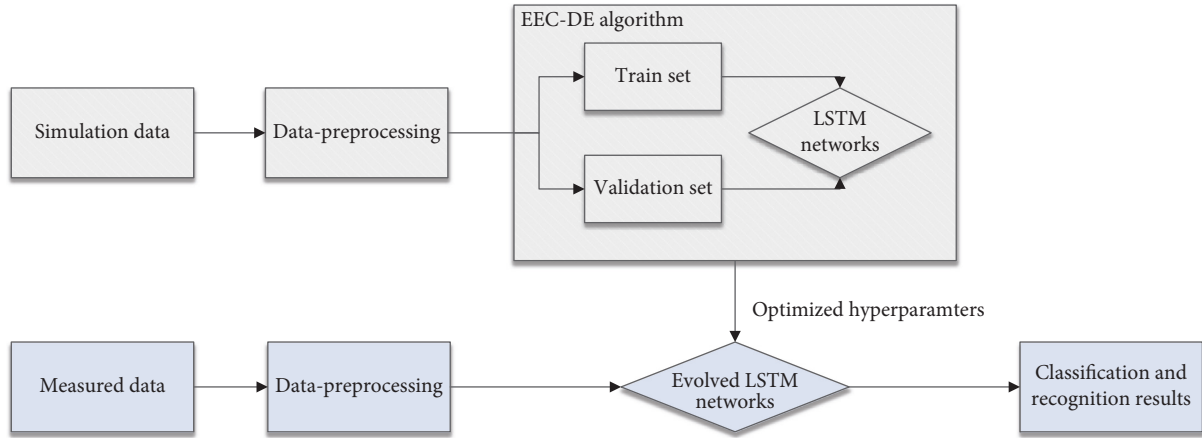
FIGURE 1: The overall framework of intelligent marine target recognition.

Step 4: apply the judgment procedure by the EEC mechanism and decide whether to explore or exploit according to the current state of the algorithm.

Step 5: select the corresponding operator for mutation and crossover. Complete the update for each individual in the population.

Step 6: if the maximum number of iterations is reached, terminate the iteration loop and output the optimal solution in the last generation, otherwise, go to Step 3.

Step 7: train the LSTM network using the training set and validation set and finally evaluate the evolved LSTM network on the test set or measured data and output the recognition results.

## 4. Experiments

In order to fully verify the performance of the design scheme, this section will expand from the following aspects. First, the performance analysis of the EEC-DE algorithm will be reflected in the standard benchmark function. Then, the results of the UIoT-based evolutionary LSTM network and several comparison methods will be given based on the simulated data.

The widely used $CEC'2013$ standard set of benchmark functions is adopted to evaluate the performance of the EEC-DE algorithm. Four of them are chosen to evaluate the search ability of the algorithm, namely Ackley's function, Beale's function, Rastrigin's function, and Levi's function. All of these benchmark functions are transferred into maximization optimization problems. The differential evolution (DE), the Genetic algorithm (GA), the particle swarm optimization (PSO), and the Imperialist Competitive Algorithm (ICA) are deployed as competitors. In order to compete fairly, the overall population size $N_p$ of all algorithms is set to 60, and the iteration termination criteria $G$ is 60 [19]. For EEC-DE, $F$ is a random number in $[0.5, 1]$, $\eta = 0.95$, $\mu = 100$. As for other comparison algorithms, parameter setting refer to [20–22]. All the experiments were run independently for 1000 times. The best score obtained so far is defined as the difference between the objective function

value of the best individual in the current iteration and the global optimal value. The average convergence curve is shown in Figures 2–5, and the average running time (s) is shown in Table 2.

It can be concluded from the results that the proposed EEC-DE algorithm outperforms all the other algorithms, especially when dealing with multimodal problems such as Rastrigin's function and Ackley's function; it has achieved superior results, and other algorithms may get stacked into the local optima. This superiority comes from the EEC schema, which enables the algorithm to switch between exploration and exploitation modes flexibly. By combining the advantages of several mutation operators, the algorithm is universally applicable to complex problems.

By comparing the average computing time, the proposed EEC-DE algorithm does not show the advantage in speed when dealing with benchmark functions. However, for the evolving networks problem we focus on, the searching ability of the algorithm is the most important because most of the computation is generated by the training and validation process of the networks in the evolution process.

After testing on several classical benchmark functions, the powerful and efficient searchability of the EEC-DE algorithm is proved. Now it will be applied to evolving LSTM networks to realize the identification based on the UIoT.

