IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 December 2023, accepted 30 December 2023, date of publication 9 January 2024,
date of current version 25 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3351696

RESEARCH ARTICLE

An Improved Reinforcement Learning Method Based on Unsupervised Learning

XIN CHANG[1], YANBIN LI[1], GUANJIE ZHANG[1], DONGHUI LIU[2], AND CHANGJUN FU[1]
[1]The 54th Research Institute of China Electronics Technology Group Corporation (CETC54), Shijiazhuang 050081, China
[2]School of Management, Shijiazhuang Tiedao University, Shijiazhuang 050043, China

Corresponding author: Xin Chang (changxinydb@163.com)

This work was supported in part by the China Postdoctoral Science Foundation under Grant 2021M693002.

ABSTRACT The approach of directly combining clustering method and reinforcement learning (RL) will lead to encounter the issue where states may have different state transition processes under the same action, resulting in poor policy performance. To address this challenge with multi-dimensional continuous observation data, an improved reinforcement learning method based on unsupervised learning is proposed with a novel framework. Instead of dimensionality reduction methods, unsupervised clustering is employed to indirectly capture the underlying structure of the data. First, the proposed framework incorporates multi-dimensional information, including the current observation data, the next observation data and reward information, during the clustering process, leading to a more accurate and comprehensive low-dimensional discrete representation of the observation data while retaining preserving transition of Markov decision process. Second, by compressing the observation data into a well-defined state space, the resulting cluster labels serve as the low-dimensional discrete label-states for reinforcement learning to generate more effective and robust policies. Comparative analysis with state-of-the-art RL methods demonstrates that the improved RL methods base on framework achieves higher rewards, indicating its superior performance. Furthermore, the framework exhibits computational efficiency, as evidenced by its reasonable time complexity. This structural innovation allows for better exploration and exploitation of the transition, leading to improved policy performance in engineering applications.

INDEX TERMS Reinforcement learning, unsupervised learning, supervised learning, deep learning, dimensionality reduction.

I. INTRODUCTION

Reinforcement learning (RL) has emerged as a powerful approach for learning state-to-action mappings to achieve goals in various domains [1], [2], such as conversational voice assistant [3], autonomous vehicles [4] and game [5], [6]. In the field of communication countermeasure, deep reinforcement learning (DRL) methods have demonstrated their prowess by generating jamming or anti-jamming policies directly from multi-dimensional observation data using large-scale neural networks [7], [8], [9], [10], [11]. The multi-dimensional observation data is defined as that data is composed of multiple parameters intercepted by communication countermeasure devices, referred to as jammers. However, its application in engineering practice faces four challenges, particularly in communication countermeasure device-level applications. The first challenge faced by jammers is multi-dimensional and continuous observation data. It's evident that as the dimensionality of the continuous observation data increases, the neural networks used for jamming policy generation need to be larger, involving a greater number of parameters that require optimization. From the perspective of computing resource, especially Graphics Processing Units (GPUs), which have become the foundation of artificial intelligence, the practical use of such methods in engineering applications may become more challenging [12], [13], [14]. This is especially pertinent when there are constraints related to communication efficiency, installation space, heat dissipation, and other factors. In certain scenarios, using a higher-dimensional observation data might offer more

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

precise representations. However, research is motivated by the practical constraints and device-level applications, where reducing state dimensionality is essential for feasibility and efficiency. The second challenge is hyperparameter tuning. Engineering applications typically require simplified and practical methods to ensure ease of implementation. This extends to hyperparameters tuning. If hyperparameters tuning becomes overly complex, the practical use of such methods in engineering applications may become more challenging. Strategies for hyperparameter tuning include trial-and-error [15], grid search [15], Bayesian optimization [16], [17], and evolutionary algorithms [18]. While strategies can help find the best combination of parameters, they may require substantial computational resources and time, especially in the case of evolutionary algorithms dealing with multi-objective optimization problems [19], [20]. The third challenge is insensitivity and robustness. Parameter insensitivity often indicates that a method exhibits strong robustness to changes in parameters, including jamming parameters and hyperparameters. This means that even without precise parameters tuning, the jamming method can still perform well [21], [22], [23], [24]. This is highly relevant in practical engineering applications where parameter choices may be constrained or fine-tuning may be difficult. The fourth challenge is interpretability. End-to-end lacks interpretability, and the lengthy waiting during training can test engineers' patience and doesn't easily provide interim engineering results and explanations for the causes [25].

Therefore, the challenge lies in efficiently and effectively generating and optimizing policies with low computational demands while maintaining interpretability. Research aims to address specific challenges related to observation data dimensionality reduction, extracts low-dimensional features and policy interpretability in jammers. Machine learning and RL are employed as alternatives to deep reinforcement learning, thereby avoiding the computational complexity introduced by neural network and hyperparameter tuning. By first extracting feature representations from communication parameters to estimate the communication device's modes, the multi-dimensional continuous observation data are transformed into low-dimensional discrete label-state, referred to as label-state. Then, by utilizing RL based on these label-states, jamming policies are generated, enhancing the interpretability of policy generation.

Dimensionality reduction methods of unsupervised learning (UL), particularly Principal Component Analysis (PCA) [26] and Auto Encoder [27], serve as crucial approaches for realizing this idea. These techniques can achieve high performance in reducing data dimensionality by addressing the correlation among features, finding a low-dimensional representation of the data that retains the most important variation [28]. However, solely focusing on dimensionality reduction may result in a loss of state diversity, potentially leading to inaccuracies in representing the underlying Markov decision process (MDP), which leads to encounter the issue where states may have different state transition processes under the same action.

An alternative approach is to use clustering results as the discrete state and extracting the feature of the observation data. Clustering methods like k-Means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) have shown excellent performance in automatically identifying cluster structures and determining the number of clusters [28]. However, DBSCAN requires global parameters tuning, making it less suitable as a key component of an enhanced solution, including the distance of the cluster formation and the minimum number of data objects [29]. Improved DBSCAN methods reduce the complexity of parameter selection, and DBSCAN clustering parameters could be adjusted directly using expert knowledge or indirectly through RL or genetic algorithms [30], including the cutoff distance parameters, grid parameters, Gaussian distribution parameter or fixed MinPts parameters [29]. While subsequent derivative algorithms may address the challenges of parameter selection, it remains to be seen whether their performance as stable as the DBSCAN implementation provided in packages like sklearn. On the other hand, k-Means is a classic clustering algorithm that can estimate the number of clusters based on significant changes in the clustering curve or predetermined criteria like the silhouette score [31]. However, if k-Means directly applied to group the most similar continuous observation data and create label states, it will lead to clustering error and is possible to encounter the issue where states may have different state transition processes under the same action.

To address these limitations, a RL framework is proposed that integrates UL and feature engineering to optimize policies in multi-dimensional continuous observation data. Feature engineering incorporates MDP transition information by augmenting the current observation data with the future observation data and rewards obtained after taking actions, which ensures that clustered states within the same label share the same state transition processes under the same action. The augmented data, referred to as feature data, undergoes k-Means to obtain labels, which are then transformed into states, effectively reducing the dimensionality. RL are subsequently employed to maximize the cumulative reward based on the reduced state space [32], [33]. By utilizing label-state, the underlying MDP transitions can be more easily captured, leading to more efficient policy generation and optimization. Importantly, the number of meaningful components in the reduced label-state space is significantly smaller than the dimensionality reduced by traditional dimensionality reduction methods.

Experimental evaluations are conducted to compare the proposed framework with classical direct combination methods and state-of-the-art RL methods. The results demonstrate the superiority of the proposed method in terms of policy performance. The contributions of this paper lie in presenting a novel framework for multi-dimensional continuous

observation data that combines feature engineering, UL, and RL to enhance performance and offer a novel view on how to express, explain and define label-state.

The remainder of this paper is organized as follows: Section II introduces a problem faced by RL methods, while the idea of the potential solution are discussed. Additionally, the phased process of the improved method is proposed in Section III. Simulation experiments are performed and the results are analyzed in Section IV. Finally, conclusions are drawn in Section V.

II. PROBLEM FORMULATION AND SOLUTION

In this section, the specific scenario is performed to clear the scope of study, the challenges faced by RL methods are outlined and the idea of proposed solution is presented.

A. SCENARIOS

It's important to note that while the inspiration for this method comes from the field of communication countermeasure [34], [35], goal is to create a more universal environment to showcase the characteristics of the method proposed, with the hope of establishing it as a paradigmatic approach. Therefore, we strive to minimize specialized knowledge.

In the communication countermeasure environment based on the Markov Decision Process (MDP), it typically consists of four key components: observation space, action space, modes transition rules, and rewards [32].

The observation space is composed of communication parameters received and estimated by jammer, and these parameters are typically multi-dimensional and continuous. The distinction in communication device's modes is based on variations in the parameters, such as signal modulation type, signal modulation parameters, modulation order, center frequency, signal bandwidth, data rate, symbol rate, hop rate, hop range, and more [9], [36], [37]. Jammers also determine their jamming patterns through estimating communication parameters. However, considering factors, including adaptability in different modes and the estimated errors of jammers, the communication parameters obtained by jammer are distributed within a certain range. To better abstract the problem, the normalized observation space is divided into subspaces, corresponding to the number of different modes.

The action space is discrete. Typical jamming patterns exist, and the action space dimensions can be generated by adjusting the jamming parameters [8]. Without loss of generality, assume that the dimensions of the action space correspond to the number of modes.

The transition of the communication device's modes is deterministic under jamming [7]. When interfered with by the jammer, communication devices deterministically transitions to a specific mode, rather than relying on probability distributions. Simultaneously, the observation data is randomly generated from the subspace corresponding to the mode.

The rewards obtained by the jammer are derived from evaluating communication effect, which can analyzing the changes of observation data [9]. For instance, under jamming, the communication device's code rate decreases, the hop frequency range widens, and the increased delay caused by channel switching results in a decrease in information transmission rate, leading to degraded communication effect. As a result, the jammer obtains higher rewards.

In this paper, to avoid deviation from the main theme, the mechanism of communication mode transition and the evaluation of jamming effects are not delved into.

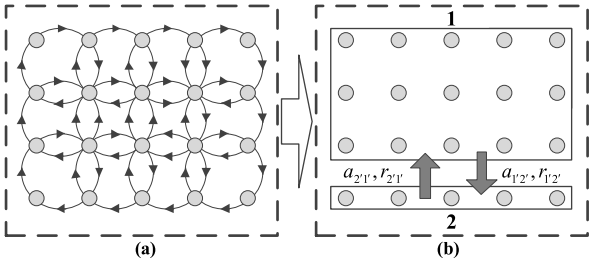

FIGURE 1. The misclassified observation data.

B. PROBLEM FORMULATION

The complexity and dynamics of real-world scenarios, like communication countermeasure, often result in multi-dimensional continuous observation data. The continuous exhibits high similarity after sampling in each dimension. This similarity can pose a challenge in accurately distinguishing different data, as the value differences between data is too small. As shown in Figure 1(a), the sampled observation data in specific scenarios exhibit proximity to each other in a two-dimensional plane, suggesting a high degree of similarity between the observed data points. If dimensionality reduction or clustering methods are directly applied to the data, they cannot abstract the feature of the communication modes. This is because these methods focus primarily on numerical similarity and reducing the dimensionality of the data. The value of data is too similar to distinguish. Furthermore, the data not contain the information, which can express the underlying relationships and transitions present in the MDPs. Then, due to misclassifying and misclustering observation data as state data, the estimated modes transitions may even become a random process, which means states may have different state transition processes under the same action, as shown in Figure 1(b). This misclassification prevents accurate capturing of the underlying modes transitions based on MDP. As a result, the compressed representation obtained through dimensionality reduction or clustering methods may not effectively abstract the feature of the modes from observation data and make it difficult to effectively represent the state space, hindering state-action value function, leading to suboptimal policy generation and decision-making. Therefore, it is crucial to correctly identify the differences between observation data and preserve the transition features during the dimensionality reduction or clustering method, ensuring the predictability and accuracy of state transitions to better guide the decision-making process.

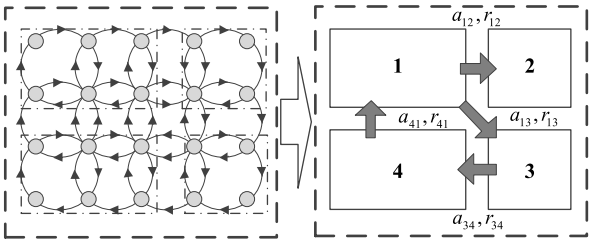

FIGURE 2. The low-dimensional label-state.

C. SOLUTION

As shown in Figure 2, this paper introduces a solution that incorporates UL and RL. The proposed framework improves upon existing approaches in two key aspects. First, the solution incorporates feature engineering to enhance the clustering process. By adding relevant features such as future observation data and rewards to the current observation data, the clustering method gains additional information about the transition with actions. This enriched feature representation enables the clustering method to change multi-dimensional continuous observation data into low-dimensional discrete label-state while preserving the essential information of mode transition. Second, this label-state enables the RL agent to make more informed decisions and optimize its policy more effectively, facilitating better policy generation and decision-making.

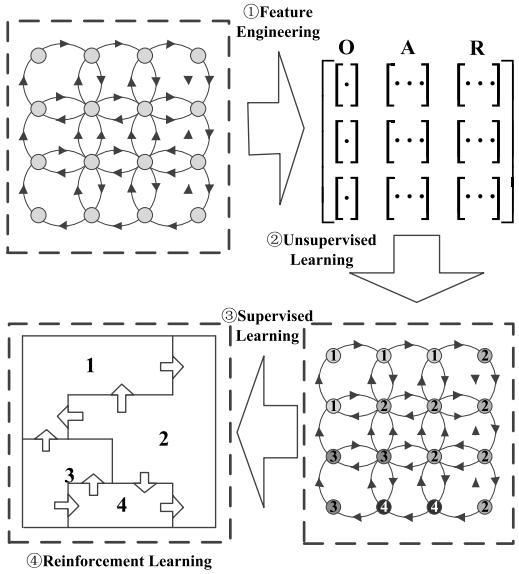

FIGURE 3. Flowchart of phased processing structure.

III. THE IMPROVED METHOD

This section provides a detailed description of the improved method. It follows a phased processing structure, which is illustrated in Figure 3. The flowchart outlines the step-by-step procedure, highlighting the key stages and their interconnections. It consists of four steps. First, feature engineering is performed to reconstruct the observation data into feature data. Second, the feature data is labeled using unsupervised learning methods, specifically k-Means. This process groups similar feature data together based on their characteristics, resulting in labeled clusters. Third, to transform the labeled clusters into states, supervised learning methods from machine learning are applied. This step involves training a model to establish a relationship between the observation data and the corresponding cluster labels. Final, the obtained labels are used as states with RL to generate an optimal policy.

A. FEATURE ENGINEERING

To address the challenge of reducing dimensionality without losing the transition, this section introduces a method that incorporates the current observation data, future observation data obtained after action selection and execution, and rewards to create feature data. The structure of feature data is illustrated in Figure 4.

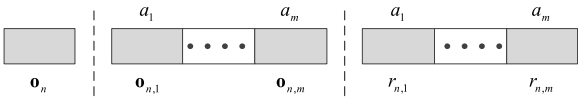

FIGURE 4. The structure of feature data.

In the figure, the nth feature data f_n consists of three parts. The first part is the current observation data $\mathbf{o}_n$. The second part is the observation data obtained after executing all executable actions starting from the current observation data $\mathbf{o}_{n,m}$. The third part is the reward $r_{n,m}$ obtained after executing all actions a_m.

It should be noted that, it is not feasible to traverse all possible outcomes of legal actions for a given observation data in real world. This method utilizes a digital twin simulation environment to generate offline data for enhancing the effectiveness of online policy generation. The digital twin simulation environment is a digital replica and simulation of the real, dynamic and complex scene.

In the dynamic scene, the process begins with random initial observation data $\mathbf{o}_n$ and involves traversing all valid actions a_m within the digital twin simulation environment to obtain corresponding future observation data $\mathbf{o}_{n,m}$ and rewards $r_{n,m}$. Subsequently, a future observation data point is randomly selected from $\mathbf{o}_{n,m}$, and the aforementioned process is repeated N times. Then, a feature sample set is generated. Importantly, this step does not necessitate the exhaustive traversal of all possible observation data; rather, it relies on the digital twin simulation environment's capacity to set observation data, enabling the repetition of all valid actions and acquisition of corresponding future observations and rewards using the same observation data. Consequently, this approach facilitates the improvement of online training through the utilization of offline data derived from the digital twin simulation environment.

B. LABEL-STATE GENERATION

During this phase, the collected feature sample set is performed by dimensionality reduction and unsupervised clustering method.

In the phase of dimensionality reduction, the main objective is to reduce the dimensionality of the data, which can be achieved through various methods such as Principal Component Analysis (PCA) and Autoencoder. However, it is crucial to strike a balance between dimensionality reduction and the performance. Excessive compression ratios can lead to clustering failure, where the inherent structure and transition may be lost. The comprehensive details of UL methods employed in this phase can be found in [28].

A function of dimensionality reduction M_1 is employed which takes in the nth multi-dimensional feature data f_n and accordingly outputs the nth low-dimensional feature data f'_n.

$$f'_n = M_1 (f_n) \tag{1}$$

In the clustering phase, the k-Means method plays a central role in this approach, leveraging its capability to address unsupervised clustering tasks without heavy reliance on prior knowledge. For a comprehensive understanding of k-Means and its application, especially in terms of how to choose the value of K, detailed descriptions can be found in [27]. The improved selection of the optimal K value can be summarized as follows: first, define the possible range of K value; then, calculate the silhouette score for each possible K value based on the feature sample set; finally, select the candidate K value corresponding to the highest silhouette score as the chosen optimal K value. Specifically, in this paper, it is necessary to ensure that the optimal K value is less than or equal to the median of the range of candidate K values. The choice of the range for the candidate K is related to the method's performance. Although increasing the range will lead to higher computational costs, the benefit of the method lies in its ability to enhance performance through using offline data before online training. The offline selection of K values is separate from the online policy generation. On one hand, offline selection allows adjustments to the range based on the trend of the silhouette score with varying K values, resulting in more accurate optimal K value estimation. On the other hand, offline selection of K values does not increase the computational burden during policy generation.

A function of unsupervised clustering M_2 that takes in the nth low-dimensional feature data f'_n and accordingly outputs the nth label-state l_n is employed.

$$l_n = M_2 \left(f'_n\right) \tag{2}$$

By leveraging the inherent properties of the k-Means method, such as its simplicity and efficiency, the proposed approach achieves effective clustering results that contribute to the subsequent steps.

C. MAPPING FUNCTION GENERATION

The training dataset consists of input observation data, denoted as $\mathbf{o}_n$, and corresponding desired output label-state, denoted as l_n. Utilizing supervised learning, the goal is to learn a mapping function that can accurately generate the label-state, l_n, for new, unseen current observation data, $\mathbf{o}_n$.

A function of supervised learning M_3 that takes in the nth multi-dimensional observation data o_n and accordingly outputs the nth label-state l_n is employed.

$$l_n = M_3 (o_n) \tag{3}$$

D. POLICY GENERATION

The Q-learning is a classical RL method for policy generation. Its detailed explanation can be found in [32], which provides a comprehensive overview of the Q-learning and its theoretical foundations. In the proposed framework, the traditional Q-learning is improved to enhance its performance. Figure 5 illustrates the improved Q-learning process.

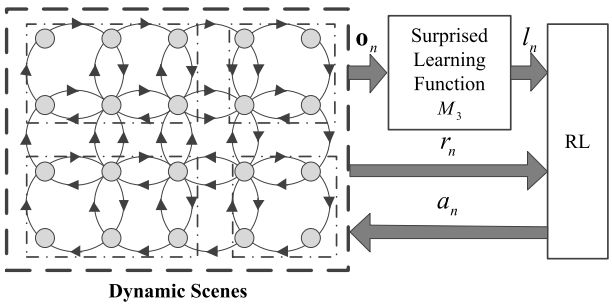

FIGURE 5. Flowchart of the improved RL.

Observation data is provided as input to the system, which is utilized to generate the corresponding low-dimension discrete label-state using the function of supervised learning M_3. When the agent is in the current label-state l_n, after accurately estimating the Q-values, the optimal policy is determined by selecting the action a_n that corresponds to the highest Q-value as follows:

$$a_n = \arg\max_a Q (l_n, a) \tag{4}$$

The estimates of Q-values can be gradually improved as follows:

$$Q (l_n, a) \leftarrow Q (l_n, a) + \eta \left[r + \max_{a' \in A} Q \left(l', a'\right) - Q (l_n, a) \right] \tag{5}$$

where Q is Q-value, η is the discount factor, l_n is the current label-state, r is the immediate reward, l' is the future label-state.

E. PROCESS

To enhance the understanding of process, an explanation of the phased process is provided.

Step 1: the current observation data $\mathbf{o}_n$, the future observation data $\mathbf{o'}_n$ after execution and selection of action and rewards r are used to form feature data f_n.

Step 2: after applying dimensionality reduction method, each multi-dimensional feature data f_n is transformed

into its corresponding low-dimensional representation, low-dimensional feature data f_n'.

Step 3: after applying unsupervised clustering method, each low-dimensional feature data f_n' is assigned to its corresponding label-state l_n.

Step 4: By employing supervised learning, the observation data $\mathbf{o}_n$ is mapped to label-state l_n representation.

Step 5: based on Q learning, the current observation data is mapped to a label-state representation using the label-state mapping from Step 4. Then, the value is estimated based on this label-state representation.

The detail process of proposed method is presented in Method 1.

Method 1: An improved reinforcement learning method based on unsupervised learning

Input: Observation data O, Action space A, Reward function R
Output: Optimal policy π
1. Preprocess the observation data o through feature engineering to obtain feature data f_n from $\mathbf{o}_n$, $\mathbf{o'}_n$ and r
2. Apply dimensionality reduction (e.g., PCA) to the feature data f_n, resulting in data f_n',from $f_n' = M_1(f_n)$
3. Apply unsupervised clustering (e.g., k-Means) to the data f_n', resulting in clustered labels l_n, from $l_n = M_2(f_n')$
4. Perform supervised learning to establish a mapping function between the observation data $\mathbf{o}_n$ and the clustered labels l_n, obtaining the label-state mapping M_3.
5. Initialize the Q-values Q for each label-state-action pair.
6. **Repeat** the following steps until a maximum number of iterations:
 6.1. **Initialize** the current observation data $\mathbf{o}_n$ as the initial observation.
 6.2. **Repeat** the following steps until reaching a terminal state:
  6.2.1 Apply supervised clustering to the data $\mathbf{o}_n$, resulting in clustered labels l_n, from $l_n = M_3(o)$.
  6.2.2. Execute the chosen action $a$and observe the reward r and the data $\mathbf{o'}_n$.
  6.2.3. Update the Q-value of the current label-state-action pair using the Q-learning update equation
  6.2.4. Set the current observation data $\mathbf{o}_n$ to the next observation data $\mathbf{o'}_n$.
7. Determine the optimal policy π based on the learned Q-values.
8. **Return** the optimal policy π.

IV. SIMULATION RESULTS AND ANALYSIS

In this section, experiments are designed and results are performed to present advantage of the proposed structure. First, in Section A, the simulation environment is detail expressed, and the value of setting parameters are given. Second, comparisons with performance of the improved method and that of straightforward combination method, are presented in Section B to further express difference. Third, in order to verify availability of the proposed structure, the performance of methods based on the structure are compared by that of classical state-of-the-art RL methods, such as Q learning, DQN, PPO and SAC.

A. SIMULATION ENVIRONMENT AND SETTING

While using real datasets or standard test environments can enhance the realism of the experiments, it may not necessarily contribute substantially to the main objectives of this paper. Standard testing environments may not fully encompass the expertise and characteristics of the discussed problem. Additionally, acquiring real datasets can be challenging and may raise privacy and confidentiality concerns. Constructing special test environments, which is controllable and transparent, allows for a clearer presentation and evaluation. This enables to conduct experiments more systematically and explore conclusions in a controlled and precise manner. It enhances the clarity and comprehensibility of research findings and ensures that the proposed method is not unduly influenced by specific details or complexities inherent in real datasets or publicly available testing environments. So a MDP environment is constructed to validate the effectiveness of the proposed method based on Section II-A.

The number of state is 16, and the dimensionality of observation is 10. Accordingly, the number of actions is 16, in which the next observation data will be transferred after getting action for the current observation data. Mode transition and rewards are set before simulation. Although simulation utilizes 16 states, adequate to demonstrate the core principles of the proposed method, such as achieving state reduction through clustering, enhancing interpretability, and improving policy generation. It allows us to illustrate how the proposed method operates effectively in practice, even when dealing with complex data. Furthermore, in comparison to traditional fields like image or audio processing, using a 10-dimensional observation data to describe the modes transition of a communication device under jamming is sufficient. This is because each dimension is not a pixel in an image or a sample point in a signal but rather represents parameters, such as signal modulation type, signal modulation parameters, modulation order, center frequency, signal bandwidth, data rate, symbol rate, hop rate, hop range, and more.

Because the performance of reinforcement learning is often unstable when running, 100 independent run of each method are performed in experiments to compare performance. For each run of a method, we pause training every 500 episodes and run 500 independent episodes with each method performing greedy action selection [32]. By analyzing the sum of reward in each episode, the performance of methods are discussed. The detailed simulation parameters are described in Table 1. The hyperparameters in this paper were not selected using any special tuning methods. While tuning hyperparameters does have a significant impact on method performance, the main emphasis is on the practicality and effectiveness of the proposed method. Therefore, a simple and easily implementable hyperparameter configuration is chosen [2], [32]. This aligns with the typical requirements for methods in engineering applications, which prioritize high robustness and interpretability.

B. COMPARISON WITH THE IMPROVED METHOD AND STRAIGHTFORWARD COMBINATION METHODS

In this section, comparison with the improved method and straightforward combination method are presented to

TABLE 1. Simulation parameters.

Parameters	Value
the number of observation data obtained before training	1000
Learning rate	0.01
Discount factor	0.9
ε greedy	0.9

highlight the advantages of the proposed framework and to further illustrate the structure which is not straightforward combination method. The comparison is conducted from the perspectives of processing flow and performance. Performance comparison is carried out in four aspects: loss value, fitting accuracy, clustering accuracy, and reward. As direct methods commonly used in engineering, KQ method and AQ method are selected as benchmarks for comparison. The straightforward combination method based on Auto Encoder and Q leaning is tagged as AQ, The straightforward combination method based on k-Means and Q leaning is tagged as KQ, and the improved method based on structure is tagged as SQ. The flowchart of these methods are shown in Figure 6.

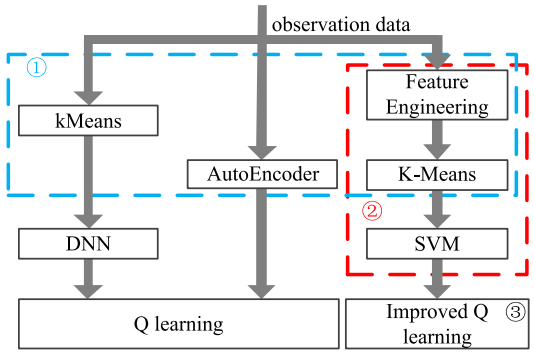

FIGURE 6. Flowchart comparison.

The comparison of the methods' structures reveals three main differences. First, as indicated by marker ① in the figure, traditional approaches often directly combine clustering method and RL to address the problem of multi-dimensional observation data. However, this direct combination approach often results in an incomplete state space. To overcome this issue, a new framework is introduced by feature engineering and k-Means. By incorporating multi-dimensional information during the clustering process, this framework compresses the observation space into a well-defined state space. Second, as indicated by marker ② in the figure, this paper primarily adopts machine learning methods. Although deep learning methods can provide more accurate results, advanced state representation and decision-making capabilities, especially when dealing with large-scale multi-dimensional observation data and abundant computational resources, machine learning methods are simpler and perform better in low-dimensional discrete label-state. Final, as indicated by marker ③ in the figure, the current observation data is mapped to a label-state representation. Then, the value is estimated based on this label-state representation.

First, Monte Carlo experiments are introduced in loss and accuracy, and results are shown in Figure 7 and 8.

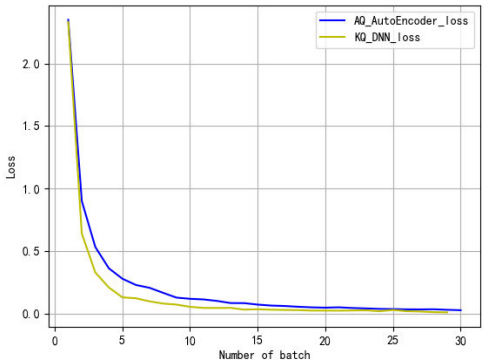

FIGURE 7. Performance comparison of loss.

By analyzing the decay rate and trend of the loss with the number of training batches in Figure 7, it can be observed that both the AutoEncoder and DNN models exhibit convergence in fitting the data. Additionally, by analyzing the Calinski-Harabasz Score curve in Figure 8, it can be observed that the curve is stable, indicating the stability of the clustering models. By analyzing the experimental results of accuracy and loss, it can be concluded that there is no issue of insufficient training leading to poor performance of the methods, which could potentially affect the credibility of subsequent experiments.

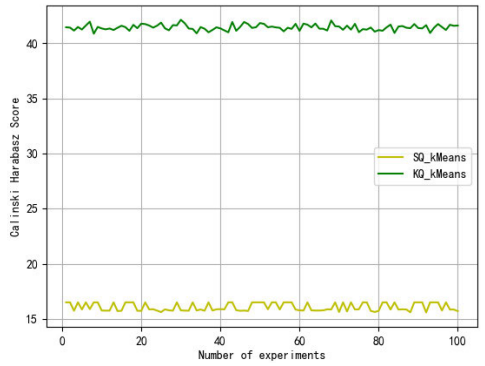

FIGURE 8. Performance comparison of accuracy.

Second, unsupervised clustering comparison is shown in Figure 9 and 10. It is important to emphasize that in real-world environments, there are no predefined K value, which represents the number of clusters, and classification labels available. In this paper, K value and classification labels are provided for evaluating the clustering results.

The curve represents the silhouette scores for different candidate K values in Figure 9, and the points on the curve indicate the optimal K values obtained using the K value selection method described in Section III of this paper. For AQ and KQ, their optimal K values are respectively 22 and 30, which deviates significantly from the true K value of 16.

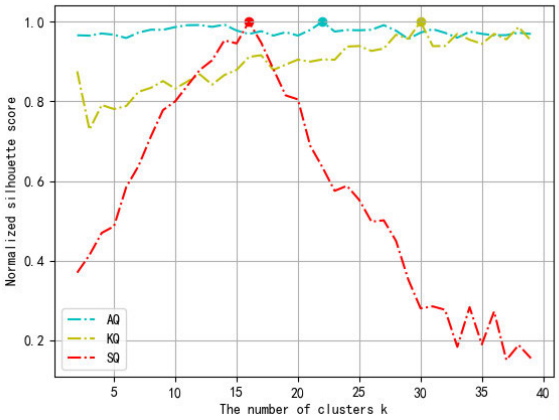

FIGURE 9. Performance comparison with the selection of k value.

However, based on the framework proposed in this paper, SQ correctly identifies the optimal K value equal to the true value. First, the experimental results demonstrate that the K value selection method in Section III-B effectively obtains the optimal K value. Second, the offline feature data generated enhances the clustering capability of the clustering method for continuous data, laying the foundation for generating a low-dimensional state space that can improve policy generation performance.

The observation data is labeled as shown in Figure 10 (a), in which label is proved by the environment for evaluation. The observation data, belongs to the same state, will be represented by the same color. For KQ, after k-Means processing, the observation data is labeled as shown in Figure 10 (b). Compared with Figure 10 (a) and (b), the results of clustering is chaos and irregular. After 100 Monte Carlo experiments, the index of similarity is approximately 0%, which means that the methods only use k-Means cannot acquire accurate feature of the modes. For AQ, after applying the Auto Encoder method for data dimension reduction, the clustering result of data is labeled as shown in Figure 10 (c). Comparing Figures 10 (a) and (c), it can be seen that the classification results are dissimilar. After 100 Monte Carlo experiments, the index of similarity is approximately 0%. For the improved method, after k-Means processing, the observation data is labeled as shown in Figure 10 (d). The results are obviously similar. After 100 Monte Carlo experiments, the index of similarity is approximately 100%.

Through the comparative analysis of similarity, it can be observed that the proposed framework effectively ensures the validity of clustering results by multi-dimensional continuous data.

Third, the comparison experiments in reward and time complexity are introduced in Figure. 11. The reward curves for 100 episodes are plotted in the graph. To enhance the clarity and highlight the trends in the line graph, the results in Figure 11 are smoothed. A moving average windows are used to reduce fluctuation and smooth curve of reward. In the original data, each experiment result will be replaced by the

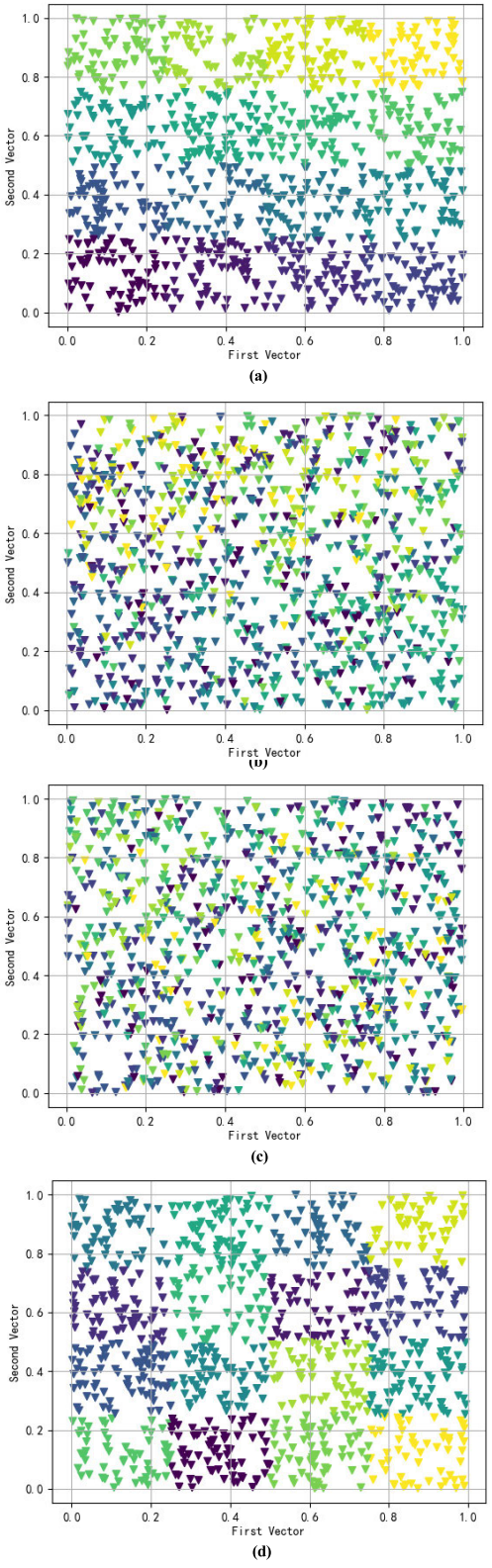

FIGURE 10. Unsupervised clustering comparison. (a) Presupposed label. (b) Clustering results of KQ. (c) Clustering results of AQ. (d) Clustering results of the proposed method.

