Ensemble Reinforcement Learning: A Survey

Yanjie Song[1], Ponnuthurai Nagaratnam Suganthan [2,*], Witold Pedrycz [3,4,5,**], Junwei Ou[1], Yongming He[1], Yingwu Chen[1], Yutong Wu[6]

Abstract

Reinforcement Learning (RL) has emerged as a highly effective technique for addressing various scientific and applied problems. Despite its success, certain complex tasks remain challenging to be addressed solely with a single model and algorithm. In response, ensemble reinforcement learning (ERL), a promising approach that combines the benefits of both RL and ensemble learning (EL), has gained widespread popularity. ERL leverages multiple models or training algorithms to comprehensively explore the problem space and possesses strong generalization capabilities. In this study, we present a comprehensive survey on ERL to provide readers with an overview of recent advances and challenges in the field. Firstly, we provide an introduction to the background and motivation for ERL. Secondly, we conduct a detailed analysis of strategies such as model selection and combination that have been successfully implemented in ERL. Subsequently, we explore the application of ERL, summarize the datasets, and analyze the algorithms employed. Finally, we outline several open questions and discuss future research directions of ERL. By offering guidance for future scientific research and engineering applications, this survey significantly contributes to the advancement of ERL.

Keywords: ensemble reinforcement learning, reinforcement learning, ensemble learning, artificial neural network, ensemble strategy

1. Introduction

Over the past several decades, reinforcement learning (RL) methods have proven to be highly effective in solving complex problems across various fields, including gaming, robotics,

*Corresponding author
**Corresponding author
Email addresses: songyj_2017@163.com (Yanjie Song), p.n.suganthan@qu.edu.qa (Ponnuthurai Nagaratnam Suganthan), wpedrycz@ualberta.ca (Witold Pedrycz), junweiou@163.com (Junwei Ou), heyongming10@hotmail.com (Yongming He), ywchen@nudt.edu.cn (Yingwu Chen), wuyutong119@gmail.com (Yutong Wu)
[1]College of Systems Engineering, National University of Defense Technology, Changsha, China
[2]KINDI Center for Computing Research, College of Engineering, Qatar University, Doha, Qatar
[3]Department of Electrical & Computer Engineering, University of Alberta, Edmonton AB, Canada
[4]Systems Research Institute, Polish Academy of Sciences, Poland
[5]Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Turkiye
[6]Department of Analytics, Operations and Systems, University of Kent, UK

and computer vision. With the emergence of breakthroughs such as deep Q neural networks [1], AlphaGo [2], video games [3, 4], and robotic control tasks [5], RL has witnessed a revitalization that outperforms human performance. The success of this approach is attributed to the agent's ability to automate feature acquisition and accomplish end-to-end learning. Artificial neural networks (ANN) and gradient descent further enhance RL's exploration and exploitation capabilities, rendering it suitable for handling laborious manual work or challenging tasks.

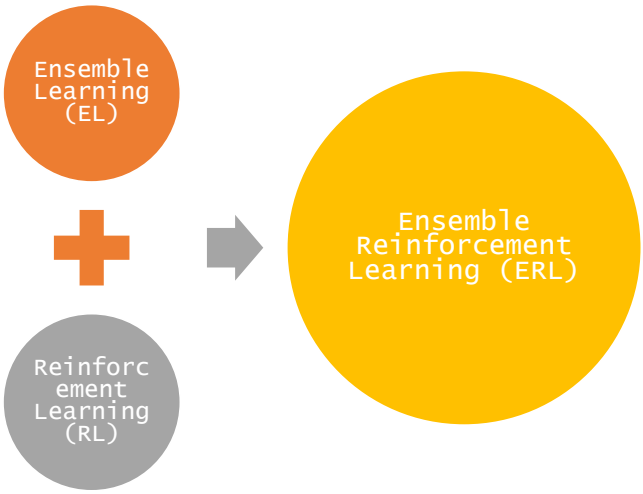


Figure 1: Components of the ERL method. The two components EL and RL are combined to form ERL.

Nevertheless, each type of RL possesses distinct advantages and limitations. For instance, deep reinforcement learning (DRL) requires extensive training to obtain a policy [4], thereby introducing additional challenges such as overfitting [6], error propagation [7], and imbalance between exploration and exploitation [8]. These challenges motivate researchers to design models or training algorithms. One approach is implementing ensemble learning (EL) into the RL framework, which enhances algorithmic learning and representation abilities (see Figure 1). The method known as ensemble reinforcement learning (ERL) has demonstrated exceptional performance across various applications. The concept of EL was initially exemplified by Marquis de Condorcet [9], who showed that average voting outperforms individual model decisions. The subsequent studies conducted by Krogh and Vedelsby [10], Breiman [11], and other researchers have theoretically demonstrated the significant advantages of ensemble methods from various perspectives. The success of ensemble methods in the field of deep learning (DL) and RL can be attributed to three factors: the decomposition of datasets [12], powerful learning capabilities [13], and diverse ensemble methods [11].

The ERL method can be categorized according to different criteria. The constituent elements allow for the classification of ERL into high-level ensembles [14] and low-level ensembles [15]. ERL can also be classified as single-agent ERL [16] and multi-agent ERL [17] based on the number of agents involved. Moreover, centralized ERL [18] and distributed ERL [19] are classifications of ERL based on how the agents work. Figure 2 presents a taxonomy according to agent cooperation and method deployment. All of these taxonomies

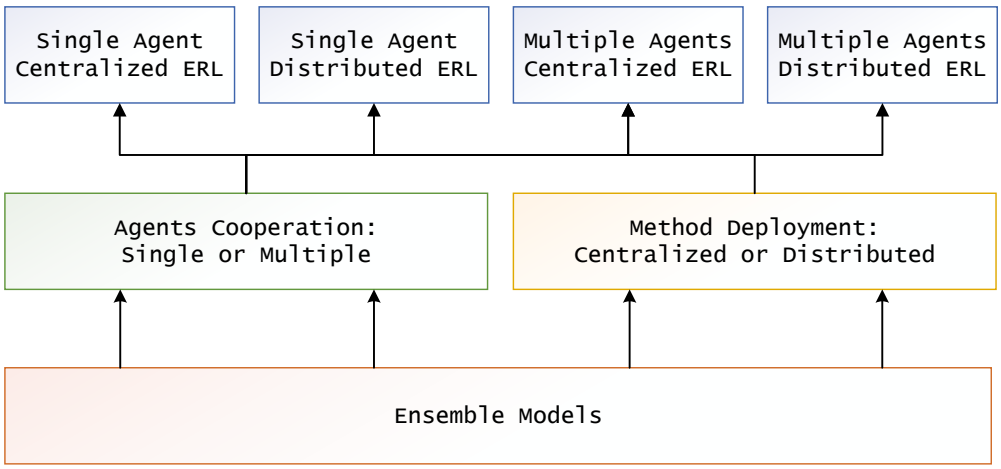


Figure 2: A taxonomy of ERL according to agent cooperation and method deployment

are reasonable and can serve as reference frameworks for designing new ERL methods. The utilization of existing frameworks enables researchers to rapidly develop novel ERL methods. Additionally, comprehending the impact of strategies can assist researchers in directing their focus toward specific strategy design. In this paper, we provide a detailed description of ERL methods according to the improvement strategies used and discuss their applications to guide the design of new methods.

The literature on ERL encompasses a broad spectrum of related work, including training algorithms, ensemble strategies, and application domains. This paper aims to provide readers with a systematic overview of the existing research, current progress, and valuable conclusions achieved in this field. **To the best of our knowledge, this is the first survey focusing solely on ensemble reinforcement learning.** In this survey, we present the strategies employed in ERL and its associated applications, deliberate upon various unresolved inquiries, and offer a roadmap for future exploration in the realm of ERL. Approaching from this perspective enables readers to swiftly comprehend the ERL methodology while also facilitating the design and enhancement of tailored ERL approaches for specific problems or application scenarios.

The remainder of this paper is structured as follows. Section 2 presents the background of ensemble reinforcement learning methods. Section 3 introduces implementation strategies in ERL. Section 4 discusses the application of ERL to different domains. Section 4 discusses the datasets and compares methods used in the ERL-related studies. Section 5 discusses several open questions and possible future research directions. Section 7 gives the conclusion of this paper. (See Figure 3).

2. Background

To enhance readers' comprehension of ensemble reinforcement learning methods, this section presents a concise overview of RL, EL, and ERL.

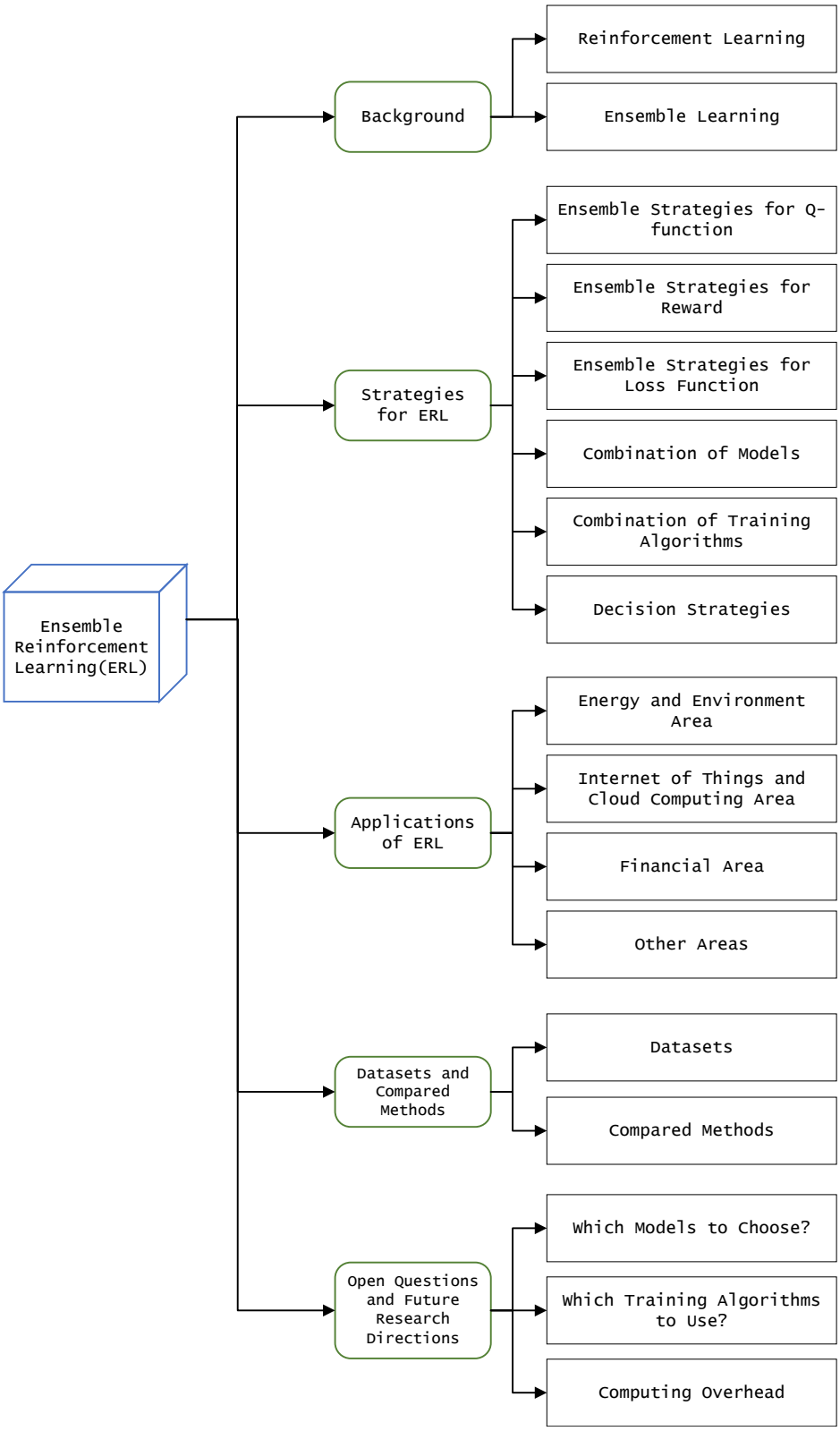


Figure 3: Structure of the survey

2.1. Reinforcement Learning

Reinforcement learning is an artificial intelligence method in which an agent interacts with an environment and makes decisions iteratively to rectify errors, aiming to achieve optimal decision-making. Agent, the core of RL, is an entity that is capable of sensing the environment, making decisions, and taking actions. Besides, the Markov Decision Process (MDP) forms the foundation for using RL to solve problems [20]. The RL approach is applicable when an agent's decision-making process is only related to the current state and not to the previous state. Figure 4 illustrates the agent-environment interaction process. A tuple $\langle S, A, P, R, \gamma \rangle$ can represent the MDP, where S denotes the state, A denotes the action, $P : S \times A \to P(S)$ denotes the state transfer matrix with the probability value $p(s' \mid s) = p(S_{t+1} = s' \mid S_t = s)$, $R : S \times A \to \mathbb{R}$ denotes the reward function, and $\gamma \in [0, 1]$ denotes the discount factor. The agent's state at time step t is s_t, and it will take the action a_t. The policy π is defined by the combination of all states and actions, while the Q-value evaluates the expected reward obtained by the agent following policy π.

