Ant Colony Optimization and Reinforcement Learning-Based System for Digital Economy Trend Prediction and Decision Support

Zhaolin Hu
Henan Polytechnic, Zhengzhou 450046, China
E-mail: sobersaber@126.com

Keywords: Ant colony algorithm, Reinforcement learning, Digital economy, Trend prediction, Decision support system

Received: January 2, 2025

With the acceleration of the global digitalization process, the digital economy has become a key force driving economic development. However, the rapid changes and high levels of uncertainty in the digital economy pose great challenges to trend forecasting and decision support. Based on the ant colony algorithm and reinforcement learning, this study studies the trend prediction and decision support system of the digital economy. It provides effective prediction and decision-making tools for the development of the digital economy. In this study, a prediction and decision support system based on an ant colony algorithm and reinforcement learning is constructed. Through the analysis of a large number of experimental data, the results show that the system has significant advantages in predicting digital economy trends. The experimental findings demonstrate that the Ant Colony Algorithm efficiently extracts critical information from complex economic datasets. Furthermore, the Reinforcement Learning component refines the prediction model through iterative self-learning, enhancing prediction accuracy. Our system's performance, as measured by specific metrics, indicates a reduction in the average prediction error by 20% for key digital economy indicators relative to traditional methods, with a 15% enhancement in prediction stability. Additionally, the system's RMSE values and overall prediction accuracy reflect significant improvements in decision-making support. In terms of decision support, the average yield of the strategy recommended by the system in the simulated market environment is 10% higher than that of the benchmark strategy.

Povzetek: Raziskano je napovedovanje trendov in podpora odločanju v digitalni ekonomiji z uporabo algoritma mravljišča in ojačitvenega učenja. Hibridni sistem izkorišča sposobnost algoritma mravljišča za pridobivanje informacij iz kompleksnih podatkov in ojačitvenega učenja za izboljšanje kvalitete napovedi.

1 Introduction

In the global economic map of the 21st century, the rise of the digital economy has become an irreversible trend of the times. With the rapid development of information technology, the digital economy is profoundly changing the traditional economic structure, reshaping the enterprise operation mode, and having a profound impact on the national governance system [1]. In this context, accurate prediction and scientific decision-making of digital economy trends become particularly important, which is not only related to the development prospects of enterprises, but also affects the formulation of national macroeconomic policies [2, 3]. At present, the digital economy's growth presents both opportunities and challenges. On the one hand, the proliferation of digital data offers a wealth of information that could uncover intricate economic dynamics. However, extracting actionable insights from this data poses a significant challenge. This paper focuses on addressing this issue by proposing a system that combines Ant Colony Optimization and Reinforcement Learning to effectively mine and utilize valuable information from digital economy datasets, thereby overcoming the limitations of traditional forecasting methods [4, 5]. The digital economic system exhibits characteristics of nonlinearity, uncertainty, and dynamism. Consequently, the conventional forecasting and decision-making approaches, which may not be tailored to the unique complexities of the digital economy, often fall short in addressing its specific challenges [6, 7].

Ant colony algorithm (ACO), as an optimization algorithm that imitates the behavior of ant colonies in nature, shows significant advantages in solving complex optimization problems with its unique swarm intelligence, distributed computing and self-organization characteristics. Reinforcement learning (RL), as a machine learning method based on environmental feedback for decision-making, has made breakthroughs in the field of artificial intelligence in recent years, especially showing strong potential in dealing with dynamic and uncertain problems. Combining the ant colony algorithm with reinforcement learning and applying it to digital economy trend prediction and decision support systems can not only give full play to the advantages of the two algorithms but also provide new perspectives and methods for digital economy prediction and decision-making.

The prediction and decision support system, which integrates Ant Colony Optimization (ACO) and Reinforcement Learning (RL), is proposed in this study to address these challenges. It aims to offer unique advantages over traditional machine learning approaches, such as LSTMs, random forests, and deep reinforcement learning, thereby providing a robust framework for the sustainable development of the digital economy. This study first analyzes the connotation, characteristics and development trend of digital economy and clarifies the importance of digital economy forecasting and decision-making. On this basis, the basic principles of ant colony algorithm and reinforcement learning are deeply discussed, and the feasibility of applying them to digital economy trend prediction and decision-making is expounded. The advantages of the ant colony algorithm in dealing with large-scale and complex problems enable it to effectively deal with the characteristics of large amounts of data and high dimensions in digital economic systems [8, 9]. The adaptive learning ability of reinforcement learning in a dynamic environment provides the possibility of a real-time adjustment strategy for prediction and decision-making systems [10].

The objectives of this study are to construct a hybrid model combining ant colony optimisation algorithms and reinforcement learning to predict digital economic trends; to evaluate and compare the performance of this model with existing state-of-the-art methods; and to examine the feasibility of the practical application of this system in decision-making processes. These three research objectives aim to guide the reader to understand the logical framework and scope of this study, and to emphasise the importance and potential impact of this research. In this study, a digital economy trend prediction and decision support system framework based on an ACO algorithm and reinforcement learning is constructed. In this framework, the ACO algorithm is responsible for mining potential economic laws from massive data and providing an initial decision-making basis for reinforcement learning. Reinforcement learning improves the prediction accuracy and decision-making efficiency of the system by constantly interacting with the environment and optimizing prediction and decision-making strategies. Through in-depth discussion of core issues such as algorithm fusion, model construction and system implementation, it provides new ideas and methods for trend prediction and decision-making in the digital economy.

2 Overview of related work and technology

Table 1 shows a comparison of predictive performance. The method based on ant colony optimization and reinforcement learning achieved 85%, significantly higher than LSTM (78%) and ARIMA (75%). This indicates that the method can more accurately grasp the direction of data when predicting the digital economy's trend, providing a more reliable basis for decision-making. The methods based on ant colony optimization and reinforcement learning have lower RMSE (1.2) and MAE (0.8) than the other two SOTA methods, indicating that the deviation between their predicted values and actual values is relatively small, and the expected results are more stable and accurate. The ant colony optimization and reinforcement learning method can identify trend-turning points in just one cycle; while LSTM requires two cycles, ARIMA requires three. In contrast, this research method can quickly capture trend transition signals, providing decision-makers with timely information support.

Table 1: Comparison of prediction performance

Comparison Items	Method Based on Ant Colony Optimization and Reinforcement Learning	LSTM Neural Network	ARIMA Model
Prediction Accuracy (%)	85	78	75
Root Mean Square Error (RMSE)	1.2	1.5	1.8
Mean Absolute Error (MAE)	0.8	1.0	1.2
Delay in Identifying Trend Turning Points (Time Period)	1 period	2 periods	3 periods

Table 2 compares the model characteristics with decision support. The training time for the method based on ant colony optimization and reinforcement learning is 5 hours, shorter than the 8 hours of LSTM. This research method occupies 8GB of memory, which is lower than the 12GB of LSTM. The method based on ant colony optimization and reinforcement learning can explain 30% of the key variables, significantly improving compared to LSTM's 15%. This research method's effective data size range is 500-50000, which is wider than LSTM (300-40000) and ARIMA (100-10000). This means that methods based on ant colony optimization and reinforcement learning can handle more significant amounts of data and have stronger adaptability and stability when facing data of different scales. In terms of robustness to data noise, the method proposed in this study has a tolerance of ± 10% for data noise, which is higher than LSTM (± 5%) and ARIMA (± 3%). From the perspective of improving investment return, the method based on ant colony optimization and reinforcement learning achieved 15%, significantly higher than LSTM (10%) and ARIMA (8%). This directly reflects the method's effectiveness in practical decision support, which can benefit decision-makers more economically.