We compare the performance of the EEC-DE-based evolving LSTM networks with other methods. The recognition methods used in the experiments are as follows: Evolutionary LSTM network based on DE, PSO, GA, and ICA. Each experiment employs the following settings to ensure fair comparisons: $c_1 = 0.01$, $c_2 = 0.001$; these multiplier parameters are given small values, so that the recognition accuracy is established as the main component of the objective function value, and the number of hidden nodes is limited, and the batch size is prevented from being too small. Population size is 60, maximum number of iterations is 100, a total of 20 epochs are used in the training phase to allow the network to make 20 passes through the data [23], and an ADAM solver is applied to the training, validating and testing process, The search range of each hyperparameter is shown in Table 3.
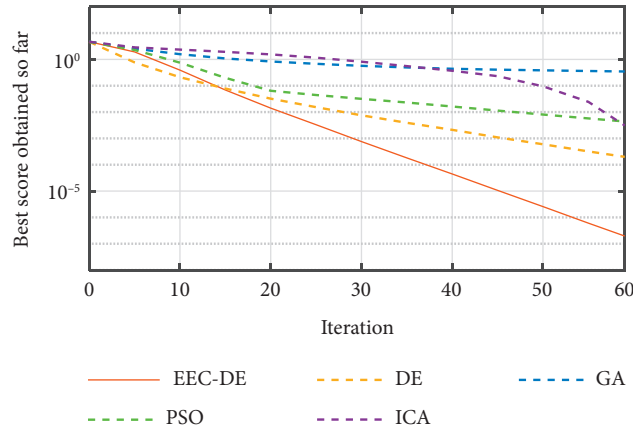
FIGURE 2: Comparison of convergence curves obtained in Ackley's function.
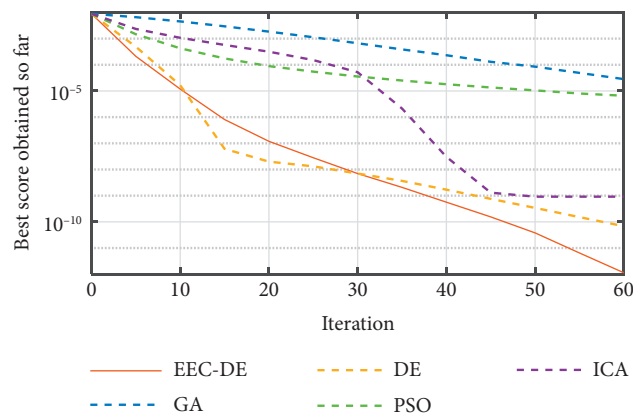


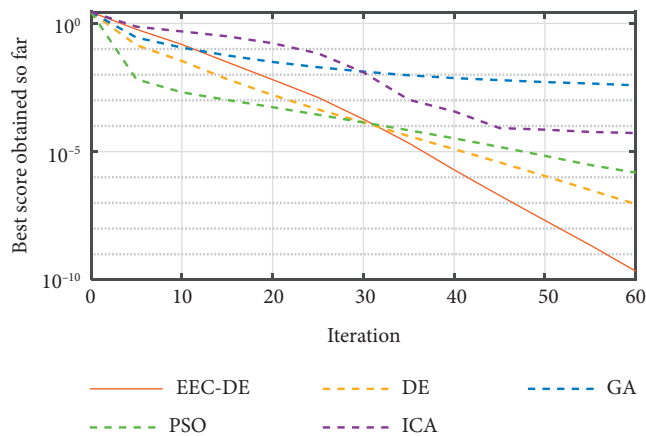FIGURE 3: Comparison of convergence curves obtained in Beale's function.



FIGURE 4: Comparison of convergence curves obtained in Rastrigin's function.

The box plot of the recognition accuracy is shown in Figure 6. On each box, the result comes from 10 independent runs, and the central red mark indicates the median of the accuracy, and the top and bottom edges of the box represent the 75th and 25th percentiles of the accuracy respectively.

The specific statistics of the results are shown in Table 4. We can draw the conclusion that the proposed EEC-DE-based evolutionary LSTM outperforms other competitors by comparing the accuracy, especially based on the comparison of the average accuracy, which indicates that a more suitable
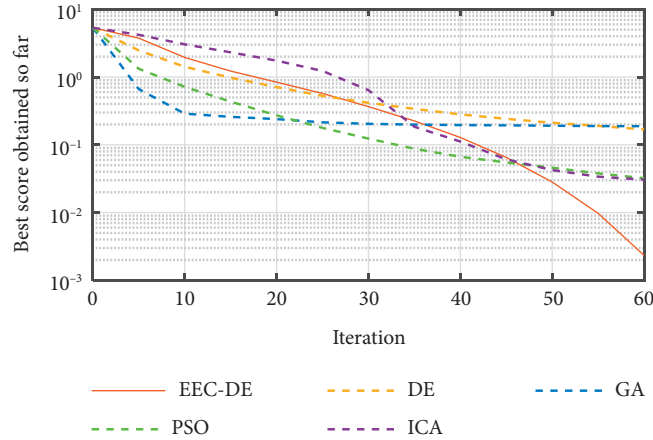
FIGURE 5: Comparison of convergence curves obtained in Levi's function.