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s, a_0 = a\right] \tag{1}$$

The objective of using RL methods is to find an optimal policy π that maximizes Q^{π}. For finite-state MDPs, Q-learning is the most prevalent RL method [21], which uses a Q-table to record the combinations of ⟨state,action⟩. Subsequently, several RL methods incorporating artificial neural networks have been proposed to cope with the infinite state space.

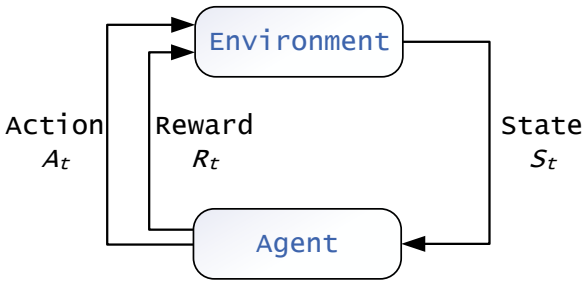


Figure 4: Interaction process between agent and environment. The Agent's performance is updated on the state through environmental evaluation.

Training algorithms can be categorized into model-based RL and model-free RL according to whether the environment model in RL is pre-defined or acquired through learning. Furthermore, these training algorithms can also be classified according to state-based, policy-based, or state-policy combination approaches. A more comprehensive account of the research progress on RL can be found in [22].

The RL methods differ distinctly from the other classical classes of ML methods, namely supervised learning (SL) and unsupervised learning (UL), in several aspects. UL involves training a model using labeled datasets to enable the algorithm to predict accurate output labels based on input data. SL is primarily employed for regression and classification

tasks. UL utilizes unlabeled data for model training to discover patterns, structures, or relationships within such data. Dimensionality reduction and clustering are representative UL techniques. As depicted in Table 1, these three types of methods exhibit significant differences across all three dimensions: data type used, feedback mechanism for the result, and target.

Table 1: Differences between ERL, SL, UL

Dimension	SL	UL	RL
Data type used	labeled	unlabeled	unlabeled
Feedback mechanism for results	direct feedback	no feedback	multi-step post-implementation feedback
Target	reduce error	find the hidden relationship	search the strategy with long-term reaward

2.2. Ensemble Learning

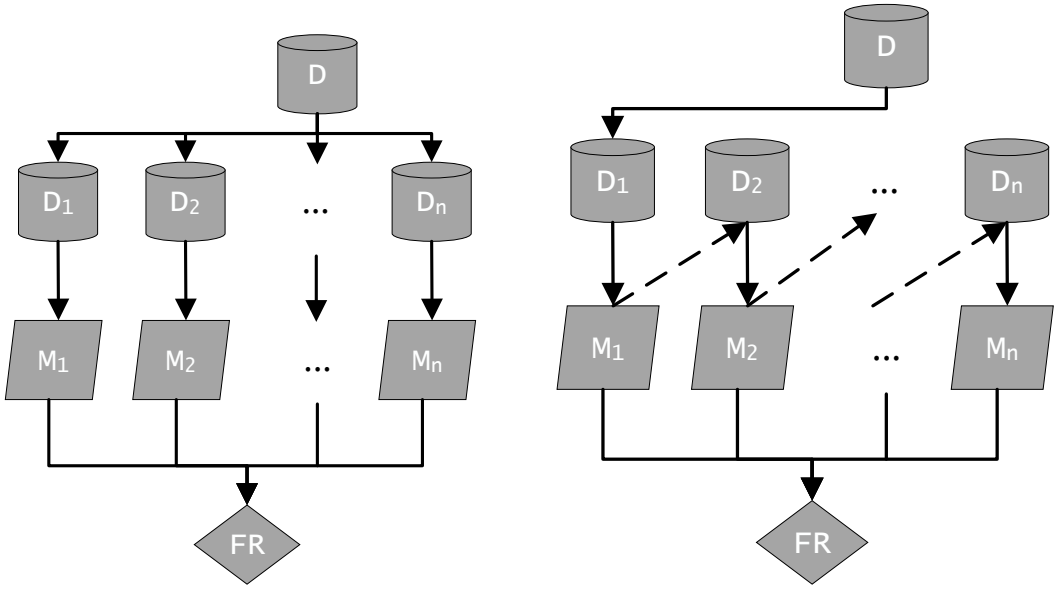
Ensemble learning (EL) is a widely adopted approach in the field of machine learning (ML). The fundamental concept behind EL methods involves training multiple predictors, combining their outputs, and aggregating them to make informed decisions as the final result of an ensemble model. Compared to individual basic models, this EL method effectively leverages the distinctive characteristics of diverse model types to enhance the predictive performance and achieve more robust results. Prominent approaches in ensemble learning encompass bagging [23], boosting [24], and stacking [25]. Figure 5 gives a schematic diagram of these three types of EL methods, where D denotes the dataset, D_1 to D_n denote the sample selection from the dataset, M_1 to M_n denote the models employed, and FR denotes the final result. The dotted line in Figure 5-(b) indicates the dynamic nature of sample weights across subsequent iterations of the dataset. The dotted line in Figure 5-(c) indicates that all datasets are used for model prediction from level_2 to level_L. The primary distinction among these three types of methods lies in the approach to sample selection. These original and improved EL methods have been extensively employed across diverse domains, with the incorporation of domain knowledge in the improved EL method yielding exceptional performance outcomes. In summary, the EL method has demonstrated its advantageous nature through three key aspects.

• **Bias–variance Decomposition**

The bias-variance decomposition has been widely employed to demonstrate the effectiveness of ensemble learning (EL) methods over individual learning methods. While bagging reduces variance among base learners, other EL methods aim to reduce both bias and variance. Krogh and Vedelsby initially demonstrated the effectiveness of EL for single data set problems by employing ambiguity decomposition to reduce variance [10]. Subsequently, Brown et al. [26] and Geman et al. [27] verified the effectiveness of EL methods for multiple data set problems. The decomposition equation can be formulated as follows [12]:

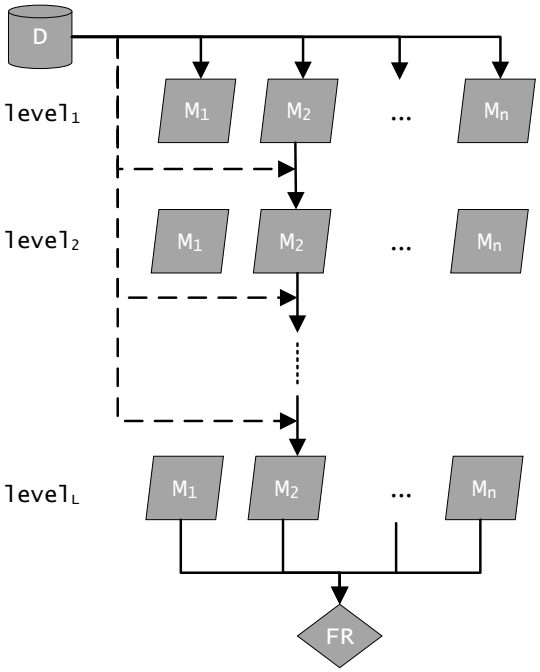
$$E[s-t]^2 = bias^2 + \frac{1}{N}var + (1-\frac{1}{N})covar \tag{2}$$

$$bias = \frac{1}{N}\sum_i (E[s_i]-t) \tag{3}$$



(a) Bagging [23]

(b) Boosting [24]



(c) Stacking [25]

Figure 5: Schematic diagram of three types of EL methods. (a) shows bagging. (b) shows boosting. (c) shows stacking.

$$var = \frac{1}{N}\sum_{i} E[s_i - E[s_i]]^2 \tag{4}$$

$$covar = \frac{1}{N(N-1)}\sum_{i}\sum_{j\neq i} E[s_i - E[s_i]][s_j - E[s_j]] \tag{5}$$

where i denotes the i-th model of EL, s denotes a solution to the problem, and N denotes the number of models in EL. The bias and variance are obtained using the average differences among multiple models, while *covar* measures the pairwise difference between models in the EL method.

The reduction in bias for an individual model is accompanied by an increase in variance. However, the ensemble model can be used for prediction purposes and effectively mitigate variance without increasing bias.

• **Statistical Perspective**

The advantages of EL from a statistical perspective are supported by the work conducted by Dietterich [13]. From a statistical point of view, machine learning problems exist within a search space encompassing multiple hypotheses. The target of the prediction model is to identify the optimal hypothesis. However, due to limited training data size relative to the expansive search space, there is an elevated risk of erroneous inferences. The use of an EL method can effectively integrate these hypotheses to enhance comprehension of the search space characteristics and mitigate the likelihood of erroneous classification or invalid prediction.

• **Diversity Perspective**

The advantages of EL from the diversity perspective are readily comprehensible and easily graspable. Dietterich highlights that the combination of different single models can enhance diversity [13]. Some typical EL methods, such as AdaBoost and random forest, show the importance of diversity in terms of training data. And the use of random noise can enhance the richness of the output. In other words, diversity allows decision-makers to combine the model output with usage requirements to obtain a more reasonable final result.

The integration of DL with EL, known as ensemble deep learning (EDL), has gained significant popularity in recent years. EDL demonstrates strong predictive capabilities by employing a model training approach that combines deep artificial neural networks (ANNs) and gradient descent [28, 29]. A comprehensive overview of the latest advancements in EDL can be found in [29]. It is noteworthy that ERL assesses the efficacy of model training based on environmental factors, whereas EDL relies on real-world datasets, emphasizing the distinction between ERL and EDL.

2.3. Ensemble Reinforcement Learning

Ensemble reinforcement learning is a new artificial intelligence method that integrates reinforcement learning training methods into the ensemble learning framework, replacing conventional model training methods with more sophisticated RL methods. The training process of ERL involves a bidirectional exchange of data between the model and the training method. Figure 6 illustrates the data flow between the ensemble model and reinforcement learning. The base learners in the ensemble model generate predictive data based on the task

inputs, which interact with the environment and serve as the data source for the RL training. Once RL meets the training criteria, it consumes the data, leading to the generation of a new set of model parameters that are subsequently used to update the model. With the continuous generation and consumption of data, the ensemble model can effectively identify an optimal combination of parameters that can be suitably adapted to the environment. Then, the ERL method can directly leverage the trained ensemble model to solve complex tasks.