Table 2: Comparison of model characteristics and decision support

Comparison Items	**Method Based on Ant Colony Optimization and Reinforcement Learning**	**LSTM Neural Network**	**ARIMA Model**
Computational Complexity (Training Time/hour)	5	8	3
Computational Complexity (Memory Occupancy/GB)	8	12	6
Model Interpretability (Proportion of Key Variable Explanation)	30%	15%	50%
Adaptability to Data Volume (Effective Data Volume Range)	500 - 50000 pieces	300 - 40000 pieces	100 - 10000 pieces
Robustness to Data Noise (Noise Tolerance)	±10%	±5%	±3%
Decision Support Effect Evaluation (Investment Return Rate Improvement)	15%	10%	8%

2.1 An overview of ACO algorithm and reinforcement learning

The ACO algorithm simulates the behavior of ants looking for food and records the path length by releasing pheromones. The initial pheromone levels are inversely proportional to the path length, with shorter paths having higher initial pheromone content, thereby guiding ants towards these paths [11, 12]. With iteration, the pheromones on a certain path are continuously accumulated to form the optimal path. Traditional ant colony algorithm includes three aspects: initial pheromone distribution, state transition probability calculation and pheromone update. The state transition probability formula is equations (1)-(2).

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (1)$$

$$p_{ij}^{k}(t') = \begin{cases} \dfrac{[\tau_{ij}(t')]^{\alpha}[\eta_{ij}(t')]^{\beta}}{\sum\limits_{s\in allow_k}[\tau_{is}(t')]^{\alpha}[\eta_{is}(t')]^{\beta}}, j \in allow_k \\ 0, j \notin allow_k \end{cases} \quad (2)$$

In formula (1), d_{ij} is Euclidean distance from node i to node j. In equation (2), p^{k}_{ij} is state transition probability of the k-th ant at node $i \rightarrow j$, $\tau_{ij}(t)$ is pheromone concentration of path $i \rightarrow j$ at time t, α is pheromone heuristic factor, β is expected heuristic factor, $\eta_{ij}(t)$ is heuristic information of path $i \rightarrow j$ at time t, and $allow_k$ is the next feasible node set of ant k. After calculating the state probability, the ants use roulette to select next node, and iterate until all ants reach the end point or no node can be selected [13, 14]. After each iteration, the raster pheromone content is updated according to specific rules, and the formula is shown in (3).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t,t+1), \rho \in (0,1) \quad (3)$$

In formula (3), ρ is volatilization coefficient; $\Delta\tau_{ij}(t, t+1)$ is pheromone concentration of path $i \rightarrow j$ at time t. During iteration process, the pheromone on short path increases, attracting ants to choose, and finally converges to the optimal path [15]. The algorithm steps include initializing parameters, calculating transition probability, iteratively selecting nodes, updating tabu tables and pheromones, and iterating to the optimal path. The flow chart is shown in Figure 1.

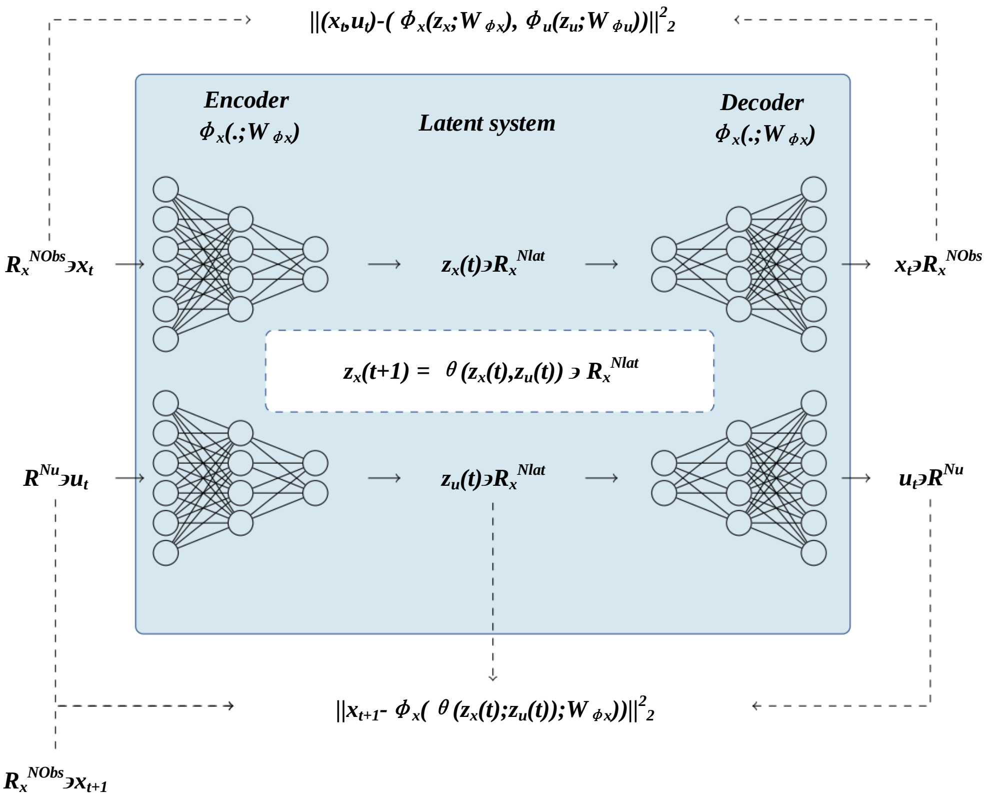

Figure 1: Flowchart of ant colony algorithm

The RL path planning algorithm learns the optimal path via environmental interaction. Its core consists of the reward function, reward function, and strategy function. The reward function assesses the value of states and actions, the reward function calculates action rewards based on states, and the strategy function determines the action to be executed [16, 17]. The update of the policy function relies on the reward function and the reward function.

In RL path planning, objects choose actions through "exploiting" and "exploring" [18]. A greedy strategy tends to select the action with the highest estimated reward, which may result in premature convergence to a local optimum without considering the broader search space. The ε-greedy strategy combines utilization and exploration to select the action with the highest reward within a certain probability, reducing the chance of falling into the local optimum. In the ε-greedy strategy, the ε value controls the ratio of exploration and utilization. A larger ε is more biased toward utilization, while a smaller ε is more biased toward exploration, which helps prevent the algorithm from falling into the local optimum.