TABLE 2: Average running time (s).

| Function | EEC-DE | DE | GA | PSO | ICA |
|---|---|---|---|---|---|
| Ackley's function | 1.036E-01 | 1.011E-01 | 4.532E-02 | 1.835E-02 | 1.047E-01 |
| Beale's function | 9.954E-02 | 9.876E-02 | 4.347E-02 | 1.969E-02 | 4.465E-02 |
| Rastrigin's function | 1.027E-01 | 9.965E-02 | 4.177E-02 | 2.029E-02 | 6.348E-02 |
| Levi's function | 1.032E-01 | 1.001E-01 | 4.189E-02 | 1.922E-02 | 6.200E-02 |

TABLE 3: The search range of the hyperparameters.

| Optimized hyperparameters | Range |
|---|---|
| Number of hidden nodes | [10, 100] |
| Batch size | [10, 1000] |
| Learning rate | [0.01, 0.5] |

TABLE 5: The mean configurations of the optimized LSTM networks.

| Configurations | EEC-DE | DE | GA | PSO | ICA |
|---|---|---|---|---|---|
| Hidden nodes | 35 | 42 | 28 | 29 | 16 |
| Batch size | 115 | 129 | 342 | 155 | 255 |
| Learn rate | 0.195 | 0.016 | 0.083 | 0.177 | 0.109 |

hyperparameter setting is obtained by the optimization process, which also shows that the searchability of the algorithm has been enhanced by the EEC scheme from another perspective. Furthermore, it can be concluded from the variance that the proposed method is more reliable.

The average hyperparameters obtained from 10 independent runs are shown in Table 5. It is worth mentioning that while ensuring high classification accuracy, the proposed evolutionary LSTM network based on EEC-DE choose a more suitable way that enables the LSTM networks to analyse time-series data more efficiently and avoid overfitting [24], while other methods are more prone to falling into local optima and leading to poor results.



FIGURE 6: Box plot of the recognition accuracy.

## 5. Conclusions

In recent times, the inefficiency and the long response time of current UIoT systems, as well as the limited computing power of the underwater embedded device, still represent key issues that prevents UIoT devices and architectures from being an appreciable and effective solution of the underwater marine target recognition and intrusion detection. This article proposes an UIoT-based intelligent marine target recognition method, in order to take advantage of the combination of edge computing and cloud computing. An

TABLE 4: The statistics of the recognition result.

| Statistics | EEC-DE | DE | GA | PSO | ICA |
|---|---|---|---|---|---|
| Average | 97.4 | 95.8 | 95.8 | 94.6 | 96.4 |
| Maximum | 98.6 | 96.9 | 97.6 | 95.8 | 98.5 |
| Minimum | 96.7 | 94.2 | 93.5 | 92.5 | 94.3 |
| Median | 97.3 | 95.9 | 95.8 | 94.7 | 96.5 |
| Variance | 0.36 | 0.65 | 1.53 | 0.94 | 1.18 |

UIoT system consisting of underwater nodes, surface ships, and seashore-based control centre is designed, and all of these components are connected through the network to perform identification and recognition tasks more efficiently. And an EEC-DE algorithm is proposed for evolving the hyperparameters automatically in LSTM networks.

The recognition accuracy of the low-power classifiers around the signal acquisition devices is improved, and the complex optimization process which consisting of alternating training and validating will be transferred to the high-performance computing device located on the land, so the advantages of cloud computing and the instantaneity of edge computing are combined. The improvement of recognition ability benefits from the effective searching algorithm. In contrast to the existing differential class algorithms or evolutionary methods, the proposed EEC-DE algorithm could switch between the exploration and exploitation search in an adaptive and flexible manner, and the algorithm can converge to the global optimum and escape from local optimum.

The experimental results show that the method proposed in this article can adjust the hyperparameters of the LSTM networks dynamically, avoid underfitting, or overfitting problems, so as to obtain better performance than the default LSTM networks or other baseline methods, and improve the recognition accuracy while reducing the computational burden of the devices.