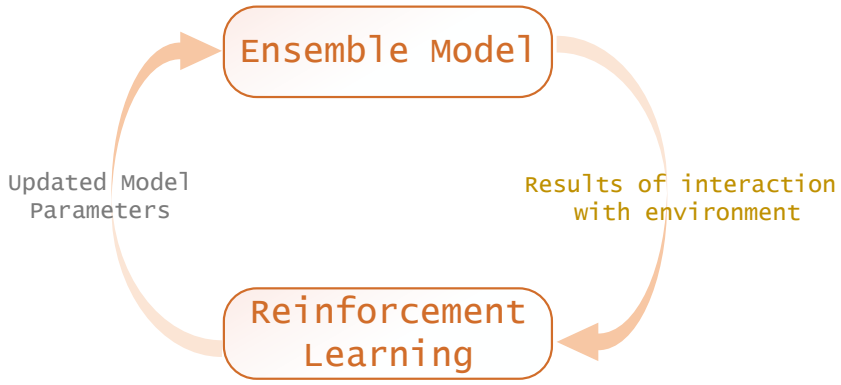


Figure 6: Data flow in ERL method. The information interaction will be continuous between RL and EL.

The structure of the ERL method exhibits a higher level of complexity compared to that of single EL or RL methods, thereby providing greater potential for enhancing method performance from various perspectives. In the next section, we will describe the improvement strategies in the ERL method in detail.

3. Strategies for Ensemble Reinforcement Learning

Previous studies have shown that the ERL method demonstrates superior average performance and sampling efficiency compared to RL methods, as evidenced by results obtained from public RL test sets and practical tasks [15, 30]. By using ERL, the performance improvement can reach up to 20% [31, 32, 33]. Moreover, for classification tasks, the ERL method achieves the best accuracy scores across multiple benchmarks in the UCI online data repository [34, 35].

The strategies employed by ERL to outperform other solution methods in various problems are closely interconnected. These strategies are categorized based on the diverse improvements made to the ERL. The ensemble strategies for ERL encompass Q-function, reward, and loss function ensembles, as well as model combinations, combination training algorithms, and decision strategies. In this section, we will introduce these strategies individually.

3.1. Ensemble Strategies for Q-functions

In most RL methods, the Q-function reflects how good the agent is in any given state [20]. A "good state" refers to a state that can achieve a high expected return, which is

contingent upon the action taken by the agent. The background section provides the Q-function formula, which is applicable to ERL methods. Additionally, tailoring Q-functions specifically for ERL methods can further enhance the algorithm performance [15, 36, 37, 38].

A Max-min Q-learning algorithm using multiple Q-functions was proposed by Lan et al. [39] to evaluate the performance. The max-min mechanism integrates multiple predicted values as a reference for agent decision-making. Specifically, the prediction term in the original Q-value calculation formula is determined by selecting the smallest value among multiple Q functions. The efficacy of this algorithm is demonstrated using the Mountain Car environment. It can be observed from the results that enhancing the Q-function positively impacts both the convergence performance and search efficiency of the algorithm.

The Q-function can also be enhanced through the utilization of Bayesian optimization. Chen et al. incorporated this concept into the design of the ERL method to update Q^* using Bayesian optimization, which proves particularly effective in addressing high-dimensional ERL problems and especially valuable when dealing with ultra-large solution spaces [8]. In this study, an upper-confidence bounds (UCB) based strategy for exploring the solution space is employed for action selection by the agent. The proposed method's performance is evaluated using Atari games in the experimental section, and the results validate its effectiveness.

The ERL methods can leverage certain Q-value approximation techniques employed in RL research. Ghosh et al. proposed an ERL approach based on a multi-agent framework to address the air traffic control problem [40]. To expedite convergence, they utilized a kernel-based Q-value approximation method that utilizes sample transitions [41].

The main targets of Q-function improvements can be summarized as follows: maintaining diversity [15, 37], enhancing algorithmic exploration performance [8], and reducing bias (e.g., underestimation bias) or coping with the effects of overestimation [39]. This improvement strategy can be regarded as a key optimization after the integration of multiple single models, which allows the components to be integrated as a whole. This improvement strategy is more thorough and easier to get high-quality results than some other improvement strategies.

3.2. Ensemble Strategies for Reward

The reward is a reflection of the agent's performance in taking actions based on the state. Generally, a high reward corresponds to a good decision, while a problematic decision prompts the agent to identify and rectify errors through the reward mechanism. Building upon this concept, Yao et al. proposed an averaging reward calculation method for the ERL method, enabling it to effectively balance exploration and exploitation [42]. Subsequently, an ANN model is trained using the soft actor-critic method. This ERL approach proves highly suitable for addressing challenges associated with exploring uncharted regions.

The combination of reward functions in ERL can also be utilized with weight aggregation. Lin et al. proposed an adaptive adjustment method for the weights of reward functions by combining Upper Confidence Bounds (UCB) and error [43]. This weight update strategy enables the ERL method to assess the accuracy of previous policies and enhance generalizability. Qi et al. also employed an ERL method with aggregated weighted reward functions to address the traffic signal control problem [44].

Although the traditional calculation method of reward is widely used, it has the shortcomings of a complex process. Compared to traditional methods, fuzzy-based methods can reduce computational costs. A fuzzy set can affect the reward value obtained by agents by measuring dissimilarity. Pan et al. proposed a dissimilarity evaluation metric for deciding the weight value of each agent's reward in ERL [45]. In this way, ERL can achieve a good training effect with fewer iterations.

Strategies for improving reward can modify the agent's action evaluation mechanism, thereby impacting both the state and adopted strategies. While most existing research focuses on processing feedback base learners in the environment and aggregating it to reward, limited studies specifically address comprehensive evaluation of differentiated performance among various base learners in ERL [42, 45]. Furthermore, designing novel methods for calculating rewards in ERL is also an improvement idea.

3.3. Ensemble Strategies for Loss Function

The loss function serves as a fundamental foundation for ERL to update network parameters and enhance the performance of agent decisions. A smaller loss value indicates a closer proximity between the predicted value of the ERL model and the actual value. However, during RL model training, two critical issues often arise: gradient explosion and gradient disappearance. Several studies in ERL have endeavored to refine the accuracy of RL model decisions by enhancing the loss function [15, 46, 47, 48]. Kumar et al. conducted theoretical analysis on bootstrapping error and proposed an approach to reduce error accumulation, thereby augmenting stability in ensemble Q-learning algorithms [7].

Designing a global loss function for all models used is another approach specific to ERL. Adebola et al. proposed an improved global loss function with each member model included in the function [49]. Moreover, an interpolation method is used to control the difference between policies that the algorithm needs to train. This ERL method can also optimize the agent's policy selection using fine-tuning techniques. Based on this idea, Jiang et al. added the training data error between models to the overall loss calculation formula for improving prediction accuracy [19]. It can be seen from the experiment that this method is applicable in mobile edge computing (MEC) systems for rational resource scheduling.

In addition, incorporating uncertainty into the analysis is an effective approach to enhance the loss function. Sun et al. employed the technique of uncertainty reduction to devise an ensemble loss function [50], which effectively mitigates the risk of the RL model getting trapped in a local optimum. Within this study, a distillation method is utilized for selecting training for the model. Subsequently, the performance of the proposed ERL method is validated through Atari game experiments.

The improvement of the loss function can make the model prediction of ERL closer to the real situation [20]. However, due to the existence of bias and variance, it is not guaranteed that a high-precision model must guarantee excellent performance in processing tasks. This is particularly true for complex sequential decision problems where environmental changes significantly impact the agent's performance. Therefore, designing a robust loss function that ensures stable performance across various scenarios becomes crucial when considering further advancements in improving the ERL method.

3.4. Combination of Models

The ensemble of different types of models is a common and simple strategy in ERL. The combination of models can achieve structural diversity. These models can be either ML models or ANN models [51, 52]. The structure of the model combination is determined according to the specific problem and the solution target. A single type or a combination of multiple types of models are both popular. For using only one type of model, ANNs with different depths can be considered. While other studies use different random initialization strategies [16] or ANN in different training stages [53]. Table 2 provides a summary of related work ensemble model combination strategies. There are three models mainly in relevant works, including ML models only, ANN models only, and ML&ANN models hybrid.

The model selection process typically involves choosing from a range of classical or recently proposed methods that are specifically designed for addressing this type of task. For instance, when it comes to prediction tasks, most researchers will choose convolutional neural networks, gated recursive units, artificial neural networks (ANN), and other SOTA approaches in this domain [52, 54, 55]. Apart from determining the potential composition of ERL, it is also crucial to determine the number of base learners. In most studies, two or three single models are commonly employed. Besides, there exist some studies where more than three models have been integrated within the ERL framework [56]. Saadallah et al. [57], Li et al. [58], and Sharma et al. [59] are some examples in this regard. These studies have extensively demonstrated that employing a combination of strategies in ERL outperforms baseline methods as well as state-of-the-art ensemble learning techniques [33, 60].

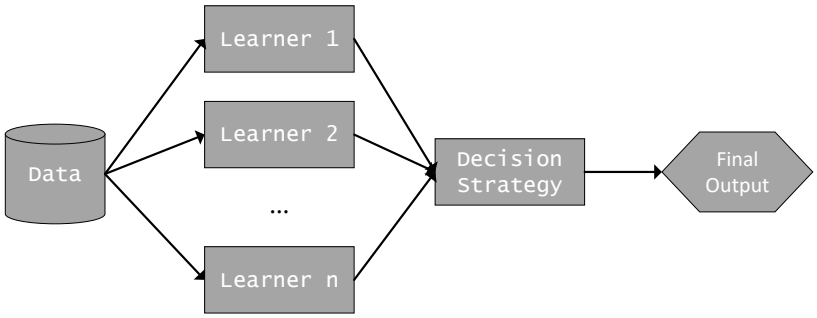


Figure 7: Parallel ensemble reinforcement learning. The respective results of base learners are processed by the decision strategy to get the final output.

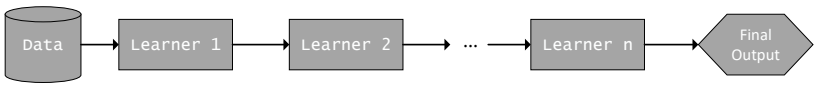


Figure 8: Sequential ensemble reinforcement learning. The output of the previous base learner will be used as the input of the following base learner.

ERL can be divided into parallel ERL and sequential ERL according to the relationship between base learners in ERL. Figure 7 and Figure 8 give schematic diagrams of these two ERL methods. In most ERL studies, such as Liu et al. [35], Schubert et al. [65], and Shen

Table 2: Combination of models

Year	Author	Combination of Models
2019	Dong et al. [51]	long short-term memory (LSTM) network, gated recurrent unit network
2019	Goyal et al. [48]	convolution neural network (CNN), gated recursive unit
2020	Liu et al. [61]	LSTM network, deep belief network, echo state network
2020	Perepu et al. [52]	linear regression model, LSTM model, ANN, random forest
2021	Liu et al. [54]	graph convolutional network, LSTM networks, gated recursive unit
2021	Saadallah et al. [57]	autoregressive integrated moving average, exponential smoothing, gradient boosting machines, gaussian processes, support vector regression, random forest, projection pursuit regression, MARS, principal component regression, decision tree regression, partial least squares regression, multilayer perceptron, LSTM network, Bi-LSTM: Bidirectional LSTM, CNN-based LSTM, convolutional LSTM
2021	Daniel L. Elliott and Charles Anderson [55]	CNN, gated recursive unit, ANN
2022	Shang et al. [32]	gated recursive unit, graph convolutional network, graph attention network
2022	Tan et al. [33]	graph attention network, long short-term memory networks, temporal convolutional network
2022	Li et al. [62]	gated recurrent unit, deep belief network, temporal convolutional network
2022	Zijie Cao and Hui Liu [56]	temporal convolutional network, Bidirectional long short-term memory network, kernel extreme learning machine
2022	Birman et al. [63]	machine learning models, ANN
2022	Li et al. [58]	naive bayes, support vector machine with stochastic gradient descent, FastText, Bi-directional LSTM
2022	Sharma et al. [59]	support vector regressor (SVR), eXtreme gradient boosting (XGBoost), Random Forest (RF), ANN, LSTM, CNN, CNN-LSTM, CNN-XGB, CNN-SVR, and CNN-RF
2022	Shi Yin and Hui Liu [64]	group method of data handling, echo state network, extreme learning machine
2023	Yu et al. [31]	graph attention network, gated recursive unit, temporal convolutional network

et al. [66], base learners are constructed in a parallel framework. These ML or ANN models are responsible for the same task. After each model processing, the final prediction result will be generated by a certain strategy. There are also some studies, such as Qin et al. and Ferreira et al., that try to construct the ERL method in the sequential framework [67, 68]. In this framework, the base learner completes the final prediction step by step in a certain order [68].

Model combination is a readily implementable strategy for enhancing ERL performance. By integrating multiple classical or advanced models, the diversity of ERL methods can be maintained while leveraging the strengths of each model [55, 59]. However, it does not necessarily follow that increasing the number and variety of models in ERL will always lead to improved performance [69]. The single models and the design of decision mechanisms significantly impact final outputs. Moreover, due to the numerous parameters involved, training these models requires substantial data and time. Therefore, achieving a balance between the number of models employed, performance enhancement, and training costs becomes a key consideration when adopting the model combination strategy.

3.5. Combination of Training Algorithms

Table 3: Combination of training algorithms

Year	Author	Combination of Training Algorithms
2008	Marco A. Wiering and Hado van Hasselt [70]	Q-learning, Sarsa, actor-critic, QV-learning, ACLA
2018	Chen et al. [71]	deep Q-networks, deep Sarsa networks, double deep Q-networks
2020	Yang et al. [14]	proximal policy optimization, advantage actor-critic, deep deterministic policy gradient
2020	Saphal et al. [72]	advantage actor-critic, sample efficient actor-critic with experience replay, actor-critic using Kronecker-factored trust region, deep deterministic policy gradient, soft actor-critic, trust region policy optimization
2021	Smit et al. [30]	double deep Q-Learning, soft actor-critic
2022	Eriksson et al. [73]	residual gradient, TD, TD(λ)
2022	Németh, Marcell and Szűcs, Gábor [74]	deep deterministic policy gradient, advantage actor-critic, proximal policy optimization

Ensemble Reinforcement Learning (ERL) can not only use model combinations to obtain diverse prediction results but can also use different training algorithms to achieve full exploration of the solution space. Parameter diversity can be achieved by using multiple training algorithms to get the respective parameters of base learner. Training algorithms can be classified into three categories: state-based, policy-based, and state-policy combination-based. Each of these training algorithms has its unique sampling strategy and output data

characteristics. Researchers can quickly use the training algorithms according to the application scenarios without focusing on data sampling technology, which is similar to the EL method [70, 74]. Table 3 provides information about studies using the combination strategy of training algorithms. There exists a new method of combining online and offline training algorithms or using training algorithms based on different optimization strategies, which can take advantage of their respective strengths to handle complex tasks. Accordingly, the complexity of ERL methods using such improved strategies increases. For this reason, the training process of the ERL method takes more time. Moreover, the ensemble model obtained also requires the design of a decision strategy to select the prediction results that are closest to the actual situation.

Currently, the research related to the combination of training algorithms is not deep enough and simply combines multiple typical algorithms. The combination of training algorithms in the ERL, similar to the model combination strategy, increases parameters (primarily hyperparameters). However, intricate combinations of training algorithms can result in significant time investment for implementation and debugging [75]. Moreover, excessive hyperparameters may lead to models lacking robustness across different environments [76]. It is worth noting that there are some similarities between training algorithms. Transfer learning can be considered to transfer sampled data to reduce the training time. In addition, the termination conditions of multiple training algorithms also deserve in-depth analysis. Using the same number of iterations may result in some models finding the best policy long ago, while some models still need further training. Therefore, more research is required to investigate the combination of training algorithms in ERL.

3.6. Decision Strategies

The ERL algorithm employs multiple base learners to generate individual results which may introduce variations among the outputs. Consequently, specific decision strategies are necessary for the ERL model and output. Commonly adopted decision strategies in existing relevant research encompass voting, optimal combination, binning, aggregation, weighted aggregation, stacking, and Boltzmann multiplication (refer to Figure 9).

Voting: Voting, as a common ERL decision strategy, records the number of occurrences of each prediction at first [77]. Then, the final prediction can be selected from the results according to the principle of majority or ranking.

Optimal Combination: For classification problems, this is a commonly used decision strategy [78]. Multiple base classifiers are trained separately, from which the optimal subset of models is selected to form an ensemble to classify the test set.

Binning: Binning is a majority voting decision strategy with continuous action space [72]. First, the action space is discretized into multiple intervals. Then, the number of occurrences of the actions in each interval is recorded. Finally, the average value of the action within the interval with the highest number of occurrences is selected as the final prediction result.

Aggregation: The prediction results of all the models in ERL are summed to produce an overall evaluation value, which is taken as the final result [58]. In the aggregation method, each model is considered to be equally reliable.

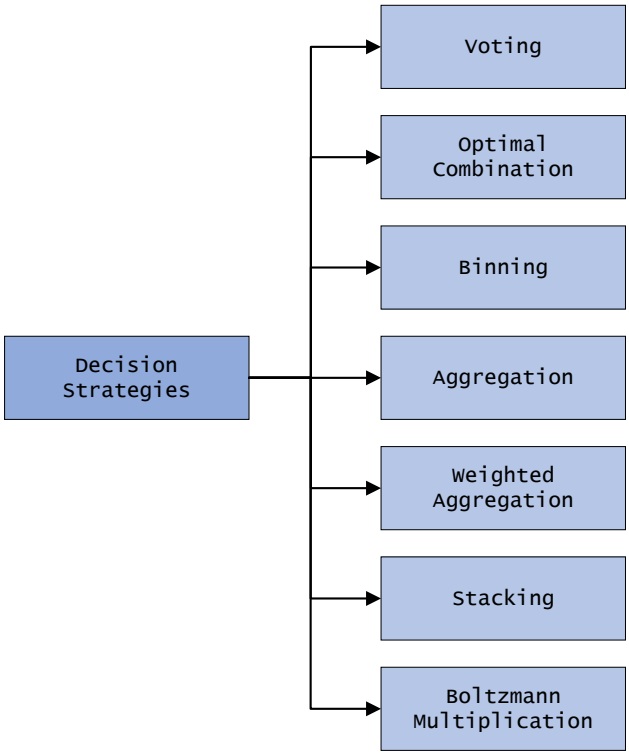


Figure 9: Decision strategies. Decision strategies mainly include: voting, optimal combination, binning, aggregation, weighted aggregation, stacking, and Boltzmann multiplication.

Weighted Aggregation: The prediction results obtained from different models are summed according to their weights [52]. A high value of weight is used for aggregation models with high prediction accuracy.

Stacking: First, an additional machine learning model is involved to further predict the results of base learners. Then, the output of this machine learning model is used as the final prediction result [63].

Boltzmann Multiplication: Boltzmann distribution is the basis for decision making [70]. The probability value of each action can be calculated according to the Boltzmann distribution. The outcome with the highest probability value is selected and will not be changed.

In summary, the selection of decision strategies in ERL depends on the specific application scenario and the characteristics of the base learners. At present, there is no related study on the in-depth analysis of the application scenarios of these strategies. The study in this area will be helpful to improve the prediction accuracy of ERL methods. It will also promote the process of ERL research.

3.7. Discussions

In recent years, a variety of improvement strategies have emerged to enhance the performance of ERL. These include adjusting the Q-function, reward, and loss function in RL, designing model combinations, training algorithm combinations, and decision-making mechanisms. These strategies have improved ERL's ability to solve complex tasks from various aspects and promoted its application in different fields. However, the rapid development of this method has exposed a deficiency in insufficiently in-depth theoretical analysis of ERL improvement strategies. Most studies only focus on improving ERL from an application perspective without analyzing the motivation for adopting these strategies. This situation may lead to numerous similar works that do not advance the field well. Therefore, we aim to discuss two important issues here with hopes of drawing attention from other researchers.

The first problem is how to select strategies to improve the ERL. Irrespective of the method employed for strategy selection, the objective remains consistent - improving ERL performance for task-solving purposes. These strategies vary significantly in terms of the effort required to adapt the approach due to varying levels of ERL modifications involved. For instance, proposing a new Q-function or loss function necessitates substantial knowledge and experience in RL and ERL methods. This strategy design approach demands considerable time and effort from researchers. However, it offers an opportunity to apply the newly proposed ERL across a range of problem scenarios or tasks, thereby driving research development forward. Many researchers facing challenges when directly designing new mechanisms for RL find it easier to combine or enhance existing techniques while applying them to novel problems. Although they may encounter difficulties when attempting to direct in other areas, this type of research effectively addresses their specific interests. There is an urgent need for systematic analysis of how different strategies impact ERL improvement and obtain applicable improvement strategies across various scenarios or problems. Such systematic analysis will foster rapid development and widespread implementation of ERL

within diverse fields. Additionally, analyzing the effects of multiple strategy combinations is essential in order to fully realize the potential of ERL.

Another question to consider is whether more complex structures are superior. Not only the ERL field but also other ML fields exhibit a trend towards increasingly intricate models and algorithms. While complex structures can indeed enhance method performance, they inevitably lead to greater computational resource consumption [37]. Additionally, the complexity of modeling structures may result in numerous hyperparameters requiring researchers to train effective models. Furthermore, these models may experience high volatility in performance when applied to different scenarios [76]. Therefore, it is crucial to design a sound strategy that achieves desired goals by analyzing the impact of increasing model numbers and types integrated into ERL method performance enhancement. Researchers should focus more on ERL strategy design if an appropriate number of models are available.

4. Applications of Ensemble Reinforcement Learning

A significant portion of existing research on ERL primarily focuses on discrete/continuous control actions [43, 79, 80, 81] and game environments [82, 83, 84] to verify the effectiveness of proposed algorithms. Additionally, researchers attempted to utilize ERL methods to solve in practical domains. Figure 10 provides an overview of the key application areas encompassing energy and environment, IoT and cloud computing, finance, and other sectors where ERL has been extensively explored. Among these domains, energy and environment emerge as the most extensively investigated area for applying ERL techniques. In this section, we discuss the application of ERL in various domains.

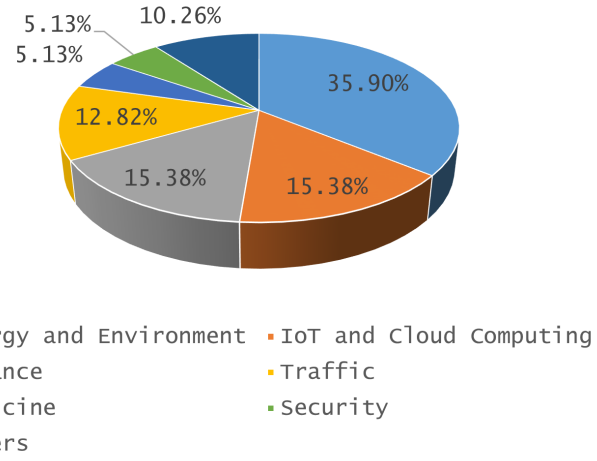


Figure 10: The summary of the different application areas of ERL. The area where ERL is most applied is energy and environment.

4.1. Energy and Environment Area

As the global economy continues to expand, energy and environmental issues have gained increasing attention worldwide. The utilization of neural network methods for predicting

future conditions based on historical data has become the policy formulation. In this type of prediction problem, there exists a spatiotemporal relationship between data, which makes the recurrent neural network (RNN) the preferred choice. In these related studies, ensemble RNN models (ERL) are employed to obtain. Table 4 presents recent applications of ERL in the field of energy and environment. It is evident that wind power and PM 2.5 prediction are prominent research topics. From a statistical perspective, sixty-four percent of ERL-based studies utilized Q-learning as the training algorithm, while the remaining studies employed Sarsa, deep Q-network, and deep deterministic policy gradient algorithms. Overall, ERL methods in the energy and environment domain primarily focus on ensemble models where reinforcement learning algorithms are predominantly used directly. Compared with traditional approaches employing ML or ANN prediction methods [85, 86], there exists a significant gap between their respective prediction results and those obtained by ERL.

Table 4: Application in energy and environment area

Year	Authors	Problem	Training Algorithm
2020	Liu et al. [61]	wind speed short term forecasting	Q-learning
2021	Jalali et al. [87]	solar irradiance forecasting	Q-learning
2021	Liu et al. [54]	PM2.5 forecasting	Q-learning
2021	Li et al. [88]	PM2.5 forecasting	Q-learning
2021	Chao Chen and Hui Liu [89]	wind speed prediction	deep Q-Network
2021	Jalali et al. [60]	wind power forecasting	Q-learning
2022	Tan et al. [33]	PM2.5 prediction	Sarsa
2022	Qin et al. [90]	unit commitment problem	deep Q-Network
2022	Sogabe et al. [91]	smart energy optimization and risk evaluation	Q-learning
2022	Sharma et al. [59]	estimating reference evapotranspiration	Q-learning
2022	He et al. [92]	wind farm control	deep deterministic policy gradient
2022	Jalali et al. [93]	solar irradiance forecasting	Q-learning
2022	Shi Yin and Hui Liu [64]	wind power prediction	Q-learning
2023	Yu et al. [31]	wind power prediction	deep deterministic policy gradient

The most commonly employed improvement strategy in related studies, such as [31, 60, 64], involves the combination of multiple models and decision strategy design. By combining models with different structures, ERL enables diverse decision-making and achieves structural diversity. For instance, [33] integrates three distinct classes of spatio-temporal

ANNs, graph attention network (GAT), LSTM networks, and temporal convolutional network (TCN), simultaneously for prediction. These ANNs employ different processing logics, effectively ensuring distinguishable output results.

The application of ERL in the field of energy and environment primarily involves forecasting tasks, for which evaluation metrics used in traditional forecasting tasks are employed. To ensure an accurate assessment of the predictive performance of this method, mean absolute error (MAE), mean absolute percentage error (MAPE), and root-mean-square error (RMSE), are utilized [54]. In addition, the standard deviation of error (SDE) is also a common metric for evaluating ERL methods [88].

4.2. Internet of Things and Cloud Computing Area

In the area of the Internet of Things (IoT) and cloud computing, ERL is widely used to optimize system performance and business processing capabilities. The IoT connects various devices such as sensors, smart terminals, and industrial systems to form a globally interconnected system. Optimizing the efficiency of these devices and facilities has a positive impact on improving the overall performance of IoT systems. Cloud computing is another technology closely related to the IoT. Users can access computing resources or services in this distributed system provided by a cloud platform over the network on demand. Resource allocation and optimization have been the focus of research in the IoT and cloud computing area. Table 5 presents the applications of ERL methods in this area. Among these related works, the diversity of ERL in [94] is primarily ensured by the different inputs obtained through the utilization of the k-means method. This approach achieves data diversity. Other studies, such as [19, 95], have made advancements towards enhancing the composition of RL. Here, most studies use the offline algorithm, except for Polyzos et al. [95], who used an online algorithm. The performance of the ERL method has been verified on simulation platforms [96]. It can be seen from experimental results that the use of ensemble models makes the ERL method schedule significantly better than compared RL methods. When applying ERL in this area, matching the application requirements of multi-agent and distributed architecture becomes a core point. This system architecture allows the ensemble models in ERL to handle the same or different tasks.

For the domains of IoT and cloud computing, the evaluation metrics exhibit some variations. The performance of the ERL method for IoT sensor calibration in [97] is based on accuracy. Unlike [97], [95] examines the proposed method's effectiveness by analyzing the total cost of IoT resource allocation. In terms of the cloud computing domain, workload serves as a crucial evaluation metric [94].

4.3. Financial Area

In the financial area, complex decision-making problems, such as pricing financial products and portfolio optimization, are being tried to be solved by ERL methods. Though single models can make predictions on a specific problem, their generalization is affected by the problem scenario. Compared with the single model, ensemble models are affected less by the problem scenario factors. Table 6 presents the applications of ERL methods in finance. In these studies, 67% used only one training algorithm, while the rest of the studies used

Table 5: Application in IoT and cloud computing area

Year	Authors	Problem	Training Algorithm
2020	Ashiquzzaman et al. [97]	IoT sensor calibration	deep Q-Network
2021	Polyzos et al. [95]	resource allocation	Sarsa
2021	Jiang et al. [19]	large-scale MEC systems	deep Q-Network
2021	Gu et al. [94]	online cloud task scheduler	deep deterministic policy gradient
2021	Liu et al. [98]	deep reinforcement learning training on GPU cloud platform	actor-critic network
2022	Mahmud et al. [99]	non orthogonal multiple access unmanned aerial network	deep Q-Network

multiple training algorithms in an ERL method. Three algorithms, namely proximal policy optimization, advantage actor-critic, and deep deterministic policy gradient, have shown good performance in training.

Table 6: Application in financial area

Year	Authors	Problem	Training Algorithm
2020	Yang et al. [14]	stock trading	proximal policy optimization, advantage actor-critic, deep deterministic policy gradient
2020	Xu et al. [100]	fuel economy improvement	Q-learning
2021	Carta et al. [53]	stock market forecasting	deep Q-Network
2022	Li et al. [62]	regional GDP prediction	deep Q-Network
2022	Zijie Cao and Hui Liu [56]	carbon price forecasting	Q-learning
2022	Németh, Marcell and Szűcs, Gábor [74]	algorithmic trading	proximal policy optimization, advantage actor-critic, deep deterministic policy gradient

The utilization of multiple training algorithms constitutes a fundamental strategy employed by ERL to uphold diversity in financial domains [14, 74]. Specifically, the base learner is trained to obtain different parameter configurations for parameter diversity. The primary objective of ERL is prediction, whereby cumulative return, annualized return, annualized volatility, Sharpe ratio, and max drawdown emerge as five commonly adopted metrics for stock trading evaluation [14]. As a forecasting task within the domain of energy environment

studies, MAE, MAPE, and SDE can also be employed to evaluate method performance [62]. Additionally, Theil U statistic 1 (U1) can be utilized as an evaluative metric [56].

4.4. Other Areas

Apart from the previous three traditional application domains, ERL has also been successfully applied in various other fields such as transportation, medicine, and security, which will be discussed in detail within this section. Table 7 provides a comprehensive overview of these ERL methods primarily focus on prediction tasks while only a limited number of classification problems like diagnosis and recognition are addressed using ensemble techniques. Notably, the work conducted by Eriksson et al. [73], who employed ERL methods to tackle autonomous driving challenges, deserves special attention. If the ERL can be effectively implemented for small-scale autonomous driving assistance systems, it is highly likely to stimulate new research endeavors and practical applications in this domain.

Table 7: Application in other areas

Year	Authors	Problem	Area
2021	Ghosh et al. [40]	air traffic control	traffic
2021	Dong et al. [51]	traffic speed forecasting	traffic
2022	Shang et al. [32]	traffic volume forecasting	traffic
2022	Qi et al. [44]	traffic signal control	traffic
2022	Eriksson et al. [73]	autonomous driving	traffic
2016	Tang et al. [101]	symptom checker	medicine
2021	Jalali et al. [78]	COVID-19 diagnosis	medicine
2022	Birman et al. [63]	malware detection	security
2022	Li et al. [58]	rumor tracking	security
2023	Henna et al. [102]	FSO/RF communication systems	optics
2019	Cuayáhuitl et al. [103]	chatbots	dialogue system
2010	Alexander Hans and Steffen Udluft [36]	pole balancing	engineering control
2018	Ferreira et al. [68]	cognitive satellite communication	aerospace

In the future, we expect to see more areas using ERL methods for complex tasks. Existing research results can provide valuable references for subsequent research, including improving existing algorithms to overcome their limitations, or extending the problem domain to obtain new insights. In the next section, we discuss some potential directions for future research on ERL methods.

5. Datasets and Compared Methods

This section examines the datasets and comparison methods used in various studies related to ensemble reinforcement learning (ERL). As presented in Table 8, experiments are

conducted to evaluate the performance of the proposed ERL methods. The datasets used in these experiments mainly include real-world data and publicly available datasets or environments. Real-world data are useful for objectively testing the predictive or classification performance of the method for specific applications. For instance, studies in the field of energy and environment have gathered data from multiple cities to predict desired outcomes [31, 61]. In contrast, publicly available datasets or environments such as the OpenAI Gym environment in the field of reinforcement learning are widely used to test the predictive performance of algorithms for continuous/discrete actions [104]. The UCI machine learning repository is widely recognized as the predominant public dataset for classification problems in academic research [35]. Furthermore, some medical-specific datasets are also utilized in studies of disease diagnosis [78].

To assess the efficacy of the proposed ERL methods, various comparative approaches have been employed in existing literature. Among these, the single model-based RL method (SM-RL) is one of the simplest ways to reflect the effectiveness of the proposed ERL method [65]. The training algorithm used in SM-RL remains consistent with that of the ERL method. However, this compared method has limited convincing power. Consequently, some other studies have used other training algorithms to compare with the proposed algorithm from another perspective [15, 40]. In order to comprehensively evaluate the effectiveness of algorithms, it is necessary to separately assess different models, training algorithms, and integration methods [61].

The comparison methods are continuously evolving through ongoing research. In other words, existing ERLs proposed in the relevant literature serve as baselines or state-of-the-art (SOTAs) for comparison when introducing a new ERL method. However, reproducing the exact method described in the literature may pose challenges due to various influencing factors such as the environment and algorithm parameters. To address this issue, many journals or conferences now require disclosure of datasets, pseudo-code, code, model structure, hyper-parameter configuration, data partitions, tuning methods and statistical tests as basic requirements. Studies like [34, 85, 109] have provided detailed information on their models and training methods which greatly assist other researchers who consider them as SOTA methods. Nevertheless, some studies still lack certain details (e.g., random seeds and training strategies), resulting in reproduced results that fall short of expectations. This discrepancy arises from the fact that ERL is a class of improved RL methods where models can vary depending on the environment.

6. Open Questions and Future Research Directions

6.1. Open Questions

In this section, we summarize three open questions in ERL-based research that can contribute to the future development of ERL (see Figure 11).

6.1.1. Which Models to Choose?

Models serve as the foundation for constructing ERL methods and exert a direct influence on the ultimate outcome. The utmost crucial aspect of model selection lies in its capabil-

Table 8: Datasets and compared Methods

Year	Authors	Dataset	Compared Methods
2016	Osband et al. [105]	Atari games	DQN
2017	Chen et al. [8]	Atari games	A3C+
2017	Partalas et al. [106]	UCI machine learning repository	classifier combination methods voting (V) and SMT and the forward selection (FS), selective fusion (SF)
2018	Pearce et al. [107]	Cart Pole control problem	Q-learning with different layer NNs
2019	Dong et al. [51]	traffic speed dataset	GRU, LSTM, MLP, RBF, LSTM-GRU-GA
2019	Pan et al. [45]	Maze, Mountain Car, Robotic Soccer Game Simulation	counterpart
2019	Goyal et al. [48]	CATS (Competition on Artificial Time series) dataset	LSTM, ANN, Linear regression, Random Forest, Online NN
2019	Macheng Shen and Jonathan P How [108]	two-player asymmetric game	single model, RNN
2020	Qingfeng Lan et al. [39]	Mountain Car	Q-learning, Double Q-learning, Averaged Q-learning
2020	Liu et al. [61]	three different groups of measured wind speed data from Xinjiang wind farms	Network: LSTM method, the DBN method, the ESN method; Training algorithm: SARSA
2020	Lin et al. [43]	Maze, soccer robot game	orthogonal projection inverse reinforcement learning method (OP-IRL)
2020	Junta Wu and Huiyun Li [79]	2D Robot Arm Open Racing Car Simulator (TORCS)	DDPG
2020	Yang et al. [14]	Dow Jones 30 constituent stocks (at 01/01/2016)	PPO, A2C, DDPG
2020	Liu et al. [35]	UCI online data repository	classifiers combination approaches majority voting (MV), weighted voting (WV), ensemble selection methods forward selection (FS)
2021	Ghosh et al. [40]	open-source air traffic simulator	PPO
2021	Jalali et al. [87]	GHI data sets	adaptive hybrid model (AHM), hybrid feature selection method (HFS), Outlier-robust hybrid model (ORHM), novel hybrid deep neural network model (NHDNNM), OHS-LSTM
2021	Liu et al. [54]	data collected from a congested intersection in Changsha	RNN, ENN, ESN, DBN, RBF, GRNN, MLP
2021	Jalali et al. [78]	two well-known open-source image datasets named Mendely and Kaggle	the original version of GSK and eight powerful evolutionary algorithms including grasshopper optimization algorithm (GOA), Slime mold algorithm (SMA), genetic algorithm, gray wolf optimizer (GWO), particle swarm optimization (PSO), differential evolution (DE), biogeography-based optimization (BBO)
2022	Hassam Ullah Sheikh et al. [15]	Mujoco environments, Atari games	TD3, SAC and REDQ
2022	Shang et al. [32]	actual traffic volume data of nine stations of Changsha freeway	Chebnet, CNN, LSTM, DBN, RNN, ESN, multi-layer perceptron (MLP)
2022	Tan et al. [33]	actual data	RNN, the deep belief network (DBN), the echo state network (ESN), the error encoding network (ENN), General Regression Neural Network (GRNN), radial basis function network (RBF), multilayer perceptron (MLP)
2022	Li et al. [62]	three sets of data from three Provinces of China	ESN, ENN, RNN, BPNN, ELM, RBF
2022	Cao et al. [56]	The data for the three carbon trading markets come from the Hubei Carbon Trading Network, Beijing Carbon Emissions Electronic Trading Platform, and International Carbon Action Partnership (ICAP)	Network: TCN, BiLSTM, KELM, BPNN, MLP, echo state network (ESN), Elman neural network (ENN), and gradient boosting decision tree (GBDT); Training algorithm: SARSA
2022	Qin et al. [90]	historical load data of the California Independent System Operator (CASIO) from January 1, 2021 to July 5, 2021	PPO guided tree search, the MIQP algorithm with Gurobi 9.1
2022	Sogabe et al. [91]	optimal energy management in a residential building microgrid	mixed-integer linear programming (MILP)
2022	Birman et al. [63]	a range of real-world scenarios	Aggregation method
2022	Li et al. [58]	PHEME, RumorEval	Naive Bayes, SVM-SGD, Dense, BiLSTM, FastText, TextCNN, VRoC, some combinations of the above methods
2022	Sharma et al. [59]	two MEC servers and 30 IoTDs randomly distributed in the squared area with size 50m×50m	Actor-Critic, DDPG
2022	Schubert et al. [65]	SymCat's symptom-disease database	single model-based RL
2023	Yu et al. [31]	actual wind power data of nine wind turbines	GMDH, DBN, ESN, ENN, the extreme learning machine (ELM), the radial basis function (RBF), multi-layer perceptron

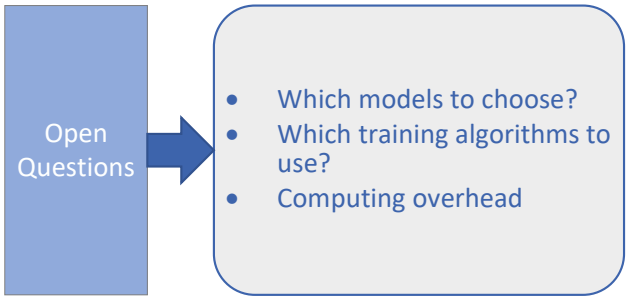


Figure 11: Open Questions. These three open questions are important and worthy of in-depth study for the development of ERL.

ity to perform feature selection and learning effectively. If a model cannot acquire valid information, it becomes devoid of meaning. Compared to single-model methods, ensemble methods employing multiple identical or diverse models can mitigate the risk of erroneous inference and enhance overall predictive performance. Consequently, it is imperative that the implemented models contribute significantly to the overall predictive performance of the ERL method.

The models implemented in ERL mainly include ML and ANN models, characterized by their simplistic structure and robust generalization capabilities, rendering them suitable for various practical applications. However, when confronted with large-scale data and numerous features, ML models may encounter challenges. In such scenarios, the ANN model excels at extracting features from datasets and generating predictions that closely align with actual values. Nevertheless, it is worth noting that certain tasks requiring comprehension pose difficulties for ANN models. To address this limitation, some studies such as [57], and [63] have explored the integration of both ML and ANN models within ERL methods.

Simultaneously utilizing models at different training stages greatly facilitates the design of ERL [53], as these models exhibit variations in parameters and distinct predicted bias and variance. When employing such an ensemble model, it is worthwhile to thoroughly investigate the conditions under which each ensemble element is preserved during the training process.

To automate the optimization of model structures, some studies have employed proposed ensemble pruning methods such as forward selection (FS) [110] and selective fusion (SF) [111]. These studies utilize a model library for selecting models, thereby reducing human effort in the selection process. The model library can also be continuously extended based on the latest research to optimize the overall performance of the ERL.

6.1.2. Which Training Algorithms to Use?

Model training poses a well-known challenge in the ERL method. For model-free based ERL, the agent updates model parameters based on state transfer or trajectory after a certain number of iterations. However, there is no guarantee that all sampled data are useful. The classic strategy is to use experience delay technology, which can improve the sampling efficiency. While Schaul et al. [112] found that the sampling inefficiency problem occurs when the sample in the relay buffer is useless. Selected experience relay makes the

algorithm training more efficient by selectively choosing the adopted data into the replay buffer [113]. It is worth noting that the experience relay buffer is only applicable to the off-policy RL method. Besides, model-based RL methods can guarantee sampling efficiency by learning environment models. However, such a training algorithm has to face a huge action space. In this case, approximating the environment model becomes an extremely difficult task.

Good training algorithms should balance exploration and exploitation, as relying solely on one approach proves challenging. Therefore, the future direction of development lies in employing multiple types of training algorithms simultaneously. This ERL method necessitates designing separate sampling techniques for the solution space based on the training strategy. Furthermore, it is crucial to consider how to use sampled data from the same strategy for model training processes. In addition, such algorithmic training imposes significant demands on the computational capabilities of both CPU and GPU.

6.1.3. Computing Overhead

Computing overhead is a closely related issue that must be taken into consideration for ERL, in addition to the aforementioned problems. Implementing multiple ML or ANN models in ERL results in an ultra-large number of parameters compared to a single model. Particularly when each ANN model possesses a complex structure, memory consumption becomes an indispensable factor. Similarly, multiple training algorithms can complicate the training process. Even with computational acceleration techniques, the time required for numerous computations can significantly exceed that of individual training sessions. Many studies have found that ERL methods can complete sampling efficiently, but are also accompanied by an increase in computation time [15, 95]. In the testing stage, complex decisions then easily lead to longer computation time than other methods [106]. So, some researchers tried to design strategies based on scenarios, which reduce the computational overhead to some extent. An et al. achieved a reduction in model training time along with a reduction in memory consumption by taking uncertainty into account in the ERL method [38]. Pan et al. reduced the time consumed by the algorithm for each iteration of training by fuzzifying the reward [45]. Up to now, the number of computationally cost-reducing models is still small. So, it is difficult to show that the improvement strategy is still applicable to large-scale models. In addition, the cost of data interaction needs extra attention when the models are deployed on multiple machines, which will affect the efficiency of the system.

The cost-effectiveness of ERL using complex structures and training processes is a fundamental basis for measuring method design and algorithm training. Increasing the number of models can improve the ensemble prediction performance but is also accompanied by an increase in computing overhead. After a certain number of models are implemented, the computing overhead of using more models can be significantly greater than the improvement in method performance. In such cases, increasing the size of the ensemble model is not advisable.

In certain practical application problems, the feasible solutions obtained through ERL methods can as problem-solving outcomes. Controlling the number of iterations of the training process is an alternative if the computing overhead of searching for the optimal

strategy is much greater than the contribution it can make. Thus, addressing the issue of computing overhead is essential for the successful application of ERL in various fields.

6.2. Future Research Directions

ERL has been extensively used in valuable research to effectively address scientific problems and various application domains. Based on the analysis of relevant literature mentioned in the survey, it is evident that a majority of the research has been concentrated within the past decade. Numerous unexplored research directions await investigation by scholars, and this section presents several potential directions for further inquiry.

1. **Randomized models**: Randomized models, such as random vector functional link networks [114], random initialized implicit layers, [115] have emerged as effective strategies for training reduction. In addition, the utilization of implicit/explicit ensembles [116] can improve the model training efficiency from the perspective of diversity. Ensuring diversity among base learners is a core problem that needs to be solved in the ERL method and deserves further study.

2. **Effect of decision strategy**: Decision strategies are employed to derive the final output based on the predictive outcomes of individual base learners. Despite various attempts made in previous studies to use different types of decision strategies, there remains a dearth of systematic research investigating their impact. Accordingly, it is imperative to conduct a comprehensive analysis of the decision strategies applicable in various integrated models and the number of training algorithms.

3. **Hierarchical ensemble**: Hierarchical reinforcement learning methods have been used to solve some challenging problems. For example, Qin et al. and Ferreira et al., respectively, endeavored to employ multiple RL models to complete different tasks separately in order [33, 67, 68]. The current model structure is designed based on empirical knowledge and lacks systematic theoretical validation, which calls for further investigation. In the context of hierarchical ensemble reinforcement learning, it is crucial to carefully evaluate the performance of both individual RL models and ensemble models. Additionally, a meticulous design should be employed to determine the specific role of each element within the hierarchical framework in order to address specific problems.

4. **Large-scale ensemble**: Existing ERL methods typically employ around three base learners [52, 55]. From the diversity perspective, incorporating a larger number of models in a new ERL method can effectively explore extensive feature information and enhance prediction accuracy. From a statistical perspective, increasing the number of ensemble components enables the generation of more hypotheses and enhances the likelihood of identifying the optimal hypothesis. By employing a large-scale ERL, it is possible to design an information-sharing mechanism, thereby reducing the total training cost of the model.

5. **Distributed approach**: Ensemble reinforcement learning can also be trained or used in a distributed manner. Existing research on distributed ERL primarily focuses on its implementation within a distributed framework and lacks methodological advancements [19, 73]. Therefore, further analysis is warranted on how to effectively leverage both ERL and distributed reinforcement learning. Integrating ERL into a distributed framework inevitably incurs augmented costs associated with model training and communication. Hence, in a

distributed framework, it becomes imperative to prioritize low-cost training methods and controlled training time to ensure the practical applicability of ERL.

6. **Online model training**: Currently, ERL adopts offline training and direct online implementation. However, this model training algorithm poses challenges in capturing the most up-to-date information, thereby affecting the optimality of the agent's strategy. To address this limitation, incorporating online or near-online model training methods would enable the timely addition of new information to the training dataset, ensuring that the model can effectively respond to dynamic situations. It is crucial for online training to focus on establishing a triggering mechanism for model updates as both over-training and under-training can adversely impact the performance of the ERL method. Moreover, forgetting history memory can facilitate ERL in discovering novel optimal strategies.

7. **Efficient training**: The sampling efficiency of DRL also deserves attention, as it remains a prevalent issue in ERL methods. Consequently, this gives rise to several associated challenges, including data set partitioning for training, model parameter initialization, hyperparameter tuning, and strategy updating. Models from different training stages can also be combined to identify the optimal configuration of model settings [53].

8. **Embedded into big data platform**: Although most of the current studies on ERL are conducted in simulation environments, there still exists a gap between these findings and their practical application. To address this issue, integrating ERL methods into a big data platform enables timely inference based on system-acquired data for various practical forecasting tasks. For both short-term forecasting and long-term forecasting objectives, diverse ERL methods can be deployed within the big data platform.

9. **Hyperparametric reduction**: The integration of multiple training algorithms in ERL can lead to a significant proliferation of hyperparameters within the method, thereby rendering model training arduous. Moreover, the trained model may exhibit instability when the scenario changes [76]. Hence, it is imperative to propose a mechanism for reducing hyperparameter complexity and alleviating the burden on researchers involved in hyperparameter tuning. One solution idea to solve this problem is to refer to the idea of pre-training by completing a rapid deployment of the model first. Subsequently, the deployed model is simply tuned to improve performance.

The above aspects provide potential directions for future research, although they are not exhaustive. It is important to acknowledge that the design and problem-solving processes of ERL methods may encounter various new situations. Furthermore, it should be noted that the no free lunch theorem applies universally to all ERL methods [20]. Therefore, when designing ERL methods, careful consideration must be given to their complexity and training time requirements.

7. Conclusion

This paper presents a comprehensive review of the research progress on ERL methods, covering various aspects including background, strategies, applications, and future directions. Firstly, the description of RL methods and EL methods has enhanced the understanding of ERL. Secondly, various strategies, such as Q-function ensemble, model combi-

nation, and decision strategies, are introduced and discussed. Subsequently, the application of ERL methods, datasets, and compared methods are described. Additionally, future research directions that can further enhance the performance of ERL have been extensively discussed.

The robust predictive and classification capabilities of ERL render it a promising framework for addressing intricate problems. E demonstrated successful applications across diverse domains, encompassing finance, robotics, and healthcare. Nevertheless, there exists substantial potential for future research endeavors. This paper highlights potential research directions including randomized models, the impact of decision strategies, hierarchical ensembles, large-scale ensembles, distributed approaches, online model training techniques, efficient training methodologies, and integration of ERL into big data platforms.

The future holds promising prospects for ERL performance across a wider range of application domains. Consequently, it is imperative for researchers to persistently explore and develop novel ERL methods that effectively tackle the challenges encountered in practical scenarios.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (72201273,72001212), the Science and Technology Innovation Team of Shanxi Province (2023-CX-TD-07), the Special Project in Major Fields of Guangdong Universities (2021ZDZX1019), and the Hunan Key Laboratory of Intelligent Decision-making Technology for Emergency Management (2020TP1013).

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, nature 518 (7540) (2015) 529–533.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, nature 529 (7587) (2016) 484–489.

[3] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al., Grandmaster level in starcraft ii using multi-agent reinforcement learning, Nature 575 (7782) (2019) 350–354.

[4] Ł. Kaiser, M. Babaeizadeh, P. Miłos, B. Osiński, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, et al., Model based reinforcement learning for atari, in: International Conference on Learning Representations, 2020.

[5] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin, B. Dresp-Langley, Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review, Robotics 10 (1) (2021) 22.

[6] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: International conference on machine learning, PMLR, 2018, pp. 1587–1596.

[7] A. Kumar, J. Fu, M. Soh, G. Tucker, S. Levine, Stabilizing off-policy q-learning via bootstrapping error reduction, Advances in Neural Information Processing Systems 32 (2019).

[8] R. Y. Chen, S. Sidor, P. Abbeel, J. Schulman, Ucb exploration via q-ensembles, arXiv preprint arXiv:1706.01502 (2017).

[9] M. d. Condorcet, Essay on the application of analysis to the probability of majority decisions, Paris: Imprimerie Royale (1785).

[10] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, Advances in neural information processing systems 7 (1995) 231–238.

[11] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.
[12] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, Information fusion 6 (1) (2005) 5–20.
[13] T. G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1, Springer, 2000, pp. 1–15.
[14] H. Yang, X.-Y. Liu, S. Zhong, A. Walid, Deep reinforcement learning for automated stock trading: An ensemble strategy, in: Proceedings of the first ACM international conference on AI in finance, 2020, pp. 1–8.
[15] H. Sheikh, M. Phielipp, L. Boloni, Maximizing ensemble diversity in deep reinforcement learning, in: International Conference on Learning Representations, 2022.
[16] X. Chen, C. Wang, Z. Zhou, K. W. Ross, Randomized ensembled double q-learning: Learning fast without a model, in: International Conference on Learning Representations, 2021.
[17] S. Faußer, F. Schwenker, Ensemble methods for reinforcement learning with function approximation, in: Multiple Classifier Systems: 10th International Workshop, MCS 2011, Naples, Italy, June 15-17, 2011. Proceedings 10, Springer, 2011, pp. 56–65.
[18] O. Anschel, N. Baram, N. Shimkin, Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning, in: International conference on machine learning, PMLR, 2017, pp. 176–185.
[19] F. Jiang, L. Dong, K. Wang, K. Yang, C. Pan, Distributed resource scheduling for large-scale mec systems: A multiagent ensemble deep reinforcement learning with imitation acceleration, IEEE Internet of Things Journal 9 (9) (2021) 6597–6610.
[20] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.
[21] C. J. Watkins, P. Dayan, Q-learning, Machine learning 8 (1992) 279–292.
[22] K. Arulkumaran, M. P. Deisenroth, M. Brundage, A. A. Bharath, Deep reinforcement learning: A brief survey, IEEE Signal Processing Magazine 34 (6) (2017) 26–38.
[23] L. Breiman, Bagging predictors, Machine learning 24 (1996) 123–140.
[24] R. E. Schapire, The boosting approach to machine learning: An overview, Nonlinear estimation and classification (2003) 149–171.
[25] D. H. Wolpert, Stacked generalization, Neural networks 5 (2) (1992) 241–259.
[26] G. Brown, J. L. Wyatt, P. Tino, Y. Bengio, Managing diversity in regression ensembles., Journal of machine learning research 6 (9) (2005).
[27] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural computation 4 (1) (1992) 1–58.
[28] J. Nalepa, M. Myller, L. Tulczyjew, M. Kawulok, Deep ensembles for hyperspectral image data classification and unmixing, Remote Sensing 13 (20) (2021) 4133.
[29] M. A. Ganaie, M. Hu, A. Malik, M. Tanveer, P. Suganthan, Ensemble deep learning: A review, Engineering Applications of Artificial Intelligence 115 (2022) 105151.
[30] J. Smit, C. T. Ponnambalam, M. T. Spaan, F. A. Oliehoek, Pebl: Pessimistic ensembles for offline deep reinforcement learning, in: Robust and Reliable Autonomy in the Wild Workshop at the 30th International Joint Conference of Artificial Intelligence, 2021.
[31] Y. Chengqing, Y. Guangxi, Y. Chengming, Z. Yu, M. Xiwei, A multi-factor driven spatiotemporal wind power prediction model based on ensemble deep graph attention reinforcement learning networks, Energy 263 (2023) 126034.
[32] P. Shang, X. Liu, C. Yu, G. Yan, Q. Xiang, X. Mi, A new ensemble deep graph reinforcement learning network for spatio-temporal traffic volume forecasting in a freeway network, Digital Signal Processing 123 (2022) 103419.
[33] J. Tan, H. Liu, Y. Li, S. Yin, C. Yu, A new ensemble spatio-temporal pm2.5 prediction method based on graph attention recursive networks and reinforcement learning, Chaos, Solitons & Fractals 162 (2022) 112405.
[34] I. Partalas, G. Tsoumakas, I. Katakis, I. Vlahavas, Ensemble pruning using reinforcement learning, in: Advances in Artificial Intelligence: 4th Helenic Conference on AI, SETN 2006, Heraklion, Crete,

Greece, May 18-20, 2006. Proceedings 4, Springer, 2006, pp. 301–310.
[35] Z. Liu, K. Ramamohanarao, Instance-based ensemble selection using deep reinforcement learning, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–7.
[36] A. Hans, S. Udluft, Ensembles of neural networks for robust reinforcement learning, in: 2010 Ninth International Conference on Machine Learning and Applications, IEEE, 2010, pp. 401–406.
[37] Q. He, H. Su, C. Gong, X. Hou, Mepg: A minimalist ensemble policy gradient framework for deep reinforcement learning, arXiv preprint arXiv:2109.10552 (2021).
[38] G. An, S. Moon, J.-H. Kim, H. O. Song, Uncertainty-based offline reinforcement learning with diversified q-ensemble, Advances in neural information processing systems 34 (2021) 7436–7447.
[39] Q. Lan, Y. Pan, A. Fyshe, M. White, Maxmin q-learning: Controlling the estimation bias of q-learning, arXiv preprint arXiv:2002.06487 (2020).
[40] S. Ghosh, S. Laguna, S. H. Lim, L. Wynter, H. Poonawala, A deep ensemble method for multi-agent reinforcement learning: A case study on air traffic control, in: Proceedings of the International Conference on Automated Planning and Scheduling, Vol. 31, 2021, pp. 468–476.
[41] D. Ormoneit, A. Sen, Kernel-based reinforcement learning, Machine learning 49 (2-3) (2002) 161.
[42] Y. Yao, L. Xiao, Z. An, W. Zhang, D. Luo, Sample efficient reinforcement learning via model-ensemble exploration and exploitation, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021, pp. 4202–4208.
[43] J.-L. Lin, K.-S. Hwang, H. Shi, W. Pan, An ensemble method for inverse reinforcement learning, Information Sciences 512 (2020) 518–532.
[44] R. Qi, J. Huang, H. Li, Q. Tan, L. Huang, J. Cui, Random ensemble reinforcement learning for traffic signal control, arXiv preprint arXiv:2203.05961 (2022).
[45] W. Pan, R. Qu, K.-S. Hwang, H.-S. Lin, An ensemble fuzzy approach for inverse reinforcement learning, International Journal of Fuzzy Systems 21 (2019) 95–103.
[46] S. Lee, Y. Seo, K. Lee, P. Abbeel, J. Shin, Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble, in: Conference on Robot Learning, PMLR, 2022, pp. 1702–1712.
[47] Z. Yang, K. Ren, X. Luo, M. Liu, W. Liu, J. Bian, W. Zhang, D. Li, Towards applicable reinforcement learning: Improving the generalization and sample efficiency with policy ensemble, arXiv preprint arXiv:2205.09284 (2022).
[48] A. Goyal, S. Sodhani, J. Binas, X. B. Peng, S. Levine, Y. Bengio, Reinforcement learning with competitive ensembles of information-constrained primitives, arXiv preprint arXiv:1906.10667 (2019).
[49] S. Adebola, S. Sharma, K. Shivakumar, Deft: Diverse ensembles for fast transfer in reinforcement learning, arXiv preprint arXiv:2209.12412 (2022).
[50] Y. Sun, P. Fazli, Ensemble policy distillation in deep reinforcement learning, in: Workshop on Reinforcement Learning in Games, 2020, pp. 1–9.
[51] S. Dong, C. Yu, G. Yan, J. Zhu, H. Hu, A novel ensemble reinforcement learning gated recursive network for traffic speed forecasting, in: 2021 Workshop on Algorithm and Big Data, 2021, pp. 55–60.
[52] S. K. Perepu, B. S. Balaji, H. K. Tanneru, S. Kathari, V. S. Pinnamaraju, Reinforcement learning based dynamic weighing of ensemble models for time series forecasting, arXiv preprint arXiv:2008.08878 (2020).
[53] S. Carta, A. Ferreira, A. S. Podda, D. R. Recupero, A. Sanna, Multi-dqn: An ensemble of deep q-learning agents for stock market forecasting, Expert systems with applications 164 (2021) 113820.
[54] X. Liu, M. Qin, Y. He, X. Mi, C. Yu, A new multi-data-driven spatiotemporal pm2.5 forecasting model based on an ensemble graph reinforcement learning convolutional network, Atmospheric Pollution Research 12 (10) (2021) 101197.
[55] D. L. Elliott, C. Anderson, The wisdom of the crowd: Reliable deep reinforcement learning through ensembles of q-functions, IEEE transactions on neural networks and learning systems (2021).
[56] Z. Cao, H. Liu, A novel carbon price forecasting method based on model matching, adaptive decomposition, and reinforcement learning ensemble strategy, Environmental Science and Pollution Research (2022) 1–24.
[57] A. Saadallah, K. Morik, Online ensemble aggregation using deep reinforcement learning for time series

forecasting, in: 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2021, pp. 1–8.
[58] G. Li, M. Dong, L. Ming, C. Luo, H. Yu, X. Hu, B. Zheng, Deep reinforcement learning based ensemble model for rumor tracking, Information Systems 103 (2022) 101772.
[59] G. Sharma, A. Singh, S. Jain, Deepevap: Deep reinforcement learning based ensemble approach for estimating reference evapotranspiration, Applied Soft Computing 125 (2022) 109113.
[60] S. M. J. Jalali, G. J. Osório, S. Ahmadian, M. Lotfi, V. M. Campos, M. Shafie-khah, A. Khosravi, J. P. Catalão, New hybrid deep neural architectural search-based ensemble reinforcement learning strategy for wind power forecasting, IEEE Transactions on Industry Applications 58 (1) (2021) 15–27.
[61] H. Liu, C. Yu, H. Wu, Z. Duan, G. Yan, A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting, Energy 202 (2020) 117794.
[62] Q. Li, C. Yu, G. Yan, A new multipredictor ensemble decision framework based on deep reinforcement learning for regional gdp prediction, IEEE Access 10 (2022) 45266–45279.
[63] Y. Birman, S. Hindi, G. Katz, A. Shabtai, Cost-effective ensemble models selection using deep reinforcement learning, Information Fusion 77 (2022) 133–148.
[64] S. Yin, H. Liu, Wind power prediction based on outlier correction, ensemble reinforcement learning, and residual correction, Energy 250 (2022) 123857.
[65] F. Schubert, C. Benjamins, S. Döhler, B. Rosenhahn, M. Lindauer, Polter: Policy trajectory ensemble regularization for unsupervised reinforcement learning, arXiv preprint arXiv:2205.11357 (2022).
[66] M. Shen, J. P. How, Robust opponent modeling via adversarial ensemble reinforcement learning in asymmetric imperfect-information games, arXiv preprint arXiv:1909.08735 (2019).
[67] Y. Qin, Z. Wang, C. Chen, Hrl2e: Hierarchical reinforcement learning with low-level ensemble, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 1–7.
[68] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. C. Reinhart, D. J. Mortensen, Multiobjective reinforcement learning for cognitive satellite communications using deep neural network ensembles, IEEE Journal on Selected Areas in Communications 36 (5) (2018) 1030–1041.
[69] A. Cully, Y. Demiris, Quality and diversity optimization: A unifying modular framework, IEEE Transactions on Evolutionary Computation 22 (2) (2017) 245–259.
[70] M. A. Wiering, H. Van Hasselt, Ensemble algorithms in reinforcement learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38 (4) (2008) 930–936.
[71] X. Chen, L. Cao, C. Li, Z. Xu, J. Lai, Ensemble network architecture for deep reinforcement learning, Mathematical Problems in Engineering 2018 (2018).
[72] R. Saphal, B. Ravindran, D. Mudigere, S. Avancha, B. Kaul, Seerl: Sample efficient ensemble reinforcement learning, arXiv preprint arXiv:2001.05209 (2020).
[73] H. Eriksson, D. Basu, M. Alibeigi, C. Dimitrakakis, Sentinel: taming uncertainty with ensemble based distributional reinforcement learning, in: Uncertainty in Artificial Intelligence, PMLR, 2022, pp. 631–640.
[74] M. Németh, G. Szűcs, Split feature space ensemble method using deep reinforcement learning for algorithmic trading, in: Proceedings of the 2022 8th International Conference on Computer Technology Applications, 2022, pp. 188–194.
[75] Y. Wang, K. Xue, C. Qian, Evolutionary diversity optimization with clustering-based selection for reinforcement learning, in: International Conference on Learning Representations, 2021.
[76] B. Zhang, R. Rajan, L. Pineda, N. Lambert, A. Biedenkapp, K. Chua, F. Hutter, R. Calandra, On the importance of hyperparameter optimization for model-based reinforcement learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 4015–4023.
[77] S. Faußer, F. Schwenker, Neural network ensembles in reinforcement learning, Neural Processing Letters 41 (2015) 55–69.
[78] S. M. J. Jalali, M. Ahmadian, S. Ahmadian, A. Khosravi, M. Alazab, S. Nahavandi, An oppositional-cauchy based gsk evolutionary algorithm with a novel deep ensemble reinforcement learning strategy for covid-19 diagnosis, Applied Soft Computing 111 (2021) 107675.

[79] J. Wu, H. Li, Deep ensemble reinforcement learning with multiple deep deterministic policy gradient algorithm, Mathematical Problems in Engineering 2020 (2020) 1–12.
[80] H. Sheikh, K. Frisbee, M. Phielipp, Dns: Determinantal point process based neural network sampler for ensemble reinforcement learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 19731–19746.
[81] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, H. Lee, Sample-efficient reinforcement learning with stochastic ensemble value expansion, Advances in neural information processing systems 31 (2018).
[82] G. Chen, Y. Peng, M. Zhang, Effective exploration for deep reinforcement learning via bootstrapped q-ensembles under tsallis entropy regularization, arXiv preprint arXiv:1809.00403 (2018).
[83] O. Peer, C. Tessler, N. Merlis, R. Meir, Ensemble bootstrapping for q-learning, in: International Conference on Machine Learning, PMLR, 2021, pp. 8454–8463.
[84] A. Brown, M. Petrik, Interpretable reinforcement learning with ensemble methods, arXiv preprint arXiv:1809.06995 (2018).
[85] U. Pak, J. Ma, U. Ryu, K. Ryom, U. Juhyok, K. Pak, C. Pak, Deep learning-based pm2. 5 prediction considering the spatiotemporal correlations: A case study of beijing, china, Science of The Total Environment 699 (2020) 133561.
[86] J. Ma, Z. Yu, Y. Qu, J. Xu, Y. Cao, et al., Application of the xgboost machine learning method in pm2. 5 prediction: A case study of shanghai, Aerosol and Air Quality Research 20 (1) (2020) 128–138.
[87] S. M. J. Jalali, M. Khodayar, S. Ahmadian, M. Shafie-Khah, A. Khosravi, S. M. S. Islam, S. Nahavandi, J. P. Catalão, A new ensemble reinforcement learning strategy for solar irradiance forecasting using deep optimized convolutional neural network models, in: 2021 International Conference on Smart Energy Systems and Technologies (SEST), IEEE, 2021, pp. 1–6.
[88] Y. Li, Z. Liu, H. Liu, A novel ensemble reinforcement learning gated unit model for daily pm2.5 forecasting, Air Quality, Atmosphere & Health 14 (2021) 443–453.
[89] C. Chen, H. Liu, Dynamic ensemble wind speed prediction model based on hybrid deep reinforcement learning, Advanced Engineering Informatics 48 (2021) 101290.
[90] J. Qin, Y. Gao, M. Bragin, N. Yu, An optimization method-assisted ensemble deep reinforcement learning algorithm to solve unit commitment problems, arXiv preprint arXiv:2206.04249 (2022).
[91] T. Sogabe, D. B. Malla, C.-C. Chen, K. Sakamoto, Attention and masking embedded ensemble reinforcement learning for smart energy optimization and risk evaluation under uncertainties, Journal of Renewable and Sustainable Energy 14 (4) (2022) 045501.
[92] B. He, H. Zhao, G. Liang, J. Zhao, J. Qiu, Z. Y. Dong, Ensemble-based deep reinforcement learning for robust cooperative wind farm control, International Journal of Electrical Power & Energy Systems 143 (2022) 108406.
[93] S. M. J. Jalali, S. Ahmadian, B. Nakisa, M. Khodayar, A. Khosravi, S. Nahavandi, S. M. S. Islam, M. Shafie-khah, J. P. Catalão, Solar irradiance forecasting using a novel hybrid deep ensemble reinforcement learning algorithm, Sustainable Energy, Grids and Networks 32 (2022) 100903.
[94] D. Gu, J. Chen, X. Shi, L. Ran, Y. Zhang, M. Shang, Heterogeneous-aware online cloud task scheduler based on clustering and deep reinforcement learning ensemble, in: Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery, Springer, 2021, pp. 152–159.
[95] K. D. Polyzos, Q. Lu, A. Sadeghi, G. B. Giannakis, On-policy reinforcement learning via ensemble gaussian processes with application to resource allocation, in: 2021 55th Asilomar Conference on Signals, Systems, and Computers, IEEE, 2021, pp. 1018–1022.
[96] A. Sadeghi, F. Sheikholeslami, G. B. Giannakis, Optimal and scalable caching for 5g using reinforcement learning of space-time popularities, IEEE Journal of Selected Topics in Signal Processing 12 (1) (2017) 180–190.
[97] A. Ashiquzzaman, H. Lee, T.-W. Um, J. Kim, Energy-efficient iot sensor calibration with deep reinforcement learning, IEEE Access 8 (2020) 97045–97055.
[98] X.-Y. Liu, Z. Li, Z. Yang, J. Zheng, Z. Wang, A. Walid, J. Guo, M. I. Jordan, Elegantrl-podracer: Scalable and elastic library for cloud-native deep reinforcement learning, arXiv preprint arXiv:2112.05923 (2021).

[99] S. K. Mahmud, Y. Chen, K. K. Chai, Ensemble reinforcement learning framework for sum rate optimization in noma-uav network, in: 2022 IEEE World AI IoT Congress (AIIoT), IEEE, 2022, pp. 032–038.
[100] B. Xu, X. Hu, X. Tang, X. Lin, H. Li, D. Rathod, Z. Filipi, Ensemble reinforcement learning-based supervisory control of hybrid electric vehicle for fuel economy improvement, IEEE Transactions on Transportation Electrification 6 (2) (2020) 717–727.
[101] K.-F. Tang, H.-C. Kao, C.-N. Chou, E. Y. Chang, Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning, in: NIPS workshop on deep reinforcement learning, 2016.
[102] S. Henna, A. A. Minhas, M. S. Khan, M. S. Iqbal, Ensemble consensus representation deep reinforcement learning for hybrid fso/rf communication systems, Optics Communications 530 (2023) 129186.
[103] H. Cuayáhuitl, D. Lee, S. Ryu, Y. Cho, S. Choi, S. Indurthi, S. Yu, H. Choi, I. Hwang, J. Kim, Ensemble-based deep reinforcement learning for chatbots, Neurocomputing 366 (2019) 118–130.
[104] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, arXiv preprint arXiv:1606.01540 (2016).
[105] I. Osband, C. Blundell, A. Pritzel, B. Van Roy, Deep exploration via bootstrapped dqn, Advances in neural information processing systems 29 (2016).
[106] I. Partalas, G. Tsoumakas, I. Vlahavas, Pruning an ensemble of classifiers via reinforcement learning, Neurocomputing 72 (7-9) (2009) 1900–1909.
[107] T. Pearce, N. Anastassacos, M. Zaki, A. Neely, Bayesian inference with anchored ensembles of neural networks, and application to exploration in reinforcement learning, arXiv preprint arXiv:1805.11324 (2018).
[108] M. Shen, J. P. How, Robust opponent modeling via adversarial ensemble reinforcement learning, in: Proceedings of the International Conference on Automated Planning and Scheduling, Vol. 31, 2021, pp. 578–587.
[109] J. Wang, J. Hu, J. Mills, G. Min, M. Xia, Federated ensemble model-based reinforcement learning in edge computing, arXiv preprint arXiv:2109.05549 (2021).
[110] R. Caruana, A. Niculescu-Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, in: Proceedings of the twenty-first international conference on Machine learning, 2004, p. 18.
[111] G. Tsoumakas, L. Angelis, I. Vlahavas, Selective fusion of heterogeneous classifiers, Intelligent Data Analysis 9 (6) (2005) 511–525.
[112] T. Schaul, J. Quan, I. Antonoglou, D. Silver, Prioritized experience replay, arXiv preprint arXiv:1511.05952 (2015).
[113] D. Isele, A. Cosgun, Selective experience replay for lifelong learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[114] Y.-H. Pao, G.-H. Park, D. J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.
[115] Q. Shi, R. Katuwal, P. N. Suganthan, M. Tanveer, Random vector functional link neural network based ensemble deep learning, Pattern Recognition 117 (2021) 107978.
[116] B. Han, J. Sim, H. Adam, Branchout: Regularization for online ensemble tracking with convolutional neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3356–3365.