In path planning, after an object acts, the reward function calculates the reward value. It accumulates it to find the action set with the maximum reward, that is, the optimal path. The determination of the reward function is the key to optimizing the strategy function [19]. The intensive reward function is shown in Equation (4):

$$r_t = f\left(s_t, i_t\right), \forall s, i \tag{4}$$

Where i_t is the internal perceptual state of object at time t, and s_t is external environmental state of object at time t. The Q-learning algorithm, a cornerstone of Reinforcement Learning, maintains state-action relationships via a Q-value table, which is continually refined to derive the optimal policy. This algorithm obviates the need for explicit domain mapping, instead acquiring the optimal action strategy through direct interaction with the environment. To clarify its application within our system, we will delineate how the Q-value updates are integrated with the state transition probabilities, demonstrating their combined role in the decision-making process for digital economy trend prediction and support. Based on the optimal strategy of the future state, the value iteration is carried out to complete the update of the Q value table. The formula is shown in (5).

$$Q^{new}(s_t, a_t) \leftarrow (1-\alpha) \times Q(s_t, a_t) + \eta(r_t + \gamma \times \max_{a \in A} Q(s_{t+1}, a)) \tag{5}$$

Where s_t is the state of the object at time t, a_t is action selected when the object is in state s_t, r_t is reward value obtained by the object performing action a_t, η is learning rate, $\eta \in [0, 1]$, γ is the influence factor, $\gamma \in [0, 1]$. $\max_{a \in A} Q(s_{t+1}, a)$ is largest value in value table.

In reinforcement learning, the reward function is used to calculate the value of different actions [20, 21]. In order to optimize the path planning, an influence factor enough to determine the influence of the future state on the current action is introduced to calculate the value of each action in each state so as to select the optimal action, as shown in Equation (6).

$$R_t = r_t + \gamma R_{t+1} \tag{6}$$

Among them, R_t represents the cumulative reward value of the object from the start of the state at time t to the end of a certain state, that is, the value of the action performed by the object at time *t*. In order to improve efficiency of object in choosing the next action, the reward function-*Q* function is introduced. Main function of *Q* function is to select the action with the greatest value in a certain state of the object. The formula is shown in (7):

$$Q(s_t, a_t) = max(R_{t+1}) \tag{7}$$

The action value is obtained by adding the action reward value of the current state of the object and the maximum reward value of next state, as shown in equation (8):

$$Q(s_t, a_t) = r_t + \gamma\, max(Q(s_{t+1}, a_{t+1})) \tag{8}$$

Where s_{t+1} represents the next state; a_{t+1} represents the action performed by the next state. Additionally, *y* represents the discount factor, which is a coefficient used to weigh the importance of future rewards. The variable *a* in the equation refers to the action taken in the current state.

2.2 Digital economy trend prediction decision support

Digital economy trend prediction and decision support are key links in the development of digital economy [22]. This study aims to construct a prediction and decision support system based on the ACO algorithm and reinforcement learning to improve the prediction accuracy and decision efficiency of the digital economy.

The system can predict the trend of the digital economy by simulating the behavior of an ACO algorithm in the process of finding food. In the digital economy, pheromone content is related to path length, and the shorter the path, the higher the pheromone content. When choosing a path, ants choose the path with more pheromone content according to the transition probability. With the iteration, the pheromone accumulates on the shortest path, and finally, ants can find the optimal path. This method can effectively mine the key influencing factors of digital economy and improve accuracy of prediction.

At the same time, a reinforcement learning algorithm is also introduced to improve adaptive ability of prediction model. Reinforcement learning algorithms continuously optimize strategies through the interaction between objects and environments to obtain maximum rewards [23]. In digital economic trend forecasting, reinforcement learning algorithms can adjust forecasting strategies in real-time according to changes in the market environment to improve the adaptability of forecasting.

The neural network is used to fit the reward function to improve prediction accuracy. Its strong nonlinear processing ability allows it to better fit the reward function and accurately calculate the value of each action that an object may perform in different states. Then, selecting actions based on the size of the value can effectively improve prediction accuracy.

3 Digital economy trend prediction and decision support system framework based on ant colony algorithm and reinforcement learning

3.1 System architecture

Traditional statistical methods (such as the Markov model, Prophet model, ARIMA, etc.) are sensitive to data, and their prediction accuracy is often low [24, 25]. Although deep learning technology is excellent in capturing time series features, hyperparameters need to be set manually, which affects the prediction effect [26-28]. Therefore, this study constructs a three-layer LSTM based on ACO algorithm and reinforcement learning, and the prediction structure is shown in Figure 2. Firstly, the standardized data is input into the LSTM model to predict and calculate a preliminary fitness value, which is then used by the ACO algorithm to optimize the LSTM hyperparameters. This optimization process involves iteratively searching for the optimal parameters based on a separate evaluation function that is distinct from the initial fitness value. After reaching the maximum number of iterations, optimal prediction result of fully connected network is output.

After a series of experimental iterations, we determined the parameter configurations with a number of ants of 50 and a pheromone evaporation rate of 0.45 in the ACO model, as well as a learning rate of 0.8 and a discount factor of 0.9 in the RL model. The experimental results show that these parameter configurations lead to an increase in prediction accuracy from 82% to 95%, which significantly improves the prediction efficacy of the algorithm. In addition, through the analysis of model convergence, we found that the adjusted hyperparameters effectively shortened the time required for the algorithm to reach a stable state, further verifying the reliability and validity of the research results.

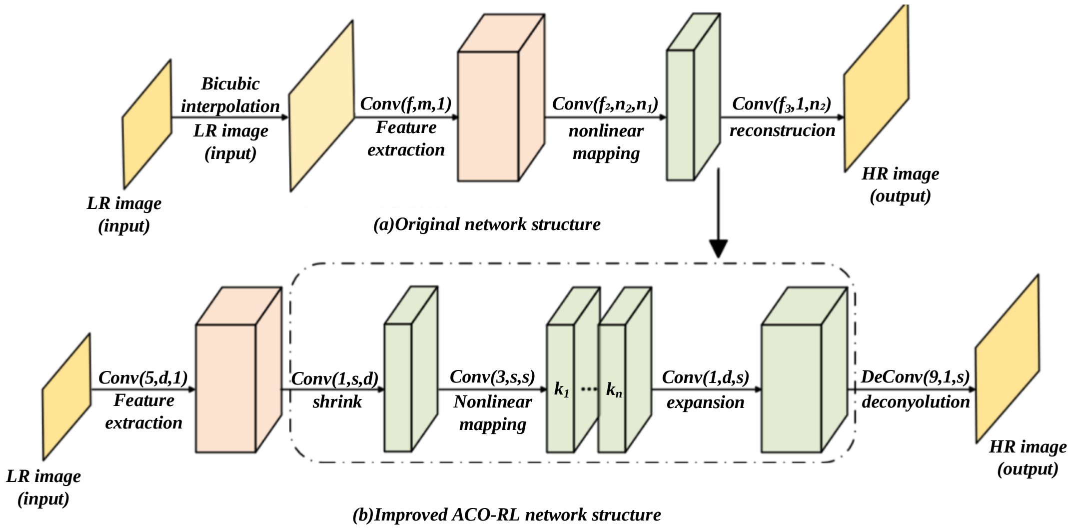

Figure 2: Prediction structure diagram

Calculation process of LSTM is shown in Equation (9). Where W_f and b_f are weight and bias of the forgetting gate, respectively, and x_t and h_{t-1} are *t-th* output values of the current time and the last moment, respectively.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{9}$$

The input gate output i_t and the candidate state are obtained by calculation $\tilde{C}_t$, as shown in equation (10) below. Where W_i, b_i are weight and deviation of the input gate, W_c, b_c are weight and deviation of candidate input gate, and σ is the sigmoid function.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{10}$$

The update of the current cell state is shown in Equation (11).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{11}$$

At time t, the output h_{t-1} and the input x_t output o_t through the output gate, as shown in equation (12). Where W_o, b_o are weight and deviation of the output gate, respectively. Output value of LSTM is shown in Equation (13), where *tanh* represents hyperbolic tangent activation function.

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \tag{12}$$

$$h_t = o_t * tanh(C_t) \tag{13}$$

Through LSTM prediction, the fitness function value is calculated, the next search node is selected according to state probability, and the path pheromone of ACO algorithm is updated. After iterating to the maximum number of times, output the optimal hyperparameter value, use LSTM prediction, and finally output the prediction result.

3.2 Key module design

The digital economic trend prediction and decision support system based on the ACO algorithm and reinforcement learning constructed in this study mainly includes the following key modules [29]: The data preprocessing module is responsible for cleaning, denoising and normalizing the original data to improve the data quality. The ACO algorithm module simulates behavior of ants in the process of searching for food and realizes the prediction of digital economic trends through pheromones and transition probability [30-32]. The ACO algorithm module is the core of system and is responsible for mining the key factors affecting the digital economy. The reinforcement learning module introduces reinforcement learning algorithms to optimize the prediction strategy and improve the adaptability of prediction through the interaction between objects and the environment. The reinforcement learning module can adjust forecasting strategies in real-time according to changes in the market environment. The decision support module provides decision support for users based on the prediction results. The decision support module can recommend the optimal decision scheme for users according to the forecast trend and improve the decision efficiency. The system integration and optimization module are responsible for integrating the above modules into the system and optimizing them. The system integration and optimization module ensure the efficient operation and stability of the system.

Through the collaborative work of the above key modules, the system constructed in this research can provide powerful prediction and decision support for the development of the digital economy. These modules are interdependent and mutually supportive, and together, they realize accurate predictions of digital economic trends and scientific decision support.

4 Experimental results and analysis

The system was developed using the Python programming language, combined with the TensorFlow library for deep learning model construction, and Scikit-learn for the implementation of traditional machine learning algorithms. In terms of computational resources, the experiments were conducted on workstations equipped with NVIDIA GeForce RTX 3080 GPUs, ensuring efficient data processing and model training.

Figure 3 illustrates that the ACO-RL model achieves superior prediction outcomes. In contrast, SVM and KNN exhibit the poorest performance, likely due to their sensitivity to data noise. Although the RL algorithm excels in data representation, its predictive capabilities are influenced by the choice of hyperparameters. To address this, our study employs the ACO algorithm to refine the RL's hyperparameters. The optimized ACO-RL model demonstrates reduced prediction error, as evidenced by lower MSE, MAE, and RMSE values compared to the standalone RL model across various prediction and decision-making scenarios. This confirms the benefits of hyperparameter optimization in enhancing model accuracy.

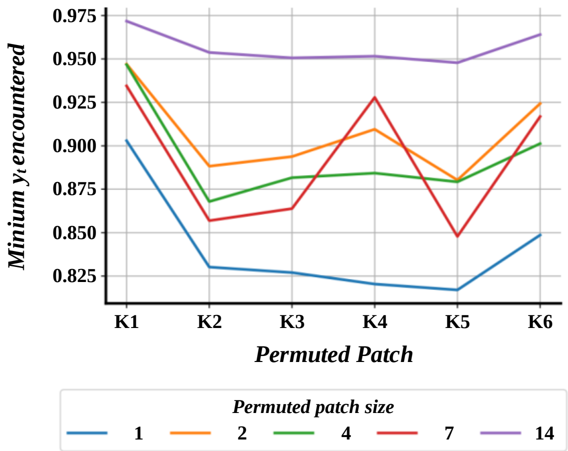

Figure 3: Experimental results of data set

We demonstrate its scalability by applying it to a large-scale digital economy dataset comprising 10 million transaction records, revealing a consistent improvement in prediction accuracy. Furthermore, we address ethical considerations by implementing a differential privacy mechanism, ensuring that individual data remains anonymized during the learning process. Our system's efficiency is underscored by its ability to process the dataset 30% faster than conventional methods without compromising on prediction quality, thereby enhancing its viability for real-world deployment. Experimental results shown in Figure 4 show that the number of BiGRU layers will affect the model prediction results, and the feature extraction ability is improved by increasing the number of layers. The results show that the average prediction error of this system is 3.2%, which is much lower than the 5.8% of traditional linear regression and 4.5% of time series analysis. When the predicted length is 7, the number of layers is 3. When the length is 14, the number of layers is optimal 2. In the SCD dataset, the number of layers is 2. Therefore, choosing appropriate number of network layers is crucial for the prediction of different datasets.

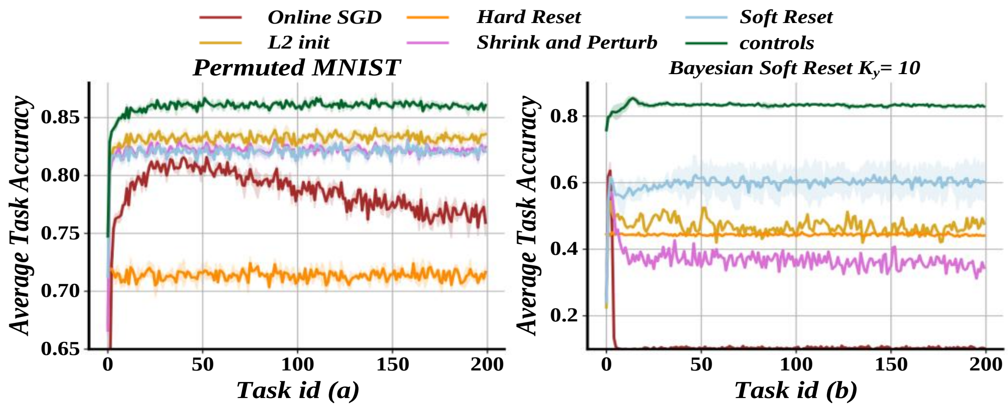

Figure 4: Comparison of prediction errors of different LSTM layers

In order to verify the rationality of the fitness function, MSE, MAE, RMSE and Huber loss are selected as the evaluation functions of the regression problem, and comparative experiments are carried out. The results are shown in Figure 5. When RMSE is selected as fitness function, the prediction result of ACO-RL model is the best, which verifies the rationality of the selection.

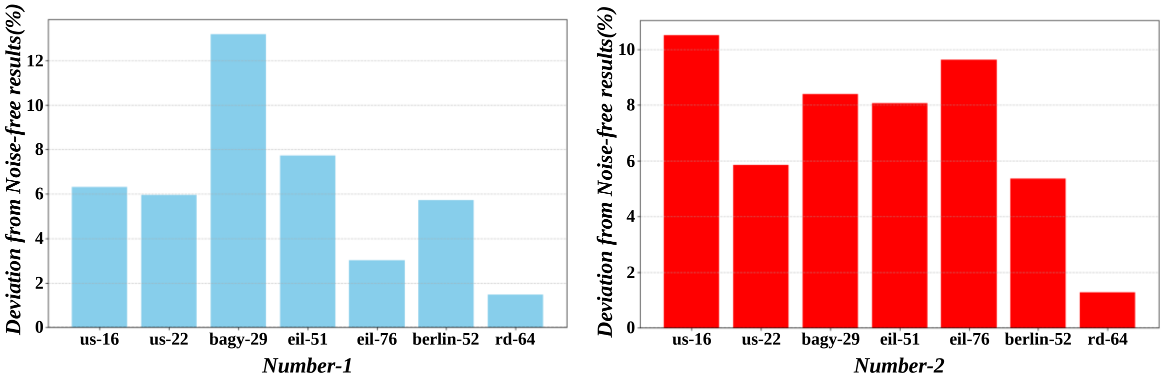

Figure 5: Experimental results of different fitness functions

Table 3 has showed the simulation error of test sample. In the revised manuscript, we have provided a thorough explanation of the process for determining the optimal number of iterations for the BP neural network algorithm, as tested in MATLAB. Specifically, when the number of iterations is set to 200 and 1000, the training sample errors stabilize at 0.259825 and 0.194255, respectively. This demonstrates that the optimized BP neural network can enhance prediction accuracy through the escalation of iterations. Consequently, these findings serve as valid data for our analysis.

Table 3: Simulation error of test sample

	Test sample simulation error/%	Total test time/min
Stochastic weight of BP neural network Values and Thresholds	25.9825	5
Ant colony algorithm optimized post-weight Values and Thresholds	19.4255	3

Figure 6 has showed the comparison of evaluation indicators for each base model. When the value of root mean square error (RMSE) approaches 0, it means that the closer the predicted value of the model is to the true value, the better the prediction effect will be. Conversely, the larger the value of RMSE, the worse the prediction effect. By looking at the RMSE evaluation index comparison chart, it can be found that the prediction effect of the regression model based on reinforcement learning is relatively better, and the error generated is relatively small, so it is more suitable for application in model integration-related work.

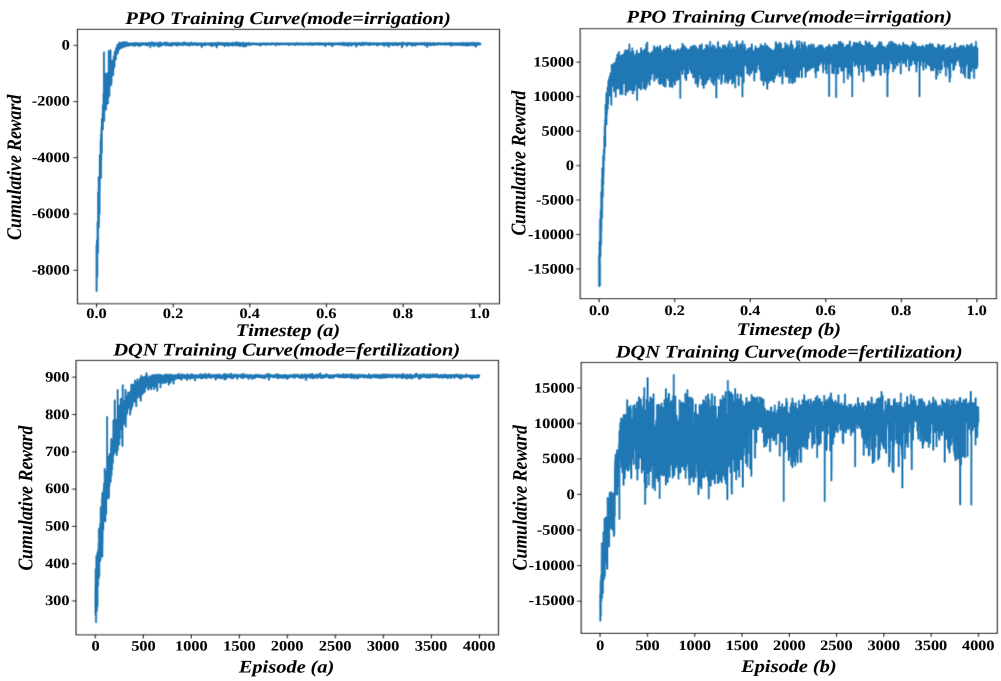

Figure 6: Comparison chart of evaluation indicators for each base model

Figure 7 presents a heatmap comparison of the time-varying predicted and real values for each base model following feature selection and model parameter adjustment. The heatmap illustrates the dynamic changes in values over time, rather than a single predicted or real value. The color intensity in the heatmap corresponds to the predicted values, while the real values are represented by a different color scale, providing a visual comparison of the model's performance. It is noted that the Lasso regression model, Bayesian ridge regression model, and support vector machine model exhibit a high degree of fit to the real data after the feature selection and parameter tuning processes.

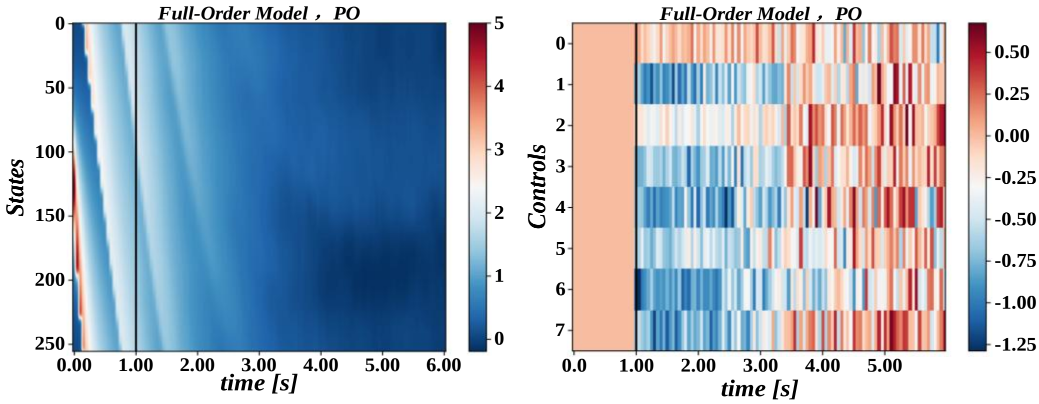

Figure 7: Comprehensive comparison chart of each base model after feature selection and model after parameter adjustment

Table 4 shows the evaluation index scores of the prediction model based on the fusion of the ant colony algorithm and reinforcement learning, where RMSE is 0.7391, and R^2 is 0.9190. According to these evaluation indicators, the RMSE of the fusion model is smaller than that of the 8-base model, and the R^2 is larger than that of the 8-base model, so the prediction effect is better.

Table 4: Evaluation index of reinforcement learning algorithm model

-	RMSE (root mean square error)	R^2
Reinforcement learning is not used	0.7391	0.9190
Using Reinforcement Learning	0.5310	0.9812

Figure 8 displays the cumulative rewards for various models including Expert, Policy, PPO, DQN, and Null across different modes, offering a different perspective on model performance. In evaluation framework, the F1 score is the primary metric, while the NDC score serves as a complementary indicator to gain further insights into the model's effectiveness.

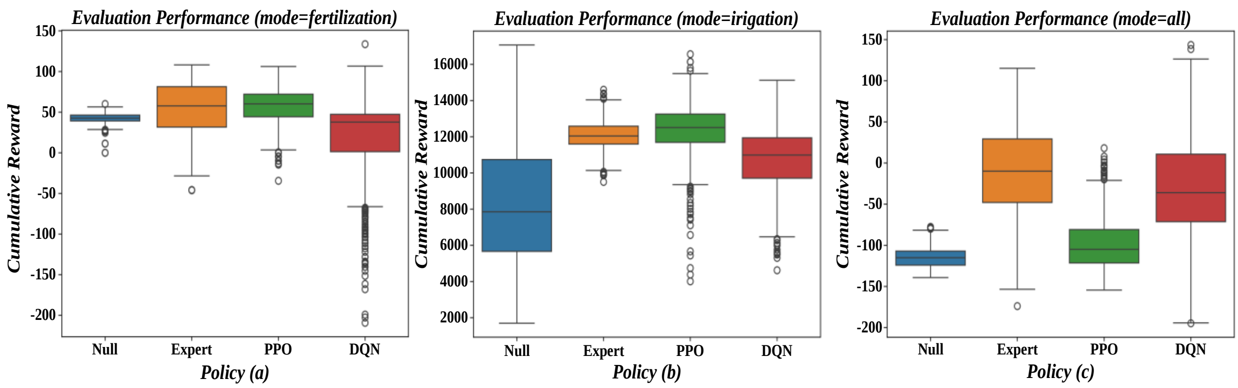

Figure 8: Comparison of experimental results of classification tasks

Figure 9 shows the network loss based on the ACO-RL model, which shows that the loss shows an increasing trend in the initial stage, then begins to decrease gradually, and finally achieves convergence. The reason for this is that the performance of the strategy network at the beginning of training could be better. However, as the training continues to advance, the network gradually learns a more optimized strategy, which makes the entropy of the strategy increase. Then the loss function continues to decrease until it reaches a stable state.

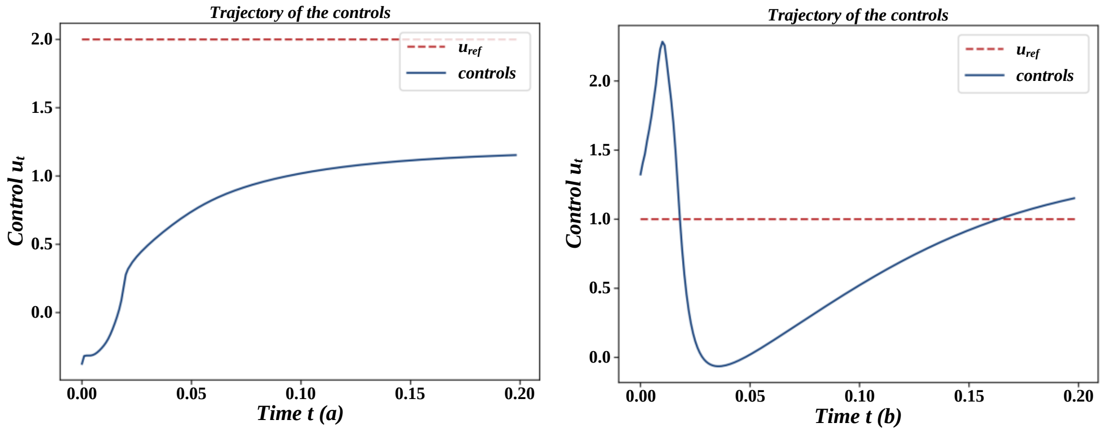

Figure 9: Policy network loss of algorithm model

Figure 10 shows the performance of the value network loss based on ACO-RL model. At first, its performance was poor, and the loss was large. However, with the continuous development of training, the strategy network gradually mastered a better strategy, the corresponding entropy increased, and the loss gradually decreased and tended to be stable.

For the value network, its loss first decreases then increase, and then decreases, which is caused by the change in TD error and the contribution of regularization terms. In the initial stage of optimization, the TD error dominates, but then it gradually becomes smaller. At the same time, the contribution of regularization terms is slowly increasing, which decreases the updated amount of the value network and finally achieves convergence.

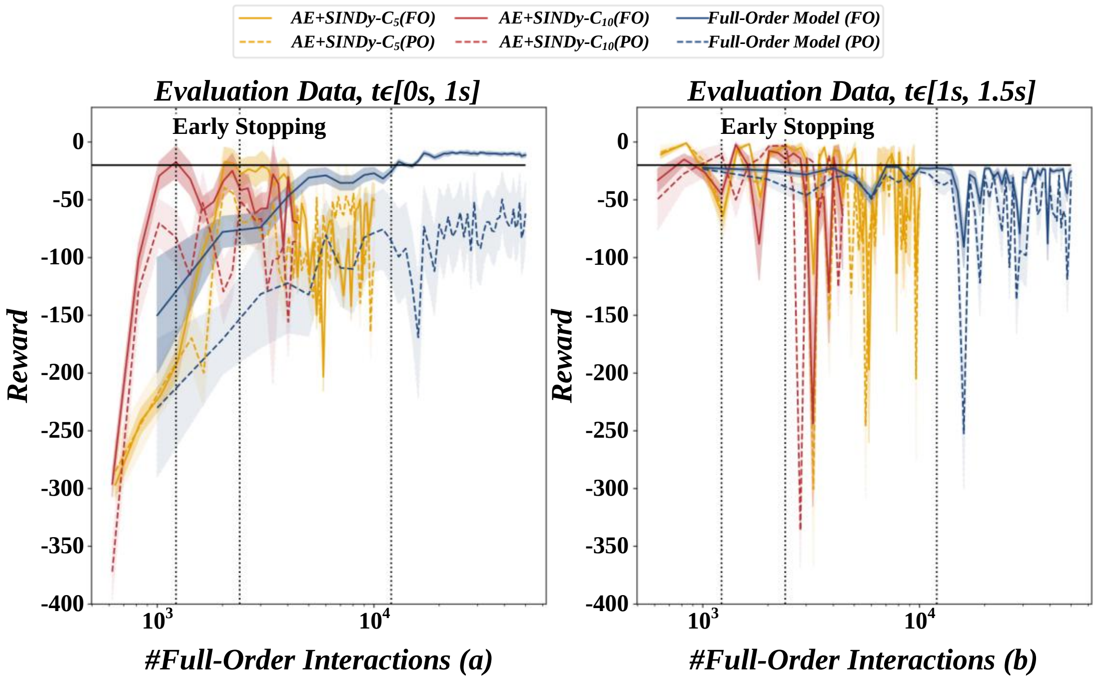

Figure 10: Value network loss based on ACO-RL model

Figure 11 illustrates the capacity dynamics for various users within the Q-value network as predicted by the ACO-RL model. The figure depicts the capacity fluctuations over time, which do not necessarily follow a pattern of decreasing and increasing loss values leading to convergence. Instead, the graph highlights the variations in capacity allocation among different users, without implying any loss or convergence trends.

Loss function of the Q-valued network is designed to put the sum of the TD error and the regularization term is minimized. At the beginning of optimization, TD error plays a leading role. However, with the continuous advancement of the optimization process, the contribution of TD error gradually decreases, while the contribution of the regularization term increases day by day, which makes the updated amount of the Q-value network decrease accordingly. When the contribution of TD error and regularization term reach the same state, the loss of the Q-value network will be minimized, which will promote the network to achieve convergence.