For future research, we will attempt to design more efficient UIoT schemes to achieve better performance and enable them to widely applied in ocean activities. We will also design novel control and search mechanisms to better achieve the balance between the local search and global convergence for evolving neural networks, which have been demonstrated to be powerful tools for solving a large amount of time series analysis problems such as pattern recognition, time-series forecasting and sequence regression.

## Data Availability

The figures and tables used to support the findings of this study are included in the article.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

## References

[1] C. Hu, Y. Pu, F. Yang, R. Zhao, A. Alrawais, and T. Xiang, "Secure and efficient data collection and storage of IoT in smart ocean," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9980–9994, 2020.

[2] T. Qiu, Z. Zhao, T. Zhang, C. Chen, and C. P. Chen, "Underwater Internet of Things in smart ocean: system architecture and open issues," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4297–4307, 2019.

[3] C. Wang, J. Lu, X. Ding, C. Jiang, J. Yang, and J. Shen, "Design, modeling, control, and experiments for a fish-robot-based IoT platform to enable smart ocean," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9317–9329, 2021.

[4] Q. Wang, H. N. Dai, Q. Wang, M. K. Shukla, W. Zhang, and C. G. Soares, "On connectivity of uav-assisted data acquisition for underwater internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5371–5385, 2020.

[5] J. Zhang, J. Sha, G. Han, J. Liu, and Y. Qian, "A cooperative-control-based underwater target escorting mechanism with multiple autonomous underwater vehicles for underwater internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4403–4416, 2021.

[6] N. Javaid, S. Cheema, M. Akbar, N. Alrajeh, M. S. Alabed, and N. Guizani, "Balanced energy consumption based adaptive routing for IoT enabling underwater WSNs," *IEEE Access*, vol. 5, pp. 10040–10051, 2017.

[7] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management," *International Journal of Information Management*, vol. 57, Article ID 102282, 2021.

[8] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Computers in Industry*, vol. 131, Article ID 103498, 2021.

[9] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang, and S. Qiao, "Attention-emotion-enhanced convolutional LSTM for sentiment analysis," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021.

[10] R. A. Viswambaran, G. Chen, B. Xue, and M. Nekooei, "Two-stage Genetic algorithm for designing long short term memory (LSTM) ensembles," in *Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC)*, pp. 942–949, IEEE, Kraków, Poland, July 2021.

[11] L. Zhang, C. P. Lim, and Y. Yu, "Intelligent human action recognition using an ensemble model of evolving deep networks with swarm-based optimization," *Knowledge-Based Systems*, vol. 220, Article ID 106918, 2021.

[12] H. Gao, Y. Su, S. Zhang, Y. Hou, and M. Jo, "Joint antenna selection and power allocation for secure co-time co-frequency full-duplex massive MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 655–665, 2021.

[13] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, 2019.

[14] W. Deng, S. Shang, X. Cai, H. Zhao, Y. Song, and J. Xu, "An improved differential evolution algorithm and its application in optimization problem," *Soft Computing*, vol. 25, no. 7, pp. 5277–5298, 2021.

[15] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and H. Faris, "MTDE: an effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems," *Applied Soft Computing*, vol. 97, Article ID 106761, 2020.

[16] R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, and F. Li, "Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal

clustering," *Expert Systems with Applications*, vol. 169, Article ID 114513, 2021.

[17] F. Kamalov, "Kernel density estimation based sampling for imbalanced class distribution," *Information Sciences*, vol. 512, pp. 1192–1201, 2020.

[18] M. Khosravy, N. Gupta, N. Dey, and P. M. Ger, "Smart green ocean underwater IoT network by ICA-based acoustic blind MIMO OFDM transceiver," *Earth Science Informatics*, vol. 14, no. 2, pp. 1073–1081, 2021.

[19] H. Gao, Y. Su, S. Zhang, and M. Diao, "Antenna selection and power allocation design for 5G massive MIMO uplink networks," *China Communications*, vol. 16, no. 4, pp. 1–15, 2019.

[20] V. Kachitvichyanukul, "Comparison of three evolutionary algorithms: GA, PSO, and DE," *Industrial Engineering and Management Systems*, vol. 11, no. 3, pp. 215–223, 2012.

[21] A. M. Manasrah and H. Ba Ali, "Workflow scheduling using hybrid GA-PSO algorithm in cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1934784, 16 pages, 2018.

[22] S. Hosseini and A. Al Khaled, "A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research," *Applied Soft Computing*, vol. 24, pp. 1078–1094, 2014.

[23] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[24] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, 2019.