average value of the adjacent 50 experiment results. Through comparative analysis of the reward curves, it can be observed

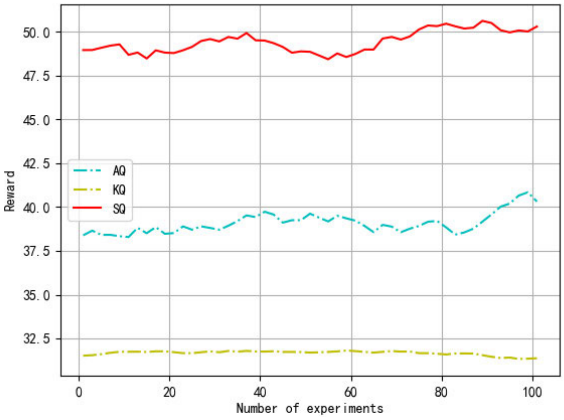

FIGURE 11. Performance comparison with the improved method and straightforward combination methods.

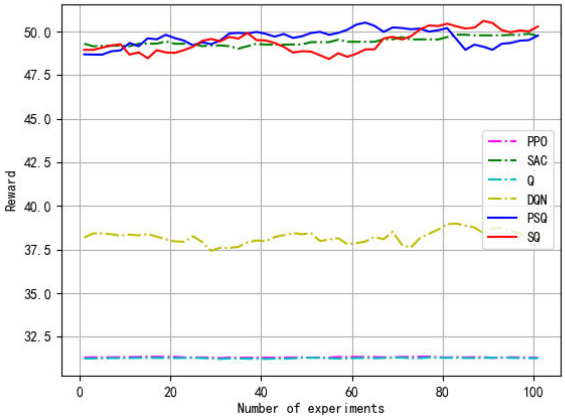

FIGURE 12. Performance comparison with the improved method and classical algorithms.

that the performance of the improved method outperforms the two baseline methods.

TABLE 2. Performance comparison with the improved method and straightforward combination methods.

Method	Average reward	Time complexity
AQ	39.11	166.03
KQ	31.41	968.86
SQ	49.39	1022.24

Then, for further analysis, the average reward and time complexity are calculated in Table 2. Table 2 presents a performance comparison among three methods: AQ, KQ, and SQ. The focus is on highlighting the performance advantage of SQ, despite its higher computational complexity.

Regarding average reward, the AQ method achieves a score of 39.11, while the KQ method performs at a lower level with a score of 31.41. In contrast, the SQ method outperforms both, demonstrating a significantly higher average reward of 49.39.

When considering time complexity, the AQ method exhibits a relatively low complexity of 166.03. The KQ method requires more computational resources, as indicated by its higher complexity of 968.86. In comparison, the SQ method presents a slightly higher time complexity of 1022.24.

These findings underscore the superior performance of the SQ method in terms of average reward, despite its increased computational demands.

In conclusion, the three validation results demonstrate that the performance of the improved method is superior to the two baseline methods. Furthermore, this indicates that the framework presented is not a simple combination of methods but rather a novel approach that yields improved performance.

C. COMPARISON WITH THE IMPROVED METHOD AND CLASSICAL ALGORITHMS

As shown in Figure 12, the performance of comparison among state of the art RL methods, such as Q-learning [32], DQN [32], [33], PPO [38] and SAC [39], [40], [41], and the improved method based on the proposed structure including SQ and PSQ. SQ is defined as an improved method based on k-Means, SVM and Q learning, and PSQ is defined as an improved method based on PCA, k-Means, SVM and Q learning.

The line graph compares the total reward of improved methods with that of classical RL methods over the one hundred of independent experiments. A moving average window is employed to smooth the curve of rewards, enhancing the clarity and interpretability of the results. It is obviously present that SQ, PSQ and SAC have higher performance.

TABLE 3. Performance comparison with the improved method and classical methods.

Method	Average reward	Time complexity
PPO	31.47	7004.77
SAC	49.88	3265.17
Q	31.19	1249.00
DQN	39.36	211.08
PSQ	49.78	1103.13
SQ	49.63	1022.24

Then, for further analysis, the average reward and time complexity are calculated in Table 3. It provides a performance comparison among various methods, with a focus on highlighting the superiority of SQ and PSQ, particularly in comparison to SAC.

In terms of average reward, SAC achieves the highest score of 49.88, outperforming both SQ and PSQ. But the different of average reward among SAC, SQ and PSQ is too small to be noticed. SQ and PSQ can obtained with the same performance as SAC.

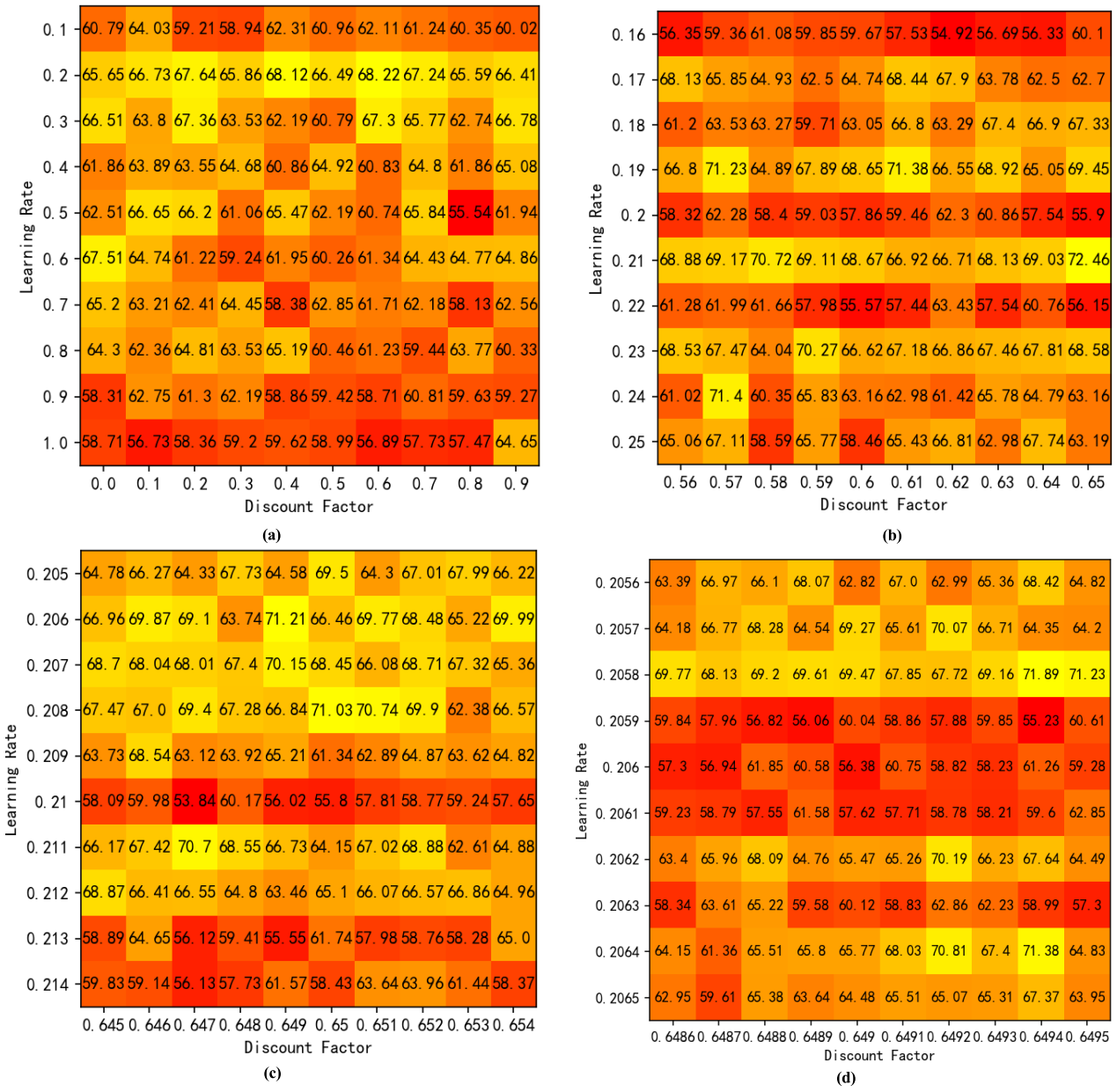

FIGURE 13. Grid search result varying with round. (a) The first round. (b) The second round. (c) The third round. (d) The fourth round.

When considering time complexity, both SQ and PSQ demonstrate significantly lower values compared to SAC. SQ has a time complexity of 1022.24, while PSQ has a time complexity of 1103.13. In contrast, SAC has a higher time complexity of 3265.17. This implies that SQ and PSQ are computationally more efficient and require less processing time compared to SAC.

Overall, SQ and PSQ may achieve the same level of average reward and reward variance as SAC, they offer a distinct advantage in terms of reduced time complexity. This suggests that SQ and PSQ can provide satisfactory performance while requiring less computational resources compared to SAC.

D. HYPERPARAMETERS SELECTION

The purpose of this section is to demonstrate the influence of hyperparameters on the total reward and to illustrate the method of selecting optimal hyperparameters through the analysis of total rewards.

The major characteristic of the proposed method is the adoption of machine learning as the primary framework. This feature reduces the number of hyperparameters that need selection, enhancing the method's robustness. The two key hyperparameters that this method requires adjustment are the learning rate and the discount factor.

To enhance the clarity, the impact of hyperparameters on the method's performance will be presented in a visual

format. The joint analysis of multiple hyperparameters, presenting the relationship between performance and hyperparameters in matrix form, is chosen. The reason is that the objective functions faced by RL methods are typically non-convex, and the trend of average reward based on a hyperparameter is influenced by another hyperparameter. In this method, different learning rates, for example, can lead to different trends in average rewards as the discount factor changes. The specific trend of average rewards with the discount factor under a certain learning rate is not universally applicable. Analysis is aimed at finding the optimal hyperparameters. Benefiting from the limited number of hyperparameters, a grid search method is employed for hyperparameters selection [15], [42].

The optimal values are determined through grid search by exploring potential values of hyperparameters within a specified range. A larger search range and step size is initially utilized to identify potential positions for the optimal values. Subsequently, the range and step size are systematically narrowed down to seek more precise optimal values. The range for the learning rate and the discount factor are (0, 1] and [0, 1), with the initial search step size being 0.1 and each subsequent search step being one-tenth of the previous value. In each search iteration, each hyperparameters pair undergo 10 experiments, and the average reward obtained is averaged to enhance the stability and accuracy of the results. Considering computational burden, the termination criterion for the search is the identification of the first peak that appears with the increasing number of search iterations. More specifically, it is ensured that the number of iterations corresponding to the optimal value is less than or equal to the median of the total search iterations.

The search results are illustrated in Figure 13, corresponding to four rounds of search.

The hyperparameters corresponding to the maximum value in each round of experiments are shown in Table 4.

TABLE 4. Average reward varying with round.

Round	Hyperparameters (learning rate and the discount factor)	Average reward
1	(0.2, 0.6)	68.22
2	(0.21, 0.65)	72.46
3	(0.206, 0.649)	71.21
4	(0.2058, 0.6494)	71.89

From the table, it can be observed that the maximum average reward value is obtained in the second round of searches. If the corresponding hyperparameters is considered as the optimal ones, the number of rounds equals the median of the total search rounds, which is 4. Therefore, following the optimal hyperparameters selection method in this section, choosing the hyperparameters values obtained in the second round of searches as the optimal hyperparameters is reasonable.

Comparing the results of average reward in Section IV-C, it is evident that the average reward obtained after hyperparameters selection is significantly higher than the average reward obtained using typical hyperparameters. Experimental results indicate that, with the selection of hyperparameters using fewer resources, the average reward can be effectively enhanced, further demonstrating the efficiency of this method in policy generation. Moreover, the process is concise and easy to implement in engineering.

V. CONCLUSION

When methods based on RL generate and optimize policy with multi-dimensional continuous observation data, these methods trends to use reduction dimensionality and clustering method, such as PCA and AutoEncoder, and they will face great risk that states may have different state transition process under the same action. To address this challenge, an improved RL method based on UL is proposed. By using feature engineering, the current observation data, the future observation data and reward information are added to form feature data. Then, a low-dimensional representation will be found by utilizing reduction dimensionality and clustering method while retaining MDP transition. The resulting labels are used as the state of policy generation, and the low-dimensional discrete label-state has advantages to solve the computational expensive problems. A detail description of the method flow is provided to ensure the novel of proposed framework can be understood accurately. The effectiveness of the proposed method is validated through two experiments. In the first experiment, the performance of the proposed method compared with that of two baseline methods, AQ and KQ, which are direct combinations of machine learning methods. Evaluation parameters, including average reward and time complexity, are employed for comparison. The results clearly indicate that SQ outperforms the baseline methods in terms of average reward, demonstrating that it not a simple combination of methods but rather a novel approach. Additionally, a comparative analysis of the clustering results obtained by the proposed framework is conducted. The stability and accuracy of the method are evaluated by analyzing the Calinski Harabasz Score curve. The results reveal that the proposed framework consistently delivers stable and reliable performance. In the second experiment, by conducting comparative experiments with state-of-the-art RL methods such as PPO and SAC, the effectiveness and advantages of the proposed framework can be highlighted. The results support the claim that the proposed framework can significantly enhance policy performance and overcome this challenge associated with multi-dimensional continuous observation data. Furthermore, by selecting hyperparameters, the average reward can be effectively enhanced, further demonstrating the efficiency of this method in policy generation.

To guide future research and development efforts in the field, the following studies are recommend as follows. First, while the proposed method demonstrates effectiveness in the domain of specific communication countermeasure devices, its adaptability to other problem domains may be

constrained. Different domains often present distinct properties and requirements, necessitating potential adjustments. An in-depth analysis of the method's adaptability will be conducted to different scenarios. Second, an investigation into the impact of class imbalance on clustering results and its subsequent effects on performance will be conducted. Third, grid search exhaustively explores hyperparameters combinations, which can result in a substantial computational burden, especially when dealing with high precision in hyperparameters. Numerous advanced hyperparameters optimization techniques, such as random search, Bayesian optimization, or evolutionary algorithms, could be used to mitigate this challenge. The implementation of these techniques has the potential to automate hyperparameters tuning, mitigate the impact of class imbalance on decision outcomes, and enhance universality across diverse scenarios. Moreover, as computational resources allow, the performance and scalability of the proposed method will be explored in environments with a larger number of states.

REFERENCES

[1] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, U.K.: MIT Press, 1998.
[2] S. Ravivhandiran, *Hands-On Reinforcement Learning With Python: Master Reinforcement and Deep Reinforcement Learning Using OpenAI Gym and TensorFlow*. Birmingham, U.K.: Packt, 2018.
[3] J.-S. Sheu, S.-R. Wu, and W.-H. Wu, "Performance improvement on traditional Chinese task-oriented dialogue systems with reinforcement learning and regularized dropout technique," *IEEE Access*, vol. 11, pp. 19849–19862, 2023.
[4] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, Mar. 2023.
[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
[6] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, and J. Oh, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Oct. 2019.
[7] X. Chang, Y. Li, Y. Zhao, Y. Du, and D. Liu, "An improved anti-jamming method based on deep reinforcement learning and feature engineering," *IEEE Access*, vol. 10, pp. 69992–70000, 2022.
[8] S. Liu, Y. Xu, X. Chen, X. Wang, M. Wang, W. Li, Y. Li, and Y. Xu, "Pattern-aware intelligent anti-jamming communication: A sequential deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 169204–169216, 2019.
[9] L. Jia, Y. Xu, Y. Sun, S. Feng, and A. Anpalagan, "Stackelberg game approaches for anti-jamming defence in wireless networks," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 120–128, Dec. 2018.
[10] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.
[11] X. Liu, Y. Xu, L. Jia, Q. Wu, and A. Anpalagan, "Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 998–1001, May 2018.
[12] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," 2016, *arXiv:1604.00981*.
[13] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, "Large scale distributed neural network training through online distillation," 2018, *arXiv:1804.03235*.
[14] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimized approximation algorithm in neural networks without overfitting," *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 983–995, Jun. 2008.
[15] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," 2020, *arXiv:2003.05689*.
[16] P. I. Frazier, "A tutorial on Bayesian optimization," 2018, *arXiv:1807.02811*.
[17] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. 25th Annu. Conf. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 1–9.
[18] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu, "Population based training of neural networks," 2017, *arXiv:1711.09846*.
[19] X. Chang, Y. Li, Y. Zhao, Y. Du, D. Liu, and J. Wan, "A scattered wave deceptive jamming method based on genetic algorithm against three channel SAR GMTI," in *Proc. CIE Int. Conf. Radar (Radar)*, Hainan, China, Dec. 2021, pp. 414–419.
[20] A. P. Engelbrecht, *Computational Intelligence: An Introduction*. Hoboken, NJ, USA: Wiley, 2004.
[21] X. Chang, Y. Li, Y. Zhao, and Y. Du, "A multiple-jammer deceptive jamming method based on particle swarm optimization against three-channel SAR GMTI," *IEEE Access*, vol. 9, pp. 138385–138393, 2021.
[22] C. Dong and X. Chang, "A novel scattered wave deception jamming against three channel SAR GMTI," *IEEE Access*, vol. 6, pp. 53882–53889, 2018.
[23] J. Zhang, S. Xing, D. Dai, Y. Li, and S. Xiao, "Three-dimensional deceptive scene generation against single-pass InSAR based on coherent transponders," *IET Radar, Sonar Navigat.*, vol. 10, no. 3, pp. 477–487, Mar. 2016.
[24] J. Zhang, "Study on distributed cooperative jamming techniques against multichannel SAR," Ph.D. dissertation, Dept. Inf. Com. Eng., Natl. Univ. Def. Technol., Changsha, Hunan, China, 2016.
[25] C. Molnar, *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. Victoria, BC, Canada: Leanpub, 2023.
[26] X. Zhang, J. Wang, J. Xu, and C. Gu, "Detection of Android malware based on deep forest and feature enhancement," *IEEE Access*, vol. 11, pp. 29344–29359, 2023.
[27] A. Geron, *Hands-on Machine Learning With Scikit-Learn, Keras, and TensorFlow*. Sebastopol, CA, USA: O'Reilly Media, 2019.
[28] A. A. Patel, *Deep Hands-On Unsupervised Learning Using Python*. Sebastopol, CA, USA: O'Reilly Media, 2019.
[29] R. Zhang, H. Peng, Y. Dou, J. Wu, Q. Sun, Y. Li, J. Zhang, and P. S. Yu, "Automating DBSCAN via deep reinforcement learning," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manag.*, Atlanta, GA, USA, Oct. 2022, pp. 1–2.
[30] N. N. Mohammed, M. Cawthorne, and A. M. Abdulazeez, "Detection of genes patterns with an enhanced partitioning-based DBSCAN algorithm," *J. Inf. Commun. Eng.*, vol. 4, no. 1, pp. 188–195, 2018.
[31] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," *Int. J. Speech Technol.*, vol. 48, pp. 4743–4759, Dec. 2018.
[32] M. Lapan, *Deep Reinforcement Learning Hands-On: Apply Modern RL methods, With Deep Q-Networks, Value Iteration, Policy Gradients, TRPO, AlphaGo Zero and More*. Birmingham, U.K.: Packt, 2018.
[33] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," 2015, *arXiv:1511.06581*.
[34] L. Xiao, *Anti-Jamming Transmissions in Cognitive Radio Networks*. Cham, Switzerland: Springer, 2015.
[35] H. Zhu, C. Fang, Y. Liu, C. Chen, M. Li, and X. S. Shen, "You can jam but you cannot hide: Defending against jamming attacks for geo-location database driven spectrum sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2723–2737, Oct. 2016.
[36] K. Grover, A. Lim, and Q. Yang, "Jamming and anti-jamming techniques in wireless networks: A survey," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 17, no. 4, pp. 197–215, 2014.
[37] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: MIT Press, 2005.
[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[39] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018, *arXiv:1801.01290*.
[40] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," Jan. 2019, *arXiv:1812.05905*.

[41] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," 2017, *arXiv:1702.08165*.
[42] C.-M. Huang, Y.-J. Lee, D. K. J. Lin, and S.-Y. Huang, "Model selection for support vector machines via uniform design," *Comput. Statist. Data Anal.*, vol. 52, no. 1, pp. 335–346, Sep. 2007.

XIN CHANG is currently a Senior Engineer of The 54th Research Institute, China Electronics Technology Group Corporation (CETC54). His main research interest includes electronic science and technology.

YANBIN LI is currently a Researcher of The 54th Research Institute, China Electronics Technology Group Corporation (CETC54). His main research interest includes electronic science and technology.

GUANJIE ZHANG is currently a Senior Engineer of The 54th Research Institute, China Electronics Technology Group Corporation (CETC54). His main research interest includes electronic science and technology.

DONGHUI LIU is currently a Lecturer with the School of Economics and Management, Shijiazhuang Tiedao University. She engaged in the research of complex system analysis.

CHANGJUN FU is currently a Senior Engineer of The 54th Research Institute, China Electronics Technology Group Corporation (CETC54). His main research interest includes electronic science and technology.