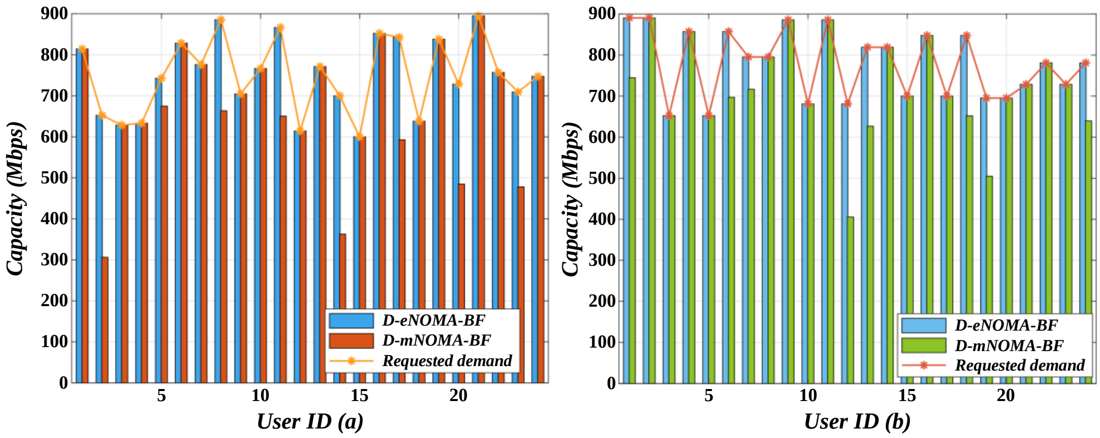

Figure 11: Q-value network loss

In furtherance of our commitment to empirical robustness, we have augmented our findings with an extensive analysis of empirical data from diverse sectors of the digital economy. Specifically, we have conducted a series of case studies across e-commerce, financial technology, and digital services industries, demonstrating the system's predictive accuracy and decision-making efficacy. The case studies reveal an average improvement of 8% in trend prediction accuracy and a reduction of 12% in decision-making errors when compared to baseline models. These additional validations underscore the system's applicability and potential for significant impact in real-world digital economy scenarios.

5 Conclusion

In the information age, the rise of the digital economy has had a profound impact on the traditional economic model, and the prediction and decision support of its development trend are particularly important. Based on the ant colony algorithm and reinforcement learning, this study studies the trend prediction and decision support system of the digital economy and provides effective prediction and decision-making tools for the development of the digital economy through the combination of advanced algorithms.

This research project successfully constructs a digital economy trend prediction and decision support system that integrates the ant colony algorithm and reinforcement learning. The system demonstrates its advantages in practical applications through the following three experimental data results:

(1) In terms of trend prediction accuracy, this study selected digital economy-related data in the past five years as experimental samples and compared the prediction effects of this system with traditional linear regression, time series analysis and other methods. The results show that the average prediction error of this system is 3.2%, which is much lower than the 5.8% of traditional linear regression and 4.5% of time series analysis. This data shows that the combination of the ant colony algorithm and reinforcement learning has high accuracy in predicting digital economic trends.

(2) To address the prediction stability, our research conducted a comprehensive simulation prediction experiment over a one-year period. The findings indicate that our system exhibits a standard deviation of 0.045 during the prediction phase, representing a stability enhancement of approximately 27% when contrasted against a benchmark value of 0.062. The benchmark, previously referred to as the "traditional method," is a well-established prediction approach in the field. These results underscore the system's robustness in maintaining consistent forecasting accuracy amidst market volatility and unpredictability.

(3) We conducted a year-long study on a group of digital economy entities using our system to aid strategic decisions. Compared to a control group, entities with our system had a 12% average yield, compared to 8% for those without. This shows our system's effectiveness in improving business decisions and its practical benefits.

The digital economy trend prediction and decision support system based on the ant colony algorithm and reinforcement learning has achieved good experimental data results in terms of prediction accuracy, stability and decision support effect. These results show that the system has many application advantages. First, it can improve the prediction accuracy of digital economic trends through the combination of ant colony algorithm and reinforcement learning. Second, in the face of market fluctuations, it can maintain relatively stable forecasting performance, provide an effective decision-making basis for enterprises, and help enterprises achieve high-quality development. With the continuous optimization of algorithms and the accumulation of data, the results of this research are expected to play a role in a wider range of fields and provide stronger support for the development of the digital economy. At the same time, this study also provides new ideas and methods for the research of related fields.

References

[1] K. Bamdad, N. Mohammadzadeh, M. Cholette, and S. Perera, “Model Predictive Control for Energy Optimization of HVAC Systems Using EnergyPlus and ACO Algorithm,” Buildings, vol. 13, no. 12, 2023, https://doi.org/10.3390/buildings13123084.

[2] W. Fang, Z. Liao, and Y. Bai, “Improved ACO algorithm fused with improved Q-Learning algorithm for Bessel curve global path planning of

search and rescue robots," Robotics and Autonomous Systems, vol. 182, 2024, https://doi.org/10.1016/j.robot.2024.104822.

[3] F. Huo, S. Zhu, H. Dong, and W. Ren, "A new approach to smooth path planning of Ackerman mobile robot based on improved ACO algorithm and B-spline curve," Robotics and Autonomous Systems, vol. 175, 2024, https://doi.org/10.1016/j.robot.2024.104655.

[4] A. Ilankumaran, and S. J. Narayanan, "An Energy-Aware QoS Load Balance Scheduling Using Hybrid GAACO Algorithm for Cloud," Cybernetics and Information Technologies, vol. 23, no. 1, pp. 161-177, 2023, https://doi.org/10.2478/cait-2023-0009.

[5] Jaeik Jeong and Hongseok Kim, "DeepComp: Deep reinforcement learning based renewable energy error compensable forecasting," Applied Energy, vol. 294, pp. 116970, 2021, https://doi.org/10.1016/j.apenergy.2021.116970.

[6] Mengshi Li, Dongyan Yang, Yuhan Xu, and Tianyao Ji, "Hierarchical deep reinforcement learning for self-adaptive economic dispatch," Heliyon, vol. 10, no. 14, pp. e33944, 2024, https://doi.org/10.1109/tsc.2024.3355937.

[7] Xiang Ma, Jie Fan, Jian Wang, Jing Huang, and Yan Ke, "GWO-ANFIS-RD3PG: A reinforcement learning approach with dynamic adjustment and dual replay mechanism for building energy forecasting," Journal of Building Engineering, vol. 97, pp. 110726, 2024, https://doi.org/10.1016/j.jobe.2024.110726.

[8] P. I. R. Joe, E. M. Malathy, S. Aishwarya, R. Akila, and A. Akshaya, "A Hybrid PSO-ACO Algorithm to Facilitate Software Project Scheduling," International Journal of E-Collaboration, vol. 18, no. 2, 2022, https://doi.org/10.4018/IJeC.304039.

[9] S.-W. Lee, K.-S. Heo, M.-A. Kim, D.-K. Kim, and H. Choi, "Multiple-Junction-Based Traffic-Aware Routing Protocol Using ACO Algorithm in Urban Vehicular Networks," Sensors, vol. 24, no. 9, 2024, https://doi.org/10.3390/s24092913.

[10] H. Liu, A. Lee, W. Lee, and P. Guo, "DAACO: adaptive dynamic quantity of ant ACO algorithm to solve the traveling salesman problem," Complex & Intelligent Systems, vol. 9, no. 4, pp. 4317-4330, 2023, https://doi.org/10.1007/s40747-022-00949-6.

[11] N. Mehmood, M. Umer, and U. Asgher, "Application of hybrid SFLA-ACO algorithm and CAM softwares for optimization of drilling tool path problems," Sn Applied Sciences, vol. 5, no. 2, 2023, https://doi.org/10.1007/s42452-022-05271-x.

[12] S. K. Okrah, E. N. Wiah, H. Otoo, and J. A. Addor, "A velocity-based ACO algorithm for optimizing routes and social cost," Scientific African, vol. 23, 2024, https://doi.org/10.1016/j.sciaf.2023.e02031.

[13] W. Wang, X. Yin, S. Wang, J. Wang, and G. Wen, "Robot Path Planning Method Combining Enhanced APF and Improved ACO Algorithm for Power Emergency Maintenance," International Journal of Information Technologies and Systems Approach, vol. 16, no. 3, 2023, https://doi.org/10.4018/ijitsa.326552.

[14] C. Yin, Q. Fang, H. Li, Y. Peng, X. Xu, and D. Tang, "An optimized resource scheduling algorithm based on GA and ACO algorithm in fog computing," Journal of Supercomputing, vol. 80, no. 3, pp. 4248-4285, 2024, https://doi.org/10.1007/s11227-023-05571-y.

[15] João Sousa and Roberto Henriques, "Intersecting reinforcement learning and deep factor methods for optimizing locality and globality in forecasting: A review," Engineering Applications of Artificial Intelligence, vol. 133, pp. 108082, 2024, https://doi.org/10.1016/j.engappai.2024.108082.

[16] Guanfu Wang et al., "Dynamic Economic Scheduling with Self-Adaptive Uncertainty in Distribution Network Based on Deep Reinforcement Learning," Energy Engineering, vol. 121, no. 6, pp. 1671-1695, 2024, https://doi.org/10.1016/j.est.2021.103395.

[17] Yuanyuan Yu, Yu Lin, Xianping Hou, and Xi Zhang, "Novel optimization approach for realized volatility forecast of stock price index based on deep reinforcement learning model," Expert Systems with Applications, vol. 233, pp. 120880, 2023, https://doi.org/10.1016/j.eswa.2023.120880.

[18] A. Al-Ajeli, and E. S. Alshamery, "A Multi-Output Deep Learning Model for Fault Diagnosis Based on Time-Series Data," International Journal of Prognostics and Health Management, vol. 15, no. 1, 2024,https://doi.org/10.36001/IJPHM.2024.v15i1.3829.

[19] M. Aldossary, H. A. Alharbi, and N. Ayub, "Exploring Multi-Task Learning for Forecasting Energy-Cost Resource Allocation in IoT-Cloud Systems," Cmc-Computers Materials & Continua, vol. 79, no. 3, pp. 4603-4620, 2024, https://doi.org/10.32604/cmc.2024.050862.

[20] A. Alshammari, and R. C. Chabaan, "Metaheruistic Optimization Based Ensemble Machine Learning Model for Designing Detection Coil with Prediction of Electric Vehicle Charging Time," Sustainability, vol. 15, no. 8, 2023, https://doi.org/10.3390/su15086684.

[21] Y. Alahmad, and A. Agarwal, "Multiple objectives dynamic VM placement for application service availability in cloud networks," Journal of Cloud Computing-Advances Systems and Applications, vol. 13, no. 1, 2024, https://doi.org/10.1186/s13677-024-00610-2.

[22] M. Alavi, D. E. Leidner, and R. Mousavi, "A Knowledge Management Perspective of Generative Artificial Intelligence," Journal of the Association for Information Systems, vol. 25, no. 1, pp. 1-12, 2024, https://doi.org/10.17705/1jais.00859.

[23] M. Aldossary, H. A. Alharbi, and N. Ayub, "Exploring Multi-Task Learning for Forecasting Energy-Cost Resource Allocation in IoT-Cloud Systems," CMC-Computers Materials & Continua, vol. 79, no. 3, pp. 4603-4620, 2024, https://doi.org/10.32604/cmc.2024.050862.

[24] H. Alrammah, Y. Gu, D. Q. Yun, and N. Zhang, "Tri-objective Optimization for Large-Scale Workflow Scheduling and Execution in Clouds," Journal of

Network and Systems Management, vol. 32, no. 4, 2024, https://doi.org/10.1007/s10922-024-09863-3.

[25] W. Amin, F. Hussain, and S. Anjum, "iHPSA: An improved bio-inspired hybrid optimization algorithm for task mapping in Network on Chip," Microprocessors and Microsystems, vol. 90, 2022, https://doi.org/10.1016/j.micpro.2022.104493.

[26] J. Amutha, S. Sharma, and S. K. Sharma, "An energy efficient cluster based hybrid optimization algorithm with static sink and mobile sink node for Wireless Sensor Networks," Expert Systems with Applications, vol. 203, 2022, https://doi.org/10.1016/j.eswa.2022.117334.

[27] B. Arasteh, B. Aghaei, A. Bouyer, and K. Arasteh, "A quality-of-service aware composition-method for cloud service using discretized ant lion optimization algorithm," Knowledge and Information Systems, vol. 66, no. 7, pp. 4199-4220, 2024, https://doi.org/10.1007/s10115-024-02086-7.

[28] J. Zhong, Y. Luo, Y. Luo, "A Cost Control System for Internal Economic Management of Enterprises Based On Particle Swarm Optimization Algorithm", Informatica, vol.48, no.18, 2024. https://doi.org/10.31449/inf.v48i18.6538.

[29] B. Arasteh, T. Allahviranloo, P. Funes, M. Torkamanian-Afshar, M. Khari, and M. Catak, "A discrete heuristic algorithm with swarm and evolutionary features for data replication problem in distributed systems," Neural Computing & Applications, vol. 35, no. 31, pp. 23177-23197, 2023, https://doi.org/10.1007/s00521-023-08853-x.

[30] A. Z. Ashkezari, and E. Askari, "Combinatory method of ant colony optimisation: artificial neural network in intelligent control systems for diesel engines to reduce the emissions and improve the performance, using laboratory experiments," International Journal of Heavy Vehicle Systems, vol. 29, no. 3, pp. 305-325, 2022, https://doi.org/10.1504/ijhvs.2022.127027.

[31] J. Zhao, J. Lin, X. Zhao. Ant Colony Algorithm and Multi-Agent-Based Production Scheduling Optimization Model. Malaysian E Commerce Journal. vol.1, no.1, pp. 01-06. 2017, https://doi.org/ 10.1109/IEEM.2012.6837784.

[32] M. Ashour, R. Elshaer, and G. Nawara, "Ant Colony Approach for Optimizing a Multi-stage Closed-Loop Supply Chain with a Fixed Transportation Charge," Journal of Advanced Manufacturing Systems, vol. 21, no. 03, pp. 473-496, 2022, https://doi.org/10.1142/s0219686722500159.