Wireless Communications and Mobile Computing

Artificial Intelligence for Urban Internet of Things

Lead Guest Editor: Han Liu
Guest Editors: Binghui Wang, Zhaoxing Li, and Xiaotong Zhang

Artificial Intelligence for Urban Internet of Things

Wireless Communications and Mobile Computing

Artificial Intelligence for Urban Internet of Things

Lead Guest Editor: Han Liu
Guest Editors: Binghui Wang, Zhaoxing Li, and Xiaotong Zhang

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9402671, 14 pages
https://doi.org/10.1155/2022/9402671

Research Article

Minimizing the Cost of Spatiotemporal Searches Based on Reinforcement Learning with Probabilistic States

Lei Han,[1] **Chunyu Tu**,[2] **Zhiyong Yu**,[2] **Fangwan Huang**,[2] **Wenzhong Guo**,[2] **Chao Chen**,[3] **and Zhiwen Yu**[1]

[1]*School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China*
[2]*College of Mathematics and Computer Sciences, Fuzhou University, Key Laboratory of Spatial Data Mining and Information Sharing, Ministry of Education, and Fujian Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou 350108, China*
[3]*School of Computer Science, Chongqing University, Chongqing 400044, China*

Correspondence should be addressed to Fangwan Huang; hfw@fzu.edu.cn

Received 2 March 2022; Accepted 29 April 2022; Published 8 June 2022

Academic Editor: Han Liu

Portraying the trajectories of certain vehicles effectively is of great significance for urban public safety. Specially, we aim to determine the location of a vehicle at a specific past moment. In some situations, the waypoints of the vehicle's trajectory are not directly available, but the vehicle's image may be contained in massive camera video records. Since these records are only indexed by location and moment, rather than by contents such as license plate numbers, finding the vehicle from these records is a time-consuming task. To minimize the cost of spatiotemporal search (a spatiotemporal search means the effort to check whether the vehicle appears at a specified location at a specified moment), this paper proposes a reinforcement learning algorithm called Quasi-Dynamic Programming (QDP), which is an improved Q-learning. QDP selects the searching moment iteratively based on known past locations, considering both the cost efficiency of the current action and its potential impact on subsequent actions. Unlike traditional Q-learning, QDP has probabilistic states during training. To address the problem of probabilistic states, we make the following contributions: 1) replaces the next state by multiple states of a probability distribution; 2) estimates the expected cost of subsequent actions to calculate the value function; 3) creates a state and an action randomly in each loop to train the value function progressively. Finally, experiments are conducted using real-world vehicle trajectories, and the results show that the proposed QDP is superior to the previous greedy-based algorithms and other baselines.

1. Introduction

Portraying the trajectories of some vehicles in a city is important to foresee potential safety issues, for example, to infer the intention of a suspicious vehicle. We only get the locations of the vehicle at some key moments, instead of at all moments for economic efficiency. Figure 1 shows a typical application scenario. The sequential numbers indicate the waypoints of the vehicle trajectory from 8 : 00 to 9 : 00 on a certain day, of which the yellow ones are the locations of some key moments. By locating a suspicious vehicle at certain times (such as target time, etc.) and marking them on a map until we can get a rough idea of its trajectory.

To track vehicles in a city, vehicle Re-Identification (Re-Id) [1–4] focuses on how to identify the target vehicle in camera video records as accurately as possible. However, Re-Id does not consider how to select records, which are massive and only indexed by location and moment. As a result, Re-Id needs to be performed on all records exhaustively. With the rapid urbanization, there are a large number of cameras in the city. By searching within the video records of these cameras, it is possible but time-consuming to find the target vehicle at the target moment, because these records are massive and only indexed by location and moment, rather than by the contents such as the license plate number. Besides the straightforward strategy that all searches are performed at

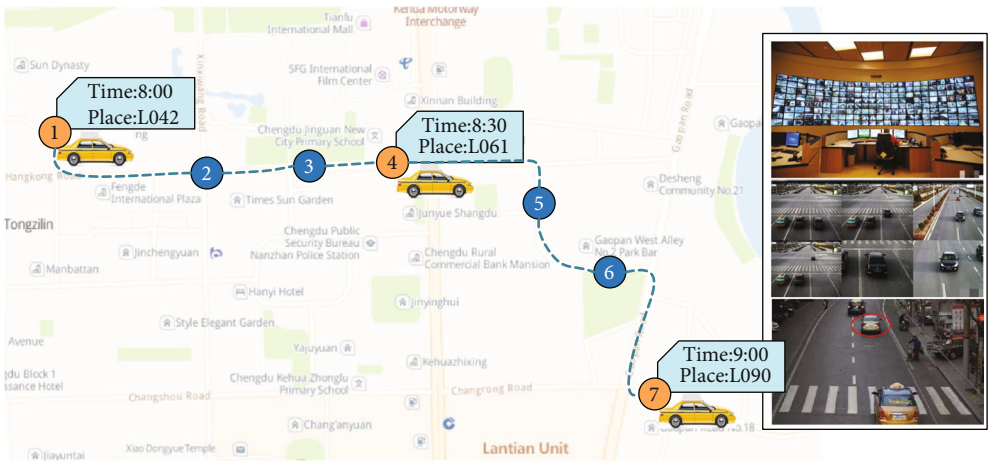


Figure 1: Vehicle location at specific moments.

the target moment, another strategy worth exploring is to spend some searches at intermediate moments to determine the vehicle's likely subsequent path, thus significantly narrowing the search area at the target moment. We call the effort to check whether the vehicle appears at a specified location at a specified moment as a spatiotemporal search.

How to determine the location of a vehicle at the target moment with the minimumcost of spatiotemporal searches in massive camera video records? The cost of a spatiotemporal search can be considered as a constant for simplicity, regarding either human or artificial intelligence. In previous work [5], we proposed two greedy algorithms named Intermediate Searching at Heuristic Spatiotemporal Units (IHUs) and Intermediate Searching at Heuristic Moments (IHMs), and some other simple but uncompetitive baselines. Compared with the algorithm IHUs that considers both moments and location, IHMs can fully utilize the information gained from previous failed searches, performing location-by-location searches at the same time. Since IHMs is based on the greedy heuristic, it may result a locally optimal solutions. The most common way to go beyond the locally optimal solution is dynamic programming. Dynamic programming is essentially an exhaustive idea. By comparing all solutions, it can guarantee that the solution is globally optimal. However, there are too many spatiotemporal points to be considered, making the solution space too large to list all solutions. Reinforcement learning can alleviate the problem through quasi-dynamic programming process. The spatiotemporal search strategy based on reinforcement learning selects an intermediate moment according to known vehicle spatiotemporal points, taking into account the cost efficiency of the current action and its potential impact on subsequent actions. The strategy of spatiotemporal search is a standard finite Markov decision process [6]. Q-learning, a reinforcement learning algorithm, has been proved to eventually converge to the optimum action values with probability 1 so long as all actions are repeatedly sampled in all states and the action values are represented discretely [7, 8]. However, if adopting Q-learning directly to address our problem, since the next state (known spatiotemporal point of the vehicle trajectory) cannot be determined after the action (i.e., selecting an intermediate moment and performing the corresponding spatiotemporal searches), it is difficult to update the next state and the value function when training. To overcome this challenge, this paper proposes Quasi-Dynamic Programming (QDP) algorithm to improve the training phase of Q-learning. The contributions of this paper include:

(1) We propose a reinforcement learning algorithm called QDP that can result the near optimal solution to the problem of minimizing spatiotemporal search cost

(2) To address the challenge of probabilistic state in the training phase, we propose a novel training method for QDP, which replaces the next state by multiple states with a probability distribution, estimates the expected search cost of subsequent selections to calculate the value function, and creates a state and an action randomly in each loop to train the value function progressively

(3) We evaluate the proposed QDP on real-world vehicle trajectories, and the results show that it is superior to the previous greedy-based algorithms (IHMs) and other baselines

The contents of this paper are arranged as following: Section 2 discusses some related work; Section 3 proposes the overall process of QDP, and explains the training method for QDP; then the experiments are conducted in Section 4; finally, the conclusions and future work are summarized in Section 5.

2. Related Work

Since this paper addresses the search problem based on reinforcement learning, we will introduce the related work from the two aspects of search theory and reinforcement learning.

2.1. Search Theory. The core issue of search theory is designing a search strategy to find a missing target, under constraints such as time limit, forces required, and task

budget. Related work mainly comes from the field of operations research or cybernetics.Koopman [9–11] used the basic probability theory to optimize the conFigureuration of budgeted search forces to find stationary targets. Stone [12, 13] presented detailed mathematical assumptions and theoretical proofs for specific problems of moving target searching and found the necessary conditions for the optimal search strategy. Besides the above theoretical analysis, some other works adopted the search theory in practical application scenarios. Researchers [14–17] established a management framework of task planning and feedback controlling based on the classical search theory. For the task planning of satellite SPOT-5, Bensana [18, 19] and Agnese [20] compared the computational performance of complete searching algorithms (i.e., Depth-first Search, Dynamic Programming, Russian Doll Search) and incomplete searching algorithms (i.e., Greedy Search, Tabu Search) under different problem scales. To locate the target to be observed in a region by satellite scanning, Lemaitre et al. [21] discussed Greedy Search, Dynamic Programming, Constrained Programming, and Genetic Algorithms to solve the problem. Different from the existing work, we study a more realistic problem with high rate of knowledge update in the application scenario of vehicle tracking.

2.2. Reinforcement Learning. Reinforcement learning [22, 23] is used to describe and solve the problem that the agent learns to obtain the maximum reward or achieve a specific goal in the process of interacting with the environment. Reinforcement learning is widely used in areas such as process control, task scheduling, robotics, and smart games [24–26]. Some complex reinforcement learning algorithms can reach or even surpass the human level in certain tasks [27, 28]. Q-learning is one reinforcement learning algorithm we pay attention to in this paper. Watkins [29] proposed Q-learning first, which utilizes the reward of state-action pair for value function updating. Rummery and Niranjan [30] proposed SARSA, a reinforcement learning algorithm with online policy based on Q-learning. Harm van Seijen et al. [31] presents a theoretical and empirical analysis of Expected Sarsa, a variation on Sarsa, the classic on-policy temporal difference method for model-free reinforcement learning. DeepMind [32] proposed DQN (Deep Q-Networks) by combining convolutional neural networks and Q-learning. Different from traditional Q-learning that the next state is uniquely determined, in this paper, the next state is multiple states of a probability distribution, i.e., probabilistic states. We propose a novel training method to overcome the difficulty brought by probabilistic states.

3. Overall Process of Quasi-Dynamic Programming (QDP)

Symbols used in this paper are shown in Notations. Let $\mathrm{O} = \{o_1, o_2, \cdots, o_{|O|}\}$ be vehicles, $\mathrm{L} = \{l_1, l_2, \cdots, l_{|L|}\}$ be locations of a city, $\mathrm{D} = \{d_1, d_2, \cdots, d_{|D|}\}$ be days, $\mathrm{Tr} = \{tr_1, tr_2, \cdots, tr_{|Tr|}\}$ be trajectories. In this paper, a trajectory is defined as a sequence of spatiotemporal points, i.e., $tr(o_x, d_j) = \langle (t_1, l_{t_1}), (t_2, l_{t_2}), \cdots \rangle | o_x, d_j$.

Definition 1 (a spatiotemporal search). The effort to check whether the target vehicle o_x appears at a specified location l_i at a specified moment t_{opt} in the camera records of that spatiotemporal point (t_{opt}, l_i). If vehicle o_x is found at location l_i at moment t_{opt}, ie $(t_{opt}, l_i) \in tr(o_x, d_j)$, it will returns 1, otherwise it returns 0. Formally,

$$s(o_x, t_{opt}, l_i) = \begin{cases} 1, & (t_{opt}, l_i) \in tr(o_x, d_j) \\ 0, & else \end{cases} \tag{1}$$

Problem 2 (tracking an object by spatiotemporal search). Give an object o_x, a day d_x, a past moment t_p of that day, o_x's location l_{t_p} at that moment, a current moment t_x, camera records that make the spatiotemporal search $s(o_x, t_k, l_i)$ available in d_x before/at t_x, as well as history trajectories TR', in which arbitrary $d_j < d_x$. When and where to execute a spatiotemporal search can base on TR' and the outputs of previous searches. The problem is how to utilize a minimal number of searchs to find the object at current moment t_x. Formally,

$$\begin{gathered} \textbf{give}\, o_x, d_x, t_x, \left(t_p, l_{t_p}\right), TR' \text{ before } s(o_x, t_k, l_i) \text{ in } d_x \\ \textbf{return}\ l_x, \ \text{s.t.} \min_{s(o_x, t_x, l_x)} |\langle s(), s(), \cdots, s(o_x, t_x, l_x) \rangle| \end{gathered} \tag{2}$$

The difficulty of a spatiotemporal search is affected by many factors in the spatiotemporal point (e.g., traffic flow, road network, and building density). It needs to be quantified by professionals or artificial intelligence. Let $\mathrm{C} = \{c_{l_1}^1, c_{l_2}^1, \cdots, c_{l_{|L|}}^1, \cdots, c_{l_1}^{|d_x|}, c_{l_2}^{|d_x|}, \cdots, c_{l_{|L|}}^{|d_x|}\}$ contains the search cost per spatiotemporal point on day d_x. It can be seen from the problem definition that we have to find the vehicle finally, so we use the cost of spatiotemporal search as the only indicator to measure the performance of different algorithms.

The overall process of QDP is shown in Figure 2. The process includes two phases: the training phase and executing phase. The training phase outputs the location decision model and the moment decision model. The location decision model is trained with the help of mobility prediction and basic optimal searching. This model will return the optimal searching location sequence at a given moment. The moment decision model is trained based on Monte-Carlo method and probabilistic states. This model will return the optimal moment that will be searched in the next action. The executing phase describes the online spatiotemporal search utilizing the trained moment decision model and location decision model.

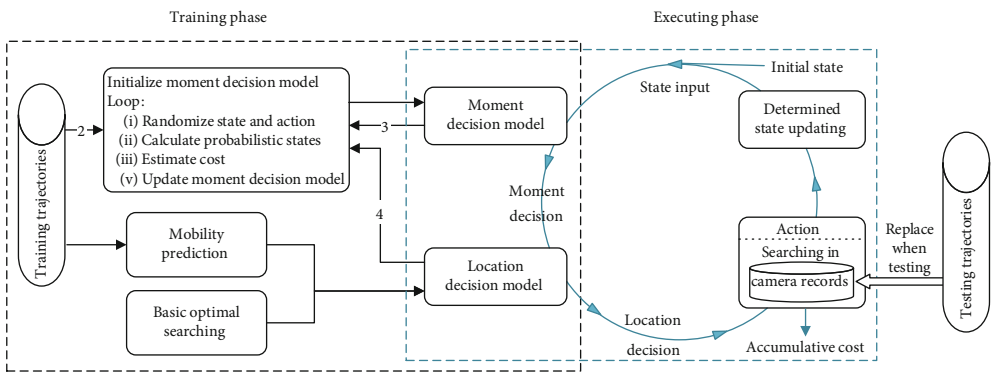


Figure 2: Overall Process of Quasi-Dynamic Programming (QDP).

3.1. Training the Location Decision Model. The location decision model is responsible to output the location decision$\langle t_{opt}, \xi_{opt}\rangle$ (the optimal searching location sequence at moment t_{opt} given by the moment decision model). It is trained with the help of mobility prediction and basic optimal searching. The mobility prediction algorithm is not the focus of this paper. In the previous work [5], we have analyzed the advantages of choosing the first-order Markov model for mobility prediction. Suppose that the transition probability matrix $TPM_{\Delta t}^{|L|\times|L|}$ represents the probabilities of a vehicle moving from one location to another after a period of time $\Delta t = t_{opt} - t_s$, of which an element is $p_{l_i}^{opt}$, i.e., the probability of vehicle o_x moves from location l_s at moment t_s to location l_i at moment t_{opt}.

$$TPM_{\Delta t}^{|L|\times|L|} = \left\{p_{l_i}^{opt}\right\} \tag{3}$$

$$p_{l_i}^{opt} = \frac{\#tr()with(t_s, l_s)\&(t_{opt}, l_i)}{\#tr()with(t_s, l_s)}, t_s = t_1, t_2, \cdots, tr() \in TR' \tag{4}$$

where the meaning of "#" is "the number of", $tr()$ is a trajectory. Equation Equation (4) represents the ratio of "the number of trajectories that pass both (t_s, l_s) and (t_{opt}, l_i)" to "the number of trajectories of only the pass (t_s, l_s)". $TPM_{\Delta t}^{|L|\times|L|}$ can be trained from historical vehicle trajectories. When given an original location, a row of probabilities is output as the predicted result of the destination location.

The basic optimal searching theory can assign the priority of locations to search at. Suppose the searching location sequence at moment t_{opt} is $\langle t_{opt}, \xi_{opt}\rangle = \langle (t_{opt}, l_1^{\xi_{opt}}), (t_{opt}, l_2^{\xi_{opt}}), \cdots, (t_{opt}, l_i^{\xi_{opt}}), \cdots\rangle, l_i^{\xi_{opt}} \in L$.

Definition 3 (probability-cost ratio). The ratio of the probability $p_{l_i^{\xi}}^{opt}$ of finding the target vehicle to the cost $c_{l_i^{\xi}}^{opt}$ of the $i-th$ spatiotemporal search in the sequence $\langle t_{opt}, \xi_{opt}\rangle$, formally,

$$\eta\left(t_{opt}, l_i^{\xi_{opt}}\right) = \frac{p_{l_i^{\xi}}^{opt}}{c_{l_i^{\xi}}^{opt}} \tag{5}$$

Theorem 4. *[33]: The necessary and sufficient condition for the sequence $\langle t_{opt}, \xi_{opt}\rangle$ to be the searching location sequence with the lowest search cost at moment t_{opt} is that the probability-cost ratios are in descending order.*

Consequently, the location decision can be made according to Theorem 4.

We need to estimate the search cost which will be used during training the moment decision model. The optimal sequence $\langle t_{opt}, \xi_{opt}\rangle$ has a expected search cost:

$$Cost\left(t_{opt}, \xi_{opt}\right) = \sum_{i=1}^{|L|} p_{l_i^{\xi}}^{opt} \times \sum_{j=1}^{i} c_{l_j^{\xi}}^{opt} \tag{6}$$

To demonstrate how to calculate the expected search cost using Equation (6), we assume the spatiotemporal search costs $c_{l_1}^3 = c_{l_2}^3 = c_{l_3}^3 = c_{l_4}^3 = 1$, the location transition probabilities $p_{l_1}^3 = 1/3, p_{l_2}^3 = 0, p_{l_3}^3 = 2/3, p_{l_4}^3 = 0$. Then, the descendingly sorted location sequence is $\langle t_3, \xi_3\rangle = \langle (t_3, l_3), (t_3, l_1), (t_3, l_2), (t_3, l_4)\rangle$ at moment t_3. Then the expected search cost is:

$$\begin{aligned} Cost(t_3, \xi_3) &= p_{l_3}^3 \times c_{l_3}^3 + p_{l_1}^3 \times \left(c_{l_3}^3 + c_{l_1}^3\right) + p_{l_2}^3 \times \left(c_{l_3}^3 + c_{l_1}^3 + c_{l_2}^3\right) \\ &\quad + p_{l_4}^3 \times \left(c_{l_3}^3 + c_{l_1}^3 + c_{l_2}^3 + c_{l_4}^3\right) = \frac{2}{3} \times 1 + \frac{1}{3} \\ &\quad \times (1+1) + 0 \times (1+1+1) + 0 \times (1+1+1+1) \\ &= \frac{4}{3} \end{aligned} \tag{7}$$

3.2. Training the Moment Decision Model. Perform spatiotemporal searches at moment t_{opt} should consider both the cost-efficiency of the current action and its potential impact on subsequent actions. So, how to select the next searching moment t_{opt} when knowing a spatiotemporal point (t_s, l_s)

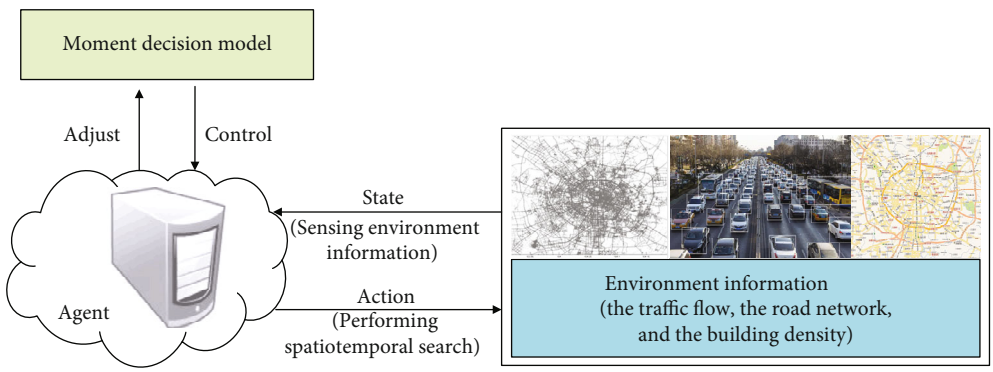


Figure 3: Training of the moment decision model.

of target vehicle o_x? The moment decision model is responsible to answer this question.

Following reinforcement learning, the moment decision model is represented as (S, A, C, Γ, Q), in which $S^s_{l_s} \in S$ denotes the current state (t_s, l_s); $A^s_{l_s} \in A$ denotes the current action $\langle t_{opt}, \xi_{opt}\rangle$; $C : S \times A \longrightarrow C$ denotes the search cost function $C^{opt} = Cost(t_{opt}, \xi_{opt})$; $\Gamma : S \times A \longrightarrow \Delta S$ denotes the state transition function; Q denotes the value function. The next state $S^s_{l_s}{}'$ is obtained by performing action $A^s_{l_s}$ at the state $S^s_{l_s}$, and the value function is updated using Equation (8), where α is the learning rate between (0, 1], and γ is the discount rate.

$$Q\left(S^s_{l_s}, A^s_{l_s}\right) = Q\left(S^s_{l_s}, A^s_{l_s}\right) + \alpha.\left(C^{opt} + \gamma.\min_{A^s_{l_s}{}'} Q\left(S^s_{l_s}{}', A^s_{l_s}{}'\right) - Q\left(S^s_{l_s}, A^s_{l_s}\right)\right) \tag{8}$$

The learning rate α can control the update rate of the value function. The discount rate γ can balance the importance between the immediate and potential reward. We use $Q(S^s_{l_s}{}', A^s_{l_s}{}')$to estimate the long-term impact of performing action $A^s_{l_s}$ at state $S^s_{l_s}$.

Figure 3 shows how to train the moment decision model. First of all, the moment decision model is initialized. In each loop, the moment decision model decides a searching moment based on the current state, and the location decision model decides the corresponding searching location sequence. Then the value function is adjusted according to the potential impact on subsequent actions and the sensed environmental information (e.g., traffic flow, road network, and building density, which will be quantified as search cost C^{opt}). The moment decision model will continually repeat the above loop until the determined searching moment t_{opt} at the current state $S^s_{l_s}$ is optimal.

To train the moment decision model in QDP, this paper proposes a training method based on Monte-Carlo method and probabilistic states to solve the problem, as shown in Figure 4.

In the training of QDP, performing the action $A^s_{l_s} = \langle t_{opt}, \xi_{opt}\rangle$ at the state $S^s_{l_s} = (t_s, l_s)$ will result in a probability distribution $\{p^{opt}_{l^\xi_1}, p^{opt}_{l^\xi_2}, \cdots, p^{opt}_{l^\xi_{|L|}}\}$ at different locations $\{l^{\xi_{opt}}_1, l^{\xi_{opt}}_2, \cdots, l^{\xi_{opt}}_{|L|}\}$ at moment t_{opt}. We use Equation (9) to estimate the long-term impact of performing action $A^s_{l_s}$ at state $S^s_{l_s}$, i.e., the minimum cost of subsequent searches. As shown in Figure 5, $\{S^{opt}_{l^\xi_i}\}$ are the next states that can be generated by performing action $A^s_{l_s}$ (i.e., spatiotemporal searches) at state $S^s_{l_s}$, and $\{A^{opt}_{l^\xi_1}\}$, $\{A^{opt}_{l^\xi_2}\}$, ... are actions that can be performed at state $S^{opt}_{l^\xi_1}$, $S^{opt}_{l^\xi_2}$, ..., respectively. Especially, when the agent finds the vehicle's location at the target moment $t_{t\arg et}$, it no longer needs to perform any action or update the state.

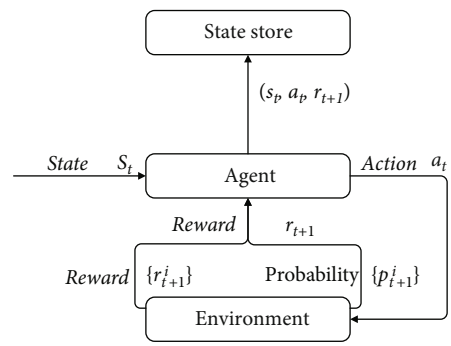


Figure 4: Training of QDP. $s_t$denotes the current state, a_t denotes the action performed at state s_t, r_{t+1} denotes the reward (reward is the inverse of cost) of the action a_t at state s_t, $\{p^i_{t+1}\}$ denotes the probability distribution of next states, and r^i_{t+1} denotes the reward of the optimal action at each possible state.

$$\min_{A^s_{l_s}{}'} Q\left(S^s_{l_s}{}', A^s_{l_s}{}'\right) = \sum_{i=1}^{|L|} p^{opt}_{l^\xi_i} \times \min_{A^{opt}_{l^\xi_i}} Q\left(S^{opt}_{l^\xi_i}, A^{opt}_{l^\xi_i}\right) \tag{9}$$

To solve the problem that QDP is difficult to update the state during training, we adopt the idea of Monte-Carlo method. In each loop, we randomize the current state and action, calculate the expected cost of the subsequent searches by Equation (9) for the value function updated by Equation (8), and iterate the loop until the

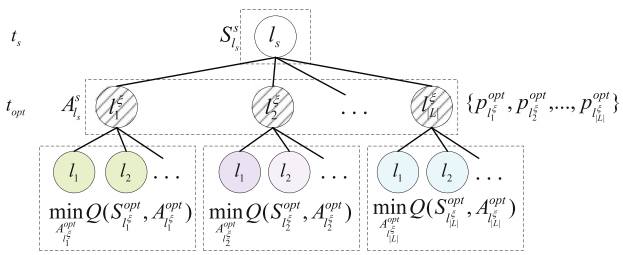


Figure 5: Estimating the subsequent search cost with probabilistic states.

value function converges. The pseudo code of QDP's training method is shown as Algorithm 1.

3.3. Spatiotemporal Searches at Executing Phase. The basic idea of QDP's executing phase is greedy. At each step, the moment decision model decides the optimal searching moment t_{opt} based on known spatiotemporal point (t_s, l_s) of the vehicle o_x, and the location decision model decides the corresponding searching location sequence $\langle t_{opt}, \xi_{opt}\rangle$. We perform spatiotemporal searches in camera records to find the vehicle o_x's location $l_i^{\xi_{opt}}$ at the moment t_{opt}, and update the known spatiotemporal point $(t_s, l_s) = (t_{opt}, l_i^{\xi_{opt}})$. The above steps are iterated until the vehicle o_x's location $l_{t\ \arg\ et}$ at the target moment $t_{t\ \arg\ et}$ is found. Finally, QDP outputs the accumulative search cost, i.e., the cost of spatiotemporal search. The complete steps of spatiotemporal search at QDP's executing phase are as follows:

(1) Quantify the search cost $C = \{c_{l_1}^1, c_{l_2}^1, \cdots, c_{l_{|L|}}^1, \cdots, c_{l_1}^{|d_x|}, c_{l_2}^{|d_x|}, \cdots, c_{l_{|L|}}^{|d_x|}\}$ at different moments and at different locations of the day d_x

(2) Initialize the known spatiotemporal point (t_s, l_s) of the target vehicle o_x and the target moment $t_{t\ \arg\ et}$ in the testing day d_x

(3) Decide the optimal searching moment t_{opt}, $t_{opt} \in (t_s, t_{t\ \arg\ et}]$, according to the moment decision model

(4) Decide the searching location sequence $\langle t_{opt}, \xi_{opt}\rangle$ according to the location decision model

(5) Perform spatiotemporal searches $\langle (s(o_x, t_{opt}, l_1^{\xi_{opt}})), (s(o_x, t_{opt}, l_2^{\xi_{opt}})), \cdots\rangle$ until $s(o_x, t_{opt}, l_i^{\xi_{opt}}) = 1$, then return the vehicle o_x's location $l_i^{\xi_{opt}}$ at moment t_{opt}

(6) Update known spatiotemporal point $(t_s, l_s) = (t_{opt}, l_i^{\xi_{opt}})$

Repeat steps (3), (4), (5), and (6) until $t_s = t_{t\ \arg\ et}$, output the vehicle o_x's location $l_{t\ \arg\ et}$ at the target moment $t_{t\ \arg\ et}$ and the accumulative cost.

4. Evaluation

4.1. Dataset. The experimental dataset (The experimental dataset of this article is provided on demand, please contact the author if necessary) is derived from the trajectories of 19,000 taxis in Chengdu, China, in August 2014. The data acquisition area is about 30KM × 30KM, and the duration of each trajectory is from 7 : 00 am to 21 : 59 pm.

The GPS waypoints of different vehicles in the original dataset are unordered. In addition, some vehicles contain various numbers of waypoints within 1 minute while some contain no waypoints. Therefore, it is firstly necessary to preprocess the dataset. In this experiment, we generate a GPS waypoint sequence of a taxi by filtering the vehicle ID and sorting the timestamps. The time is discretized into moments, each with 1 minute, and only the waypoint with the earliest timestamp per minute indicates the location of the vehicle at that moment. Trajectories with seriously missing data are discarded. The raw data details and a complete trajectory are shown in Table 1.

The data acquisition area is divided into grids each of size 1KM × 1KM, and the GPS waypoints are projected into the grids, and each grid represents a location. The gird size of 1.2KM × 1.2KM and 1.5KM × 1.5KMare also tested in our experiments. Smaller grids are not appropriate since there is enough historical data in a smaller grid. The 99,265 trajectories of the first 14 days are used for training, and randomly selected 1000 trajectories from 21,963 trajectories in the last 3 days are used for testing, i.e., 1000 tests. In this experiment, it is assumed that the search cost C at a spatiotemporal point is 1. Finally, under different grid sizes, we perform 1000 tests and average the results.

4.2. Parameters. Two parameters of QDP need to be tuned: the learning rate α and the discount rate γ in Equation (8). We choose them from practical experiences.

The learning rate can control the update rate of the value function. Small learning rate will reduce the convergence speed of the value function, and a large learning rate may fail to converge to the optimal solution. To find a proper learning rate, we compare the test results (i.e., the accumulative search cost) of QDP with different learning rates ($\alpha = 1$, 0.5, 0.3, 0.1, respectively) under the same other settings (i.e., grid size = 1KM × 1KM, $t_{t\ \arg\ et} - t_s = 30$ min, discount rate $\gamma = 1$, and start moment ts =8 : 00/10 : 00/12 : 00/14 : 00/16 : 00/18 : 00/20 : 00).

The test results verify that the value function is converged with all different learning rates. As shown in Table 2, QDP with different α output the same accumulative cost, which implies their value functions are converged to the same solution. Only the training time is different. The training is performed on a computer with a memory of 16 GB and a processor of Inter(R) Core(TM) i7-6700HQ. To speed up the training, we choose $\alpha = 1$.

Discount rate γ can balance the importance between the immediate and potential reward. $\gamma > 1$ indicates that the immediate reward is more important than the potential reward, and vice versa. To find a proper discount rate, we compare the test results of QDP with different discount rates

Input: $t_{t\ \arg\ et}$, C, α, γ, TR' *before* d_x
Output: value function Q
1 Calculate the vehicle transition probability $\{P(t_s, l_s, t_{opt}, :)\}$ at $\{(t_s, l_s)\}$ and at the moment $\{t_{opt}\}$, $t_{opt} \in (t_s, t_{t\ \arg\ et}]$ according to TR'; $//P(t_s, l_s, t_{opt}, :) = \{p_{l_1}^{t_{opt}}, p_{l_2}^{t_{opt}}, \cdots, p_{l_{|L|}}^{t_{opt}}\} = TPM$
2 Calculate the expected cost $\{C(t_s, l_s, t_{opt})\}$ of spatiotemporal searches according to Equation (6); $//C(t_s, l_s, t_{opt}) = C^{opt} = Cost(t_{opt}, \xi_{opt})$
3 Initialization value function Q;
4 **while** Q does not converge:
5 Randomize spatiotemporal point (t_s, l_s) and the moment decision t_{opt}, $t_{opt} \in (t_s, t_{t\ \arg\ et}]$;
6 Calculate $\min_{A_{l_s}^{s\ \prime}} Q(S_{l_s}^{s\ \prime}, A_{l_s}^{s\ \prime})$ according to Equation (9);
7 Update the value function $Q((t_s, l_s), t_{opt})$ according to Equation (8);
8 **end.**

Algorithm 1: QDP's training method

Table 1: Detailed description of the dataset.

(a) Raw data details

Field name	Example	Remarks
Vehicle ID	1	\
Passenger or not	0/1	\
Timestamp	1501584540	Unix timestamp, in seconds
Longitude	104.042833	GCJ-02
Latitude	30.599851	GCJ-02

(b) Complete trajectory of a vehicle in a day

Vehicle	Day	Longitude	Latitude	Moment
		104.039163	30.597572	7 : 00
		104.126565	30.599733	7 : 01
o_x	2014/8/4	...	...	...
		104.042833	30.599851	21 : 58
		104.127154	30.600009	21 : 59

(γ=0.97, 0.99, 1, 1.01, 1.03, 1.05, respectively) under the same other settings (i.e., grid size = 1KM × 1KM, $t_{t\ \arg\ et} - t_s = 30$ min, learning rate $\alpha = 1$, and start moment ts traverses every moment from 7 : 00 to 21 : 29 of a day).

As shown in Table 3, when the discount rate $\gamma = 1$, the search cost of QDP is the smallest, so we set the discount rate γ to 1.

4.3. Baselines. In this section, we will describe three spatiotemporal searching algorithms as baselines: ALT, IEM, and IHMs [5]. Given the location l_s of the vehicle o_x at moment t_s of d_j, the process of the three algorithms are as follows:

4.3.1. All Searching at the Last Time (ALT). Calculate the probability $(p_{l_1}^{t\ \arg\ et}, p_{l_2}^{t\ \arg\ et}, \cdots)$ that the vehicle o_x moves to different locations at moment $t_{t\ \arg\ et}$ according to the spatiotemporal point (t_s, l_s) and the historical trajectories TR'; Determine the searching location sequence $\langle t_{t\ \arg\ et}, \xi_{t\ \arg\ et}\rangle$ according to the location decision model; Perform spatiotemporal searches $\langle (s(o_x, t_{t\ \arg\ et}, l_1^{\xi_{t\ \arg\ et}})), (s(o_x, t_{t\ \arg\ et}, l_2^{\xi_{t\ \arg\ et}})), \cdots\rangle$ until $s(o_x, t_{t\ \arg\ et}, l_i^{\xi_{t\ \arg\ et}}) = 1$, then return the location $l_{t\ \arg\ et} = l_i^{\xi_{t\ \arg\ et}}$ of vehicle o_x at moment $t_{t\ \arg\ et}$.

4.3.2. Intermediate Searching at an Estimated Moment (IEM). Calculate probability $(p_{l_1}^{opt}, p_{l_2}^{opt}, \cdots)$ that the vehicle o_x moves to different locations at moment t_{opt}, $t_{opt} \in (t_s, t_{t\ \arg\ et})$ according to the spatiotemporal point (t_s, l_s) and the historical trajectories TR'; Determine the searching location sequence $\langle t_{opt}, \xi_{opt}\rangle$ according to the location decision model; Calculate probability $\{(p_{l_1}^{t\ \arg\ et}, p_{l_2}^{t\ \arg\ et}, \cdots)\}$ that the vehicle o_x moves to different locations at moment $t_{t\ \arg\ et}$ according to the spatiotemporal point $\{(t_{opt}, l_i^{\xi_{opt}})\}$ and TR'; Determine the searching location sequence $\{\langle t_{t\ \arg\ et}, \xi_{t\ \arg\ et}\rangle\}$. The above two steps are only used for cost estimation of different t_{opt}, but no need to perform the searches actually. Determine an intermediate search moment t_{opt} according to Equation (10); Perform spatiotemporal searches $\langle (s(o_x, t_{opt}, l_1^{\xi_{opt}})), (s(o_x, t_{opt}, l_2^{\xi_{opt}})), \cdots\rangle$, until $s(o_x, t_{opt}, l_i^{\xi_{opt}}) = 1$, update to spatiotemporal point $(t_{opt}, l_i^{\xi_{opt}})$; Perform spatiotemporal searches $\langle (s(o_x, t_{t\ \arg\ et}, l_1^{\xi_{t\ \arg\ et}})), (s(o_x, t_{t\ \arg\ et}, l_2^{\xi_{t\ \arg\ et}})), \cdots\rangle$ which correspond to the spatiotemporal point $(t_{opt}, l_i^{\xi_{opt}})$, until $s(o_x, t_{t\ \arg\ et}, l_i^{\xi_{t\ \arg\ et}}) = 1$, then return the location $l_{t\ \arg\ et} = l_i^{\xi_{t\ \arg\ et}}$ of the vehicle o_x at the moment $t_{t\ \arg\ et}$.

$$t_{opt} = \underset{t_{opt} \in (t_s, t_{t\ \arg\ et})}{\arg\min} \left(Cost(t_{opt}, \xi_{opt}) + \sum_{i=1}^{|L|} p_{l_i}^{opt} \times Cost(t_{t\ \arg\ et}, \xi_{t\ \arg\ et}) \right) \tag{10}$$

4.3.3. Intermediate Searching at Heuristic Moments (IHMs). Calculate the probability $\{(p_{l_1}^{opt}, p_{l_2}^{opt}, \cdots)\}$ of o_x moving to

Table 2: Accumulative search cost of QDP with different learning rates.

Start moment	Training time	8 : 00	10 : 00	12 : 00	14 : 00	16 : 00	18 : 00	20 : 00
$\alpha = 1.0, \gamma = 1$	**11.50** h	32.81	32.90	34.85	32.72	34.06	30.15	34.97
$\alpha = 0.5, \gamma = 1$	**30.36** h	32.81	32.90	34.85	32.72	34.06	30.15	34.97
$\alpha = 0.3, \gamma = 1$	**54.12** h	32.81	32.90	34.85	32.72	34.06	30.15	34.97
$\alpha = 0.1, \gamma = 1$	**98.34** h	32.81	32.90	34.85	32.72	34.06	30.15	34.97

Table 3: Accumulative search cost of QDP with different discount rates.

Start moment	8 : 00	10 : 00	12 : 00	14 : 00	16 : 00	18 : 00	20 : 00
$\alpha = 1, \gamma = 0.97$	34.38	35.05	36.51	34.86	35.57	32.44	36.61
$\alpha = 1, \gamma = 0.99$	33.25	33.42	35.01	33.27	34.31	30.46	35.11
$\alpha = 1, \gamma = 1.00$	**32.81**	**32.90**	**34.85**	**32.72**	**34.06**	**30.15**	**34.97**
$\alpha = 1, \gamma = 1.01$	33.59	32.95	34.91	33.08	33.72	30.18	35.14
$\alpha = 1, \gamma = 1.03$	33.15	33.19	35.79	33.24	34.10	30.36	35.13
$\alpha = 1, \gamma = 1.05$	33.68	33.64	36.17	33.92	34.96	31.21	35.79

different locations at moment $\{t_{opt}\}$, $t_{opt} \in (t_s, t_{t\ \arg\ et}]$ according to the spatiotemporal point (t_s, l_s) and the historical trajectories TR'; Determine the searching location sequence $\{\langle t_{opt}, \xi_{opt}\rangle\}$ according to the location decision model; Determine the intermediate search moment t_{opt} according to Equation (11); Perform spatiotemporal searches $\langle (s(o_x, t_{opt}, l_1^{\xi_{opt}})), (s(o_x, t_{opt}, l_2^{\xi_{opt}})), \cdots\rangle$, until $s(o_x, t_{opt}, l_i^{\xi_{opt}}) = 1$; Update spatiotemporal point $(t_s, l_s) = (t_{opt}, l_i^{\xi_{opt}})$; Iterate the above steps until $t_s = t_{t\ \arg\ et}$, and return the location $l_{t\ \arg\ et} = l_i^{\xi_{opt}}$ of the vehicle o_x at the target time $t_{t\ \arg\ et}$.

$$t_{opt} = \underset{t_{opt} \in (t_s, t_{t\ \arg\ et}]}{\arg\min} \frac{Cost(t_{opt}, \xi_{opt})}{t_{opt} - t_s} \tag{11}$$

In summary, ALT searches only at the target moment, IEM searches at an intermediate moment and the target moment, while IHMs and QDP search at multiple intermediate moments as well as the target moment. The baselines and QDP use the same location decision model as described in Section 3.1, which is not the focus of this paper. The main difference among them is how to decide the searching moment.

4.4. Comparison of Different Algorithms. This experiment compares four spatiotemporal searching algorithms: ALT, IEM, IHMs and QDP. When testing, instead of real camera records, we use the testing trajectories as the source of ground truth, which also can enable the functionality of Equation (1). The performance metric is the average search cost of the 1000 tests. As shown in Figure 6, QDP is better than the other three spatiotemporal searching algorithms in general. The four spatiotemporal searching algorithms have different performance under different grid sizes (1KM × 1KM, 1.2KM × 1.2KM, and 1.5KM × 1.5KM, respectively) and time intervals (i.e., $t_{t\ \arg\ et} - t_s$= 30 min, 40 min, 50 min, respectively). The start moment ts traverses every moment from 7 : 00 to "21:59 - time_span" of the corresponding testing day.

(1) When the grid size is 1KM × 1KM, IHMs and QDP are better than ALT and IEM at different time intervals. As the time interval becomes bigger, the advantage of QDP is increasing compared with IHMs

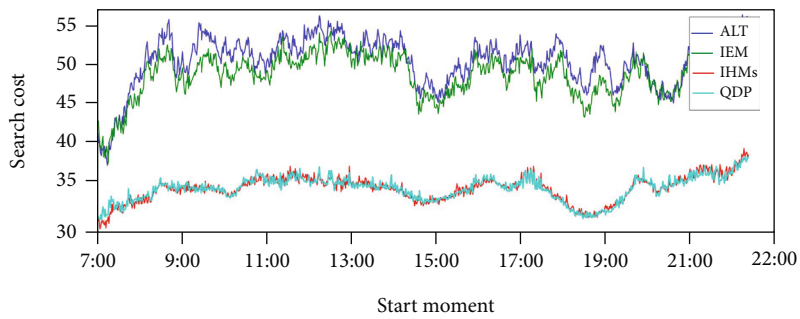
(2) When the grid size is 1.2KM × 1.2KM, IHMs and QDP are better than ALT when the time interval is 30 min or 40 min. There is no significant difference among ALT, IHMs, and QDP when the time interval is 50 min, but they are better than IEM

(3) When the grid size is 1.5KM × 1.5KM, IHMs and QDP are better than ALT when the time interval is 30 min. There is no significant difference among ALT, IHMs, and QDP when the time interval is 40 min or 50 min, but they are better than IEM

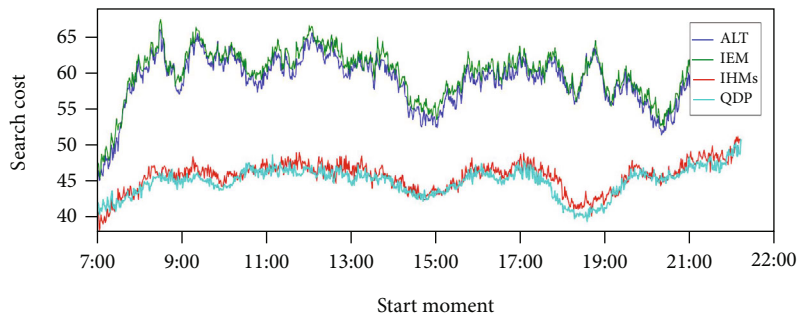
No matter what the grid size and time interval is, QDP is always better than or equal to IHMs.

4.5. Analysis and Discussion. We perform more analysis to disclose the reason of the above results. Remind that the main difference among ALT, IEM, IHMs, and QDP is how to decide the searching moment t_{opt}, i.e., ALT searches only at the target moment, IEM searches at an intermediate moment and the target moment, while IHMs and QDP search at multiple intermediate moments as well as the target moment. We intend to demonstrate the efficiency of different moment decisions. Because the search cost will increase as the timespan $\Delta t = t_{opt} - t_s$ becomes larger, the cost-timespan ratio (also the heuristic indicator of IHMs) is defined as the ratio of the estimated search cost to the timespan:

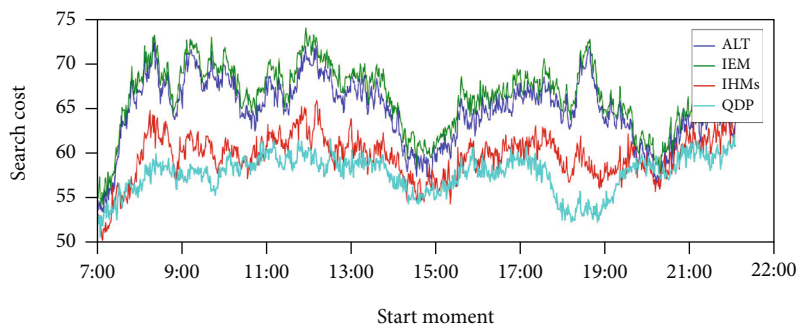
$$CostTimespanRatio(t_{opt} - t_s) = \frac{Cost(t_{opt}, \xi_{opt})}{t_{opt} - t_s} \tag{12}$$



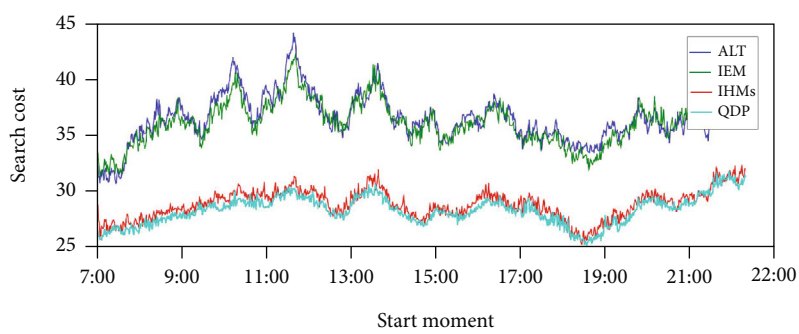
(a) grid size = 1KM × 1KM, target moment - start moment = 30 min



(b) grid size = 1KM × 1KM, target moment - start moment = 40 min

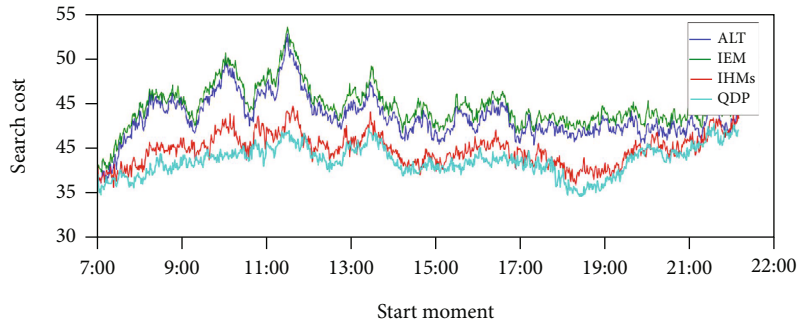


(c) grid size = 1KM × 1KM, target moment - start moment = 50 min

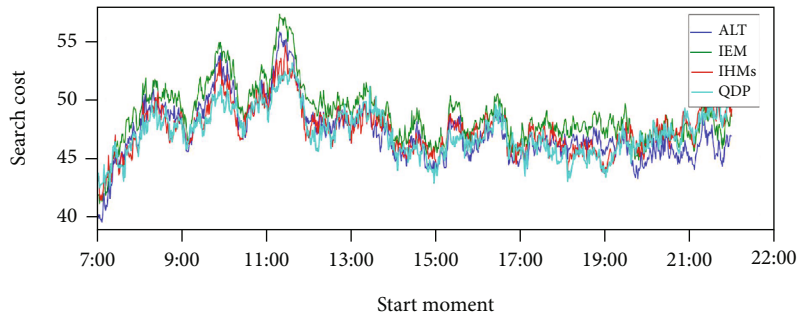


(d) grid size = 1.2KM × 1.2KM, target moment - start moment = 30 min

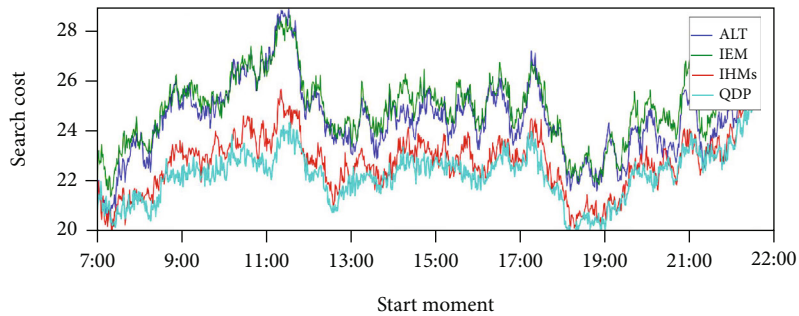
Figure 6: Continued.



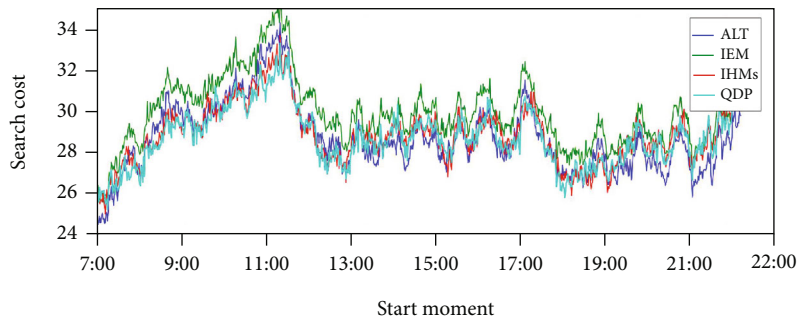
(e) grid size = 1.2KM × 1.2KM, target moment - start moment = 40 min



(f) grid size = 1.2KM × 1.2KM, target moment - start moment = 50 min

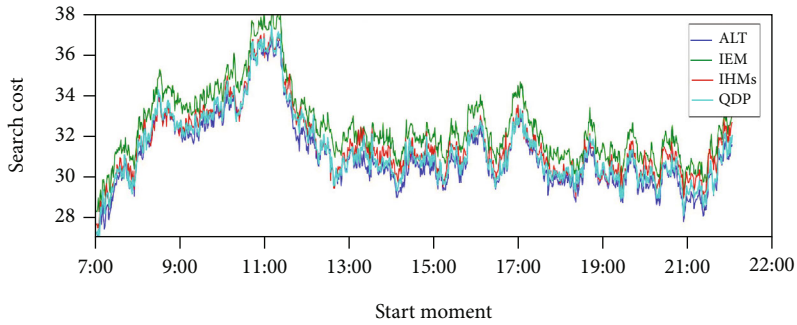


(g) grid size = 1.5KM × 1.5KM, target moment - start moment = 30 min



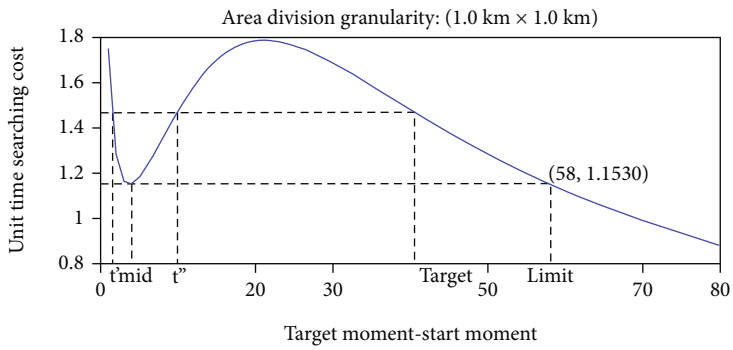
(h) grid size = 1.5KM × 1.5KM, target moment - start moment = 40 min

Figure 6: Continued.

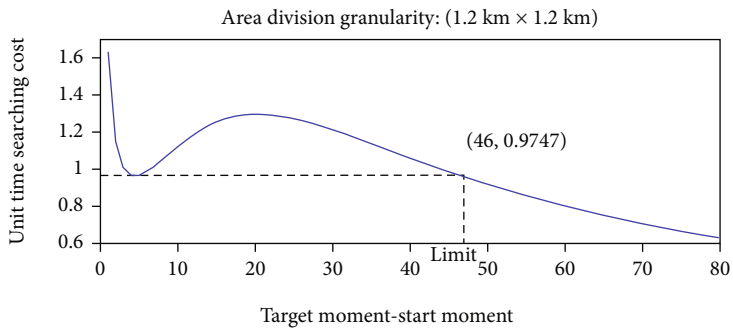


(i) grid size = 1.5KM × 1.5KM, target moment - start moment = 50 min

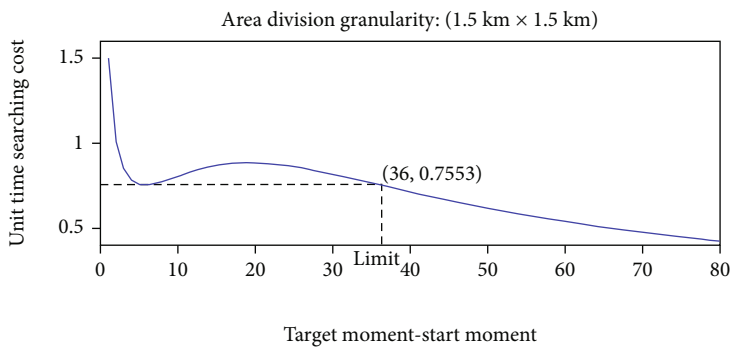
Figure 6: Comparison of four spatiotemporal searching algorithms.



(a) grid size = 1KM × 1KM



(b) grid size = 1.2KM × 1.2KM



(c) grid size = 1.5KM × 1.5KM

Figure 7: Cost-timespan ratio of different timespans.

Figure 7 shows the cost-timespan ratio under different grid sizes (1KM × 1KM, 1.2KM × 1.2KM, 1.5KM × 1.5KM). All three ratio's curves first decrease, then increase, and then decrease along with the timespan. Let preferred timespan Δt_{mid} be the timespan of the first locally minimum ratio CostTimespanRatio(Δt_{mid}). A contour timespan $\Delta t_{\lim it}$ is the timespan whose cost-timespan ratio is equal to CostTimespanRatio(Δt_{mid}).

If $\Delta t < \Delta t_{mid}$ or $\Delta t > \Delta t_{\lim it}$, the larger timespan is more efficient, thus the searching moment should be as later as possible, i.e., searching at intermediate moments cannot help to reduce the total search cost, and ALT will be better than IEM. In this situation, IHMs and QDP will also only perform searches at the target moment, which are equivalent with ALT.

If Δt is between (Δt_{mid}, $\Delta t_{\lim it}$), since CostTimespanRatio(Δt_{mid}) is the minimum ratio when timespan is from 0 to Δt. As a result, Δt_{mid} is the most efficient timespan which can decide the corresponding intermediate moment, i.e., searching at intermediate moments can reduce the total search cost, and IEM, IHMs and QDP may be better than ALT. Whether the right intermediate moments are detected is up to specific algorithms.

The different grid sizes will cause $\Delta t_{\lim it}$ to be different.$\Delta t_{\lim it}$ corresponding to grid sizes 1KM × 1KM, 1.2KM × 1.2KM, and 1.5KM × 1.5KM are calculated to be 58 min, 46 min and 36 min, respectively. Therefore, the reason of the results in Section 4.4 can be explained as:

(1) When the grid size = 1KM × 1KM, In Figures 6(a), 6(b), 6(c), since the timespans Δt = 30 min/40 min/50 min do not exceed $\Delta t_{\lim it}$ (58 min), IHMs and QDP are better than ALT

(2) When the grid size = 1.2KM × 1.2KM, Since the timespans Δt = 30 min/40 min (Figures 6(d), 6(e)) do not exceed $\Delta t_{\lim it}$ (46 min), IHMs and QDP are better than ALT. While when the timespan Δt = 50 min (Figure 6(f)) exceeds $\Delta t_{\lim it}$ (46 min), there is no significant difference among ALT, IHMs, and QDP

(3) When the grid size = 1.5KM × 1.5KM, Since the timespan Δt = 30 min (Figure 6(g)) does not exceed $\Delta t_{\lim it}$ (36 min), IHMs and QDP are better than ALT. While when the timespans Δt = 40 min/50 min exceed $\Delta t_{\lim it}$ (36 min), there is no significant difference among ALT, IHMs, and QDP

IEM is bound to search at an intermediate moment and cannot make a decision dynamically to adapt environment information. Therefore, IEM is almost the worst in all situations. QDP can surmount the shortage that IHMs may output a locally optimal solution. As Q-learning does, it finds the near global optimal solution in the sense that it minimizes the total cost of any and all successive searches.

5. Conclusions

To minimize the cost of spatiotemporal search, this paper proposes a reinforcement learning algorithm called QDP. QDP selects the next searching moment based on known vehicle's spatiotemporal point. The outcome of these searches is a new known vehicle's spatiotemporal point, which guides the next selection iteratively. To address the challenge of probabilistic state in the training phase, we propose a novel training method for QDP, which is based on Monte-Carlo method and probabilistic states. QDP replaces the next state by multiple states of a probability distribution and estimates the expected cost of subsequent actions to calculate the value function. Finally QDP creates a state and an action randomly in each loop to train the value function progressively.

We evaluate QDP on a real-world vehicle trajectory data and compare it with baseline algorithms ALT, IEM, IHMs under different grid sizes (1KM × 1KM, 1.2KM × 1.2KM, 1.5KM × 1.5KM) and different time intervals (30 min, 40 min, 50 min). The experimental results show that QDP is better than the baseline. In addition, with the grid size decreasing, the preferred timespan Δt_{mid} is also decreasing, but the contour timespan $\Delta t_{\lim it}$ is increasing. Compared with ALT, algorithms with adaptive intermediate moment selection (i.e., IHMs and QDP) can effectively reduce the total search cost when the timespan is between Δt_{mid} and $\Delta t_{\lim it}$. Compared with IHMs, QDP takes into account both the cost efficiency of the current selection (as well as performing search at that moment) and its potential impact on subsequent selections, and gives an approximate lower bound of the cost of spatiotemporal search to a certain extent.

In the future, we will test QDP with other datasets, e.g., trajectories of pedestrians, bicycles, or ride sharing cars [34, 35]. We also wish to build new models that can take consideration of both the location decision model and the moment decision model simultaneously, or can output directly a candidate spatiotemporal point to search for next. And deep reinforcement learning algorithms are worth to explore for this problem.

Notations

d_j:	a day, $d_j \in D$; d_x is the testing day
o_x:	the target vehicle, $o_x \in O$
l_i:	a location
t_k:	a moment, $d_j = \langle t_1, t_2, \cdots, t_{\lvert d_j \rvert} \rangle$
(t_s, l_s):	a known spatiotemporal point
t_{opt}:	the intermediate search moment
$t_{t\ \arg\ et}$:	the target moment
$l_{t\ \arg\ et}$:	o_x's location at $t_{t\ \arg\ et}$
$tr(o_x, d_j)$:	the trajectory of o_x in d_j, $tr(o_x, d_j) \in$TR
TR':	the training trajectories
$c_{l_i}^{opt}$:	the cost of searching at (t_{opt}, l_i)
$\langle t_{opt}, \xi_{opt} \rangle$:	the searching location sequence at moment t_{opt}, i.e., the location decision
$l_i^{\xi_{opt}}$:	o_x's location at t_{opt}, assume it is the *i-th* location of $\langle t_{opt}, \xi_{opt} \rangle$
$p_{l_i}^{opt}$:	the probability that o_x will be at l_i at t_{opt}

$TPM_{\Delta t}^{|L|\times|L|}$: the transfer probability matrix; an element is $p_{l_i}^{opt}$

$Cost(t_{opt}, \xi_{opt})$: the expected search cost of $\langle t_{opt}, \xi_{opt}\rangle$

$S_{l_s}^{s}$: a state, also denoted as (t_s, l_s)

$A_{l_s}^{s}$: an action performed under $S_{l_s}^{s}$, i.e., $\langle t_{opt}, \xi_{opt}\rangle$

$S_{l_i^{\xi}}^{opt}$: a probabilistic state after $A_{l_s}^{s}$, also denoted as $(t_{opt}, l_i^{\xi_{opt}})$

$A_{l_i^{\xi}}^{opt}$: an action performed under $S_{l_i^{\xi}}^{opt}$.

Data Availability

The experimental dataset[The experimental dataset of this article is provided on demand, please contact the author if necessary.] is derived from the trajectories of 19,000 taxis in Chengdu, China, in August 2014. The data acquisition area is about, and the duration of each trajectory is from 7 : 00 am to 21 : 59 pm.

Conflicts of Interest

Please note the author declared that there is no conflict of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant No. 61772136, and the Fujian Engineering Research Center of Big Data Analysis and Processing.

References

[1] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle Re-identification for urban surveillance," in *ECCV 2016: Computer Vision – ECCV 2016*, pp. 869–884, Springer, 2016.

[2] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang, "Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals," in *IEEE International Conference on Computer Vision*, Venice, Italy, 2017.

[3] Y. Jin, C. Li, Y. Li, P. Peng, and G. A. Giannopoulos, "Model latent views with multi-center metric learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1919–1931, 2021.

[4] X. Liu, W. Liu, T. Mei, and H. Ma, "PROVID: Progressive and Multimodal Vehicle Reidentification for Large-Scale Urban Surveillance," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 645–658, 2018.

[5] Z. Yu, L. Han, C. Chen, W. Guo, and Z. Yu, "Object Tracking by Least Spatiotemporal Searches," vol. 8, no. 16, pp. 12934–12946, 2020.

[6] Puterman and L. Martin, *Markov Decision Process. Simulation-Based Algorithms for Markov Decision Processes*, Springer, London, 2013.

[7] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[8] F. S. Melo and M. I. Ribeiro, "Convergence of Q-learning with linear function approximation," in *2007 European Control Conference (ECC)*, Kos, Greece, 2007.

[9] B. O. Koopman, "The theory of search. I. Kinematic bases," *Operations Research*, vol. 4, no. 3, pp. 324–346, 1956.

[10] B. O. Koopman, "The theory of search. II. Target detection," *Operations Research*, vol. 4, no. 5, pp. 503–531, 1956.

[11] B. O. Koopman, "The Theory of search," *Operations Research*, vol. 5, no. 5, pp. 613–626, 1957.

[12] L. D. Stone, *Theory of Optimal Search*, Academic Press, 1975.

[13] L. D. Stone, "Necessary and sufficient conditions for optimal search plans for moving targets," *Mathematics of Operations Research*, vol. 4, no. 4, pp. 431–440, 1979.

[14] J. Berger, N. Lo, and M. Noel, "Exact Solution for Search-and-Rescue Path Planning," *International Journal of Computer and Communication Engineering*, vol. 2, pp. 266–271, 2013.

[15] N. Lo, J. Berger, and M. Noel, "Toward optimizing static target search path planning," in *2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications*, Ottawa, ON, Canada, 2012.

[16] K. S. Tseng and B. Mettler, "Near-optimal probabilistic search via submodularity and sparse regression," *Autonomous Robots*, vol. 41, no. 1, pp. 205–229, 2017.

[17] J. Berger and N. Lo, "An innovative multi-agent search-and-rescue path planning approach," *Computers & Operations Research*, vol. 53, pp. 24–31, 2015.

[18] E. Bensana, G. Verfaillie, J. C. Agnese, N. Bataille, and D. Blumstein, "Exact &INEXACT Methods for Daily Management of Earth Observation Satellite," *Space Mission Operations and Ground Data Systems-SpaceOps' 96*, vol. 394, p. 507, 1996.

[19] E. Bensana, M. Lemaitre, and G. Verfaillie, "Earth observation satellite management," *Constraints*, vol. 4, no. 3, pp. 293–299, 1999.

[20] T. Benoist, "Towards optimal formwork pairing on construction sites," *RAIRO - Operations Research*, vol. 41, no. 4, pp. 381–398, 2007.

[21] M. Lemaître, G. Verfaillie, F. Jouhaud, J. M. Lachiver, and N. Bataille, "Selecting and scheduling observations of agile satellites," *Aerospace Science and Technology*, vol. 6, no. 5, pp. 367–381, 2002.

[22] F. Wörgötter and B. Porr, "Reinforcement learning," *Approximate Dynamic Programming & Reinforcement Learning*, vol. 3, no. 3, pp. 804–809, 2008.

[23] A. Plaat, *Deep Reinforcement Learning*, Springer Nature Singapore, 2022.

[24] A. Jevtić, A. Colomé, G. Alenyà, and C. Torras, "Robot motion adaptation through user intervention and reinforcement learning," *Pattern Recognition Letters*, vol. 105, pp. 67–75, 2018.

[25] A. W. Moore, "Variable Resolution Dynamic Programming: Efficiently Learning Action Maps in Multivariate Real-valued State-spaces," in *Machine Learning Proceedings 1991*, pp. 333–337, Evanston, Illinois, 1991.

[26] R. H. Crites and A. G. Barto, "Elevator group control using multiple reinforcement learning agents," *Machine Learning*, vol. 33, no. 2/3, pp. 235–262, 1998.

[27] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[28] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[29] B. J. A. Kroese, "Learning from delayed rewards," *Robotics and Autonomous Systems*, vol. 15, no. 4, pp. 233–235, 1995.

[30] G. A. Rummery and M. Niranjan, *On-Line Q-Learning Using Connectionist Systems*, University of Cambridge, Cambridge, Department of Engineering, 1994.

[31] H. van Seijen, H. van Hasselt, S. Whiteson, and M. Wiering, "A theoretical and empirical analysis of Expected Sarsa," in *2009 IEEE symposium on adaptive dynamic programming and reinforcement learning*, pp. 177–184, Nashville, TN, USA, 2009.

[32] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforcement learning," 2013, http://arxiv.org/abs/1312.5602.

[33] J. Chen, *One-way optimal search theory (in Chinese)*, National Defense Industry Press, 2016.

[34] C. Chen, Y. Ding, X. Xie, S. Zhang, Z. Wang, and L. Feng, "TrajCompressor: an online map-matching-based trajectory compression framework leveraging vehicle heading direction and change," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 2012–2028, 2020.

[35] S. Guo, C. Chen, J. Wang et al., "ROD-revenue: seeking strategies analysis and revenue prediction in ride-on-demand service using multi-source urban data," *IEEE Transactions on Mobile Computing*, vol. 19, no. 9, pp. 2202–2220, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3627246, 9 pages
https://doi.org/10.1155/2022/3627246

Research Article

Vehicle License Plate Recognition Using Shufflenetv2 Dilated Convolution for Intelligent Transportation Applications in Urban Internet of Things

Xiufeng Li,[1] **Zheng Wen**,[2] **and Qiaozhi Hua**[3]

[1]*Department of Computer Science, Changzhi University, Changzhi 046011, China*
[2]*School of Fundamental Science and Engineering, Waseda University, Tokyo 169-8050, Japan*
[3]*Computer School, Hubei University of Arts and Science, Xiangyang 441000, China*

Correspondence should be addressed to Qiaozhi Hua; alex2441@163.com

Received 27 February 2022; Revised 3 April 2022; Accepted 16 April 2022; Published 19 May 2022

Academic Editor: Han Liu

Intelligent transportation applications based on urban Internet of Things can improve the efficiency of government services and promote urban modernization. As smart cameras are more and more widely used in cities, artificial intelligence technology is an important force to achieve license plate recognition. An efficient license plate recognition algorithm not only improves the efficiency of traffic management but also saves management costs. This paper proposes a network based on the shufflenetv2 dilated convolution (SDC) model, which includes two parts: license plate location and license plate recognition. SDC model adopts shufflenetv2 as the backbone network, which combines dilated convolution and global context blocks. Therefore, the receptive field and feature expression ability of the model are enhanced. For license plate location, CIOU loss considers not only the coverage area of the bounding box but also the center distance and aspect ratio. For license plate recognition, CTC loss trains the network based on the sequence and solves the sample alignment problem, which improves the accuracy of license plate recognition. The experiments show that the precision of the SDC model in license plate location is 98.7%, which is 5.2%, 5.5%, and 4.1% higher than the precision of Faster-RCNN, YOLOv3, and SSD, respectively. The precision of the SDC model in license plate recognition is 98.2%, which is 5.3%, 3.7%, and 2.9% higher than the precision of LPRNet, AlexNet, and RPNet, respectively.

1. Introduction

Intelligent transportation is an important foundation and guarantee for national economic and social development [1]. With the innovation of Internet of Things and big data [2, 3], it is very important to further promote the digitization and intelligence of transportation industry, which is conducive to promoting the better and faster development of transportation industry. Due to the advancement of social modernization and the improvement in people's quality of life, the number of intelligent cameras and vehicles is growing rapidly, which brings great challenges to traffic management cost [4, 5]. Therefore, accurately obtaining traffic data and constructing traffic data processing model is the premise of building intelligent transportation, and this problem can be solved by big data technology [6].

Based on the Internet of Things, intelligent transportation can improve not only traffic quality but also the efficiency of traffic management [7], using advanced video monitoring equipment and intelligent identification methods to increase the temporal, spatial, and scope management, which can continuously improve the fineness of transportation systems. As the basic work of intelligent transportation, the license plate recognition technology based on deep learning has laid a good foundation for the next analysis and decision-making of the Internet of Things.

In the actual scene, there are many complex objects in the captured images, such as people and vehicles. Before

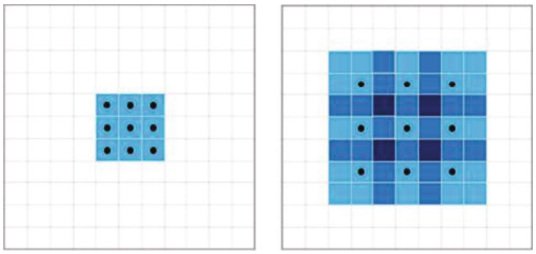


Figure 1: Comparison of convolution and dilated convolution.

we recognize the license plate, we need to locate the vehicle in the image. Traditional license plate location methods include methods based on edge detection [8], color features [9], and mathematical morphology [10]. However, the above methods are greatly affected by the external environment and image quality. Du et al. [11] proposed the SSD model with VGG19 as the basic network. SSD predicts the position offset between each bounding box and the ground-truth box. However, SSD method has a situation in which the position offset is too large and beyond the range of the image. Redmon et al. [12] proposed the Yolov3 model, which was built with darknet-53 as the backbone. YOLOv3 model controls the range from 0 to 1 by adding sigmoid activation to the predicted position offset, which solves the problem of an excessively large position offset. These two methods are types of one-stage methods [13], which have the advantage of fewer calculations and can save time, but they lack accuracy. Ren et al. [14] proposed the Faster-RCNN model, which is a type of two-stage method. It has great advantages in terms of accuracy, but it requires considerable computation and time [15].

At present, license plate recognition algorithms include template matching [16] and feature analysis matching methods [17]. The template matching method adjusts the license plate characters according to the size and matches the template characters in all sample libraries. The feature analysis matching method extracts the features and discriminates between the results according to the number and shape of the character contour. However, they are obviously affected by illumination, noise, and character occlusion. Zherzdev et al. [18] proposed the LPRNet model, which does not need to segment characters and effectively solves the gradient problem. However, the recognition accuracy of complex situations is not high, and it easily decreases during training. Xiu et al. [19] proposed an end-to-end license plate recognition algorithm based on AlexNet model. The AlexNet model uses overlapping pooling, local normalization, and dropout methods to improve the accuracy of the model. But the convolution and pooling of AlexNet model in the training process will cause the loss of features. Xu et al. [20] proposed the RPNet model, whose feature map is shared and the loss function is jointly optimized. However, the lack of the spatial information and receptive field of the model leads to incomplete feature extraction.

To solve the accuracy problem of the above algorithm, this paper proposes a network based on the shufflenetv2 [21] dilated convolution (SDC) model:

(1) SDC model adopts shufflenetv2 as the backbone network, which combines dilated convolution and global context blocks. Therefore, the receptive field and feature expression ability of the model are enhanced

(2) For license plate location, CIOU loss considers not only the coverage area of the bounding box but also the center distance and aspect ratio. For the license plate recognition, CTC loss trains the network based on the sequence and solves the sample alignment problem, which improves the accuracy of license plate recognition

2. Materials and Methods

For the problems regarding about the accuracy of license plate recognition [22], this paper proposes a network based on the shufflenetv2 dilated convolution (SDC) model, which includes two parts: license plate location and license plate recognition.

2.1. Dilated Convolution. Because vehicle recognition systems in practical applications face different scenarios (e.g., road traffic and high-speed charge stations) [23], different proportions of license plate sizes result in the whole picture. In the face of these complex situations, the use of a standard convolution receptive field cannot solve our problem.

Dilated convolution adds dilation based on an ordinary convolution [24], which increases the size of receptive fields in the calculation process. The dilation rate is used to control the interval between points of the convolution kernel. In Figure 1, the left figure shows a standard 3×3 convolution kernel, which corresponds to the dilation rate = 1, with the receptive field represented as the blue region. The left figure represents the dilated convolution, which corresponds to the dilation rate = 2. The receptive field is enlarged by 7×7, which can make the coverage of the convolution larger to add the receptive field.

The calculation of the dilated convolution receptive field is shown below:

$$F_{i+1} = \left(2^{i+2} - 1\right) \times \left(2^{i+2} - 1\right), \tag{1}$$

where F_{i+1} indicates the receptive field and $i + 1$ indicates the dilation rate.

2.2. License Plate Location. The conventional license plate location algorithm, including the one-stage method and the two-stage method, cannot achieve a good balance of speed and accuracy in the face of complex situations. Therefore, this paper uses the anchor-free location method based on four angles, and the calculation method is shown below:

$$f(x, y) = A \exp\left(-\left(\frac{(x - x_0)^2}{2\sigma_x^2} + \frac{(y - y_0)^2}{2\sigma_y^2}\right)\right) \tag{2}$$

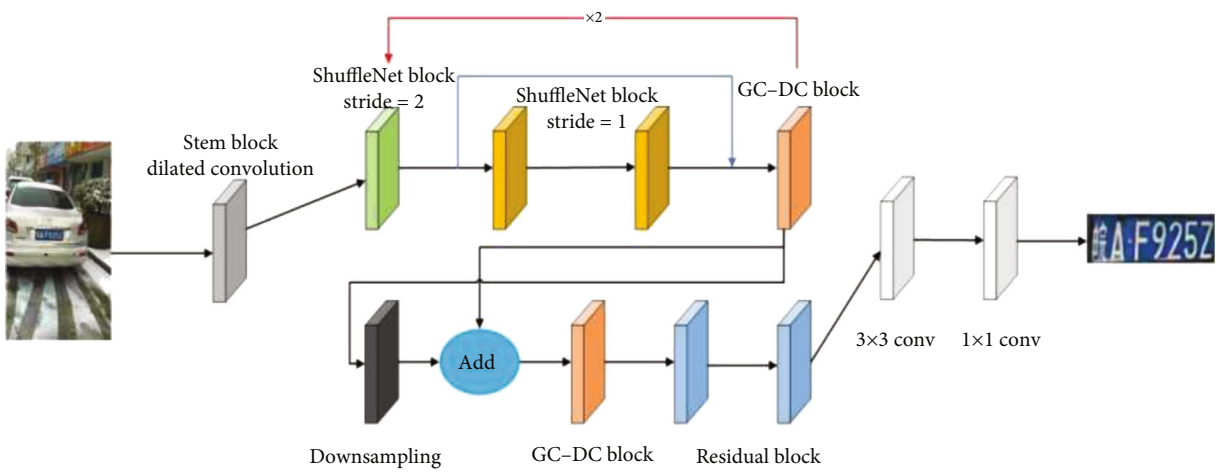


Figure 2: Structure of the license plate location network.

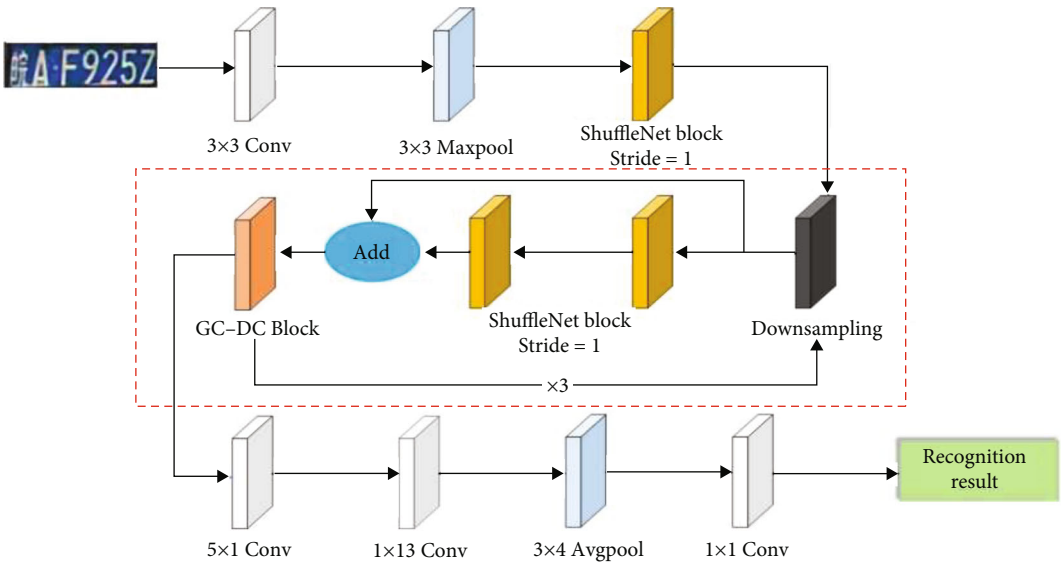


Figure 3: Structure of the license plate recognition network.

where A is the amplitude, $A = 1.(x, y)$ is the predicted coordinate of the angle, and (x_0, y_0) is the ground-truth coordinate. σ_x and σ_y represent the variance.

This paper uses the lightweight network shufflenetv2 to build the model, which reduces the training time. The lightweight ShuffleNetv2 network contains multiple Shuffle blocks. The channel split operation in the Shuffle block divides the number of feature channels into two branches, which reduces the number of parameters in each branch and improves the operation speed. Combining global context blocks and dilated convolution enhances the spatial information of the model and increases the receptive field of the model.

In Figure 2, the gray block represents the stem block with dilated convolution [25]. The stem block utilizes three 3×3 convolutions instead of a large-scale convolution to reduce the loss of information and is combined with dilated convolution to increase the receptive field size. The loop module includes three shuffle blocks and a global context-dilated convolution (GC-DC) block [26]. The green block represents the shuffle block with $stride = 2$, which is used to compress the width and height of the feature layer. The Shuffle block with $stride = 2$ not only reduces the amount of calculation, but also retains more feature information to improve the effect of feature extraction. The yellow block represents the shuffle block with $stride = 1$, which is used to deepen the number of layers of the network. The orange block is the GC-DC block, which represents the global context block with dilated convolution. The blue line represents the jump connection, and the red line indicates the loop operation. The loop module is followed by a downsampling operation and can obtain three feature maps with different scales. "Add" is the feature fusion operation. The blue block is the residual block, and the white block is the convolution. After applying the residual block and convolution, the located license plate image is obtained.

Figure 4: Photos of license plates in the CCPD2019 dataset.

Figure 5: License plate location results.

Figure 6: Results comparing the locations and the reserved license plates.

Table 1: Comparison results of four license plate location algorithms.

Model	Precision	Recall
Faster-RCNN	93.5%	88.7%
YOLOv3	93.2%	91.6%
SSD	94.6%	89.6%
Ours	98.7%	96.7%

The loss function used in license plate location is CIOU loss [27]. Compared with IOU loss [28], GIOU loss [29], and DIOU loss [27], it considers the overlapping area, center point distance, and aspect ratio. The model can accelerate convergence and improve the regression accuracy when performing bounding box selection.

$$IOU = \frac{|A \cap B|}{|A \cup B|}$$

$$Loss_{CIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

$$\alpha = \frac{v}{(1 - IOU) + v}$$

$$v = \frac{4}{\pi^2}\left[\left(\arctan\frac{\omega_1}{h_1} - \arctan\frac{\omega_1^{gt}}{h_1^{gt}}\right) + \left(\arctan\frac{\omega_2}{h_2} - \arctan\frac{\omega_2^{gt}}{h_2^{gt}}\right)\right] \tag{3}$$

where A and B are the two bounding boxes; IOU is the ratio of the intersection and union of these two boxes; b and b^{gt} denote the center points of the predicted and ground-truth boxes, respectively; ρrepresents the Euclidean distance between the bounding box and the center point of the ground truth; c represents the diagonal distance of the minimum circumscribed matrix of the predicted box and the ground-truth box; ω_i and h_i are the width and height of the bounding box, respectively; and ω_i^{gt} and h_i^{gt} are the width and height produced by the ground-truth value, respectively.

2.3. License Plate Recognition. The license plate recognition method requires character segmentation training, which increases the training time. Therefore, this paper uses the lightweight ShuffleNetv2 network as the backbone network, combined with dilated convolution and CTC Loss [30] to establish a model and adopts end-to-end training without character segmentation. In Figure 3, the white block is convolution, and the blue block is pooling. The yellow block represents the shuffle block with *stride* = 1. The part in the red box is the loop module. The black block represents the downsampling operation, and "Add" is the feature fusion operation. The orange block is the GC-DC block, which represents the global context block with dilated convolution.

The loss function used in the license plate recognition network is CTC loss. The traditional BP neural network is trained by the frame, and CTC loss is trained by the sequence. The training based on the frame needs to align samples. However, the location and proportion of license plates on each frame are different, which makes it difficult to align samples in practice. CTC loss is a loss function without the need to align samples, and only the corresponding sequence must be obtained for the same character.

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in L'^T, \tag{4}$$

$$p(l|x) = \sum_{\pi \in B^{-1}(1)} p(\pi|x), l \in L^{\leq T}, \tag{5}$$

$$p(l|x) = \sum_{\pi \in B^{-1}(l), \pi_t = l_k} \frac{\alpha_t(l_k)\beta_t(l_k)}{y_{l_k}^t}, \tag{6}$$

$$\alpha_t(l_k) = \sum_{\pi \in B(\pi_{1:t}) = l_{1:l_k}} \prod_{t'=1}^{T} y_{\pi_{t'}}^{t'}, \tag{7}$$

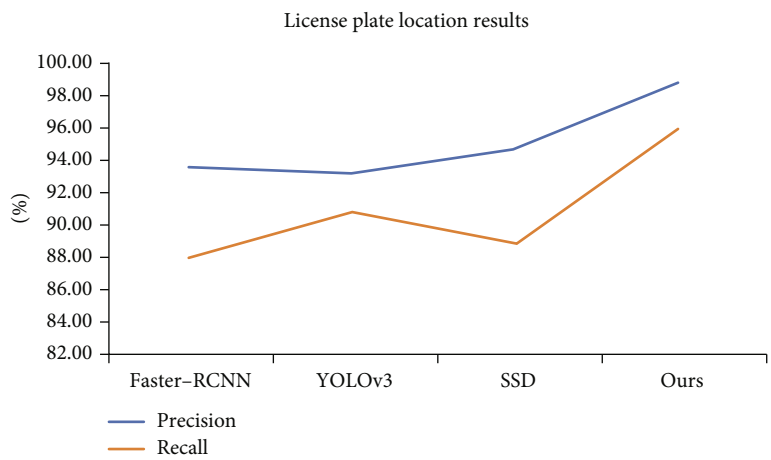


Figure 7: License plate location comparing results.

Figure 8: Results of license plate recognition.

Table 2: Comparison results of four license plate recognition algorithms.

Model	Precision	Recall
LPRNet	92.9%	90.5%
AlexNet	94.5%	93.6%
RPNet	95.3%	95.8%
Ours	98.2%	97.6%

$$\beta_t(l_k) = \sum_{\pi \in B(\pi_{t:T}) = l_{k:|l|}} \prod_{t'=t}^{T} y_{\pi_{t'}}^{t'}. \tag{8}$$

In Formula (4), x is the input data, and y is the output data. L is the sequence label, and L' is the sequence label with blank character. In Formula (5), B represents the set of many-to-one mappings. In Formula (6), $p(l|x)$ is the sum of probabilities that the input is x and the output is l, $\alpha_t(l_k)$ is the forward recursive probability sum, and $\beta_t(l_k)$ is the reverse recursive probability sum.

3. Results and Discussion

The dataset used in our experiment is the CCPD2019 dataset (https://github.com/detectRecog/CCPD). The CCPD2019 dataset is an open source license plate dataset of the University of Science and Technology of China, which is an authoritative dataset in the field of license plate recognition. In Figure 4, the photos of license plates in the CCPD2019 dataset involve a variety of complex environments. The CPU used in our experiment is an Intel(R) Core(TM) i9-9820X CPU @ 3.30 GHz, and the GPU is an NVDIA GeForce RTX 2080 Ti GPU.

Precision and recall are used as the evaluation criteria in our experiment [31], as shown in Formulas (9) and (10), where TP represents the number of license plates correctly determined by the model, FP represents the number of non-license plates incorrectly determined as license plates by the model, and FN represents the number of license plates incorrectly determined as non-license plates by the model.

$$Precision = \frac{TP}{TP + FP}, \tag{9}$$

$$Recall = \frac{TP}{TP + FN}. \tag{10}$$

3.1. License Plate Location Experiment. This experiment adopts the anchor-free location method based on four angles. In the training stage, it is necessary to locate the four-angle positions of license plates in each image and obtain the corresponding coordinates. We send the corresponding coordinate and the pictures into the model for training. The effect of license plate location is expressed by the Gaussian score. When the Gaussian score of training is higher than the set threshold, the model will start training the recognition part. The threshold of this experiment is set to 0.85. The license plate location results are shown in Figure 5.

The located image will not be directly input into the recognition network. There will be an interception operation to eliminate other parts and only retain the license plate. Figure 6 shows the results comparing the locations and the reserved license plates.

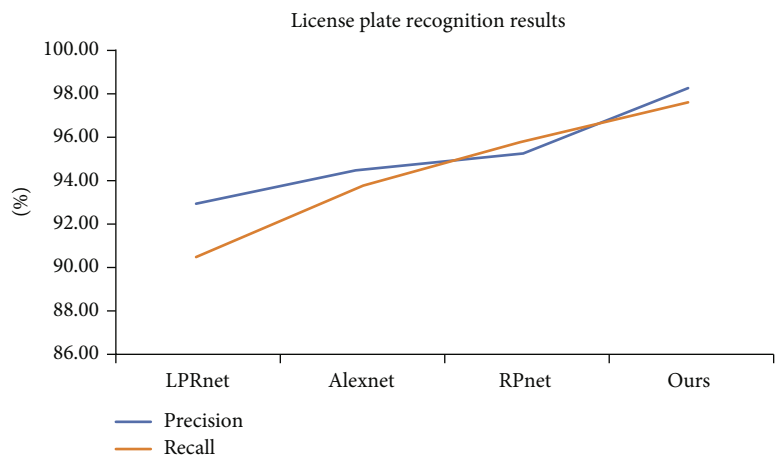


Figure 9: License plate recognition comparing results.

Our approach compares with three classical license plate location algorithms (fast RCNN, YOLOv3, and SSD). The comparison results on the CCPD2019 dataset are shown in Table 1.

As seen from the results in Table 1 and Figure 7, for the precision evaluation criterion, our approach is 5.2% more precise than Faster-RCNN, 5.5% more precise than YOLOv3, and 4.1% more precise than SSD. For the recall evaluation criterion, the recall of our approach is 8.0% higher than Faster-RCNN, 5.1% higher than YOLOv3, and 7.1% higher than SSD.

3.2. License Plate Recognition Experiment. The image obtained by the Section 3.1 is input to the license plate recognition network. The results of license plate recognition are shown in Figure 8.

Our approach is compared with three representative license plate recognition algorithms (LPRNet, AlexNet, and RPNet). The comparison results of the four license plate recognition algorithms are shown in Table 2.

As seen from the results in Table 2 and Figure 9, for the precision evaluation criterion, our approach is 5.3% more precise than LPRNet, 3.7% more precise than AlexNet, and 2.9% more precise than RPNet. For the recall evaluation criterion, the recall of our approach is 7.1% higher than LPRNet, 4.0% higher than AlexNet, and 1.8% higher than RPNet.

4. Conclusions

The combination of the big data and Internet of Things can achieve good results by training the model with data obtained from urban intelligent cameras [32]. Big data helps to establish a comprehensive traffic information system. By integrating the "data warehouse" in different regions and fields, the integrated utilization mode of public transport information is constructed. The Internet of Things and big data technology are research focus and would have a profound impact on the intelligent transportation.

Therefore, this paper proposes an SDC model, which combines dilated convolution and global context blocks. Therefore, the receptive field and feature expression ability of the model are enhanced. For license plate location, CIOU loss considers not only the coverage area of the bounding box but also the center distance and aspect ratio. For license plate recognition, CTC loss trains the network based on the sequence and solves the sample alignment problem, which improves the accuracy of license plate recognition.

As a part of intelligent transportation, license plate recognition technology based on deep learning has laid a good foundation for the future analysis and decision-making of the Internet of Things [33]. However, the accuracy for complex situations is not high. For example, for snowy and rainy days, license plates re prone to partial occlusion, which will lead to license plate recognition errors. For dim or nighttime environments, due to the lack of light, there is a lack of color information in the picture, making it difficult to locate and recognize license plates. Therefore, adjusting the parameters of the deep learning model in complex situations will become a difficult problem in the Internet of Things.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by horizontal scientific research project of Campus Network Design Scheme (HX2021251), in part by the Hubei Natural Science Foundation under grant 2021CFB156 and the JSPS KAKENHI under grant JP21K17737.

References

[1] L. Tan, K. Yu, L. Lin et al., "Speech emotion recognition enhanced traffic efficiency solution for autonomous vehicles in a 5G-enabled space-air-ground integrated intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2830–2842, 2022.

[2] K. Yu, L. Tan, C. Yang et al., "A blockchain-based Shamir's threshold cryptography scheme for data protection in industrial Internet of Things settings," *IEEE Internet of Things Journal*, 2021.

[3] L. Zhen, Y. Zhang, K. Yu, N. Kumar, A. Barnawi, and Y. Xie, "Early collision detection for massive random access in satellite-based internet of things," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5184–5189, 2021.

[4] F. Ding, K. Yu, Z. Gu, X. Li, and Y. Shi, "Perceptual enhancement for autonomous vehicles: restoring visually degraded images for context prediction via adversarial training," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[5] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13784–13795, 2020.

[6] X. Shang, L. Tan, K. Yu, J. Zhang, K. Kaur, and M. M. Hassan, "Newton-interpolation-based zk-SNARK for artificial Internet of Things," *Ad Hoc Networks*, vol. 123, article 102656, 2021.

[7] L. Tan, N. Shi, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered access control framework for smart devices in green Internet of Things," *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1–20, 2021.

[8] H. Lin, J. Zhao, S. Li, and G. Qiu, "License plate location method based on edge detection and mathematical morphology," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, 2020.

[9] M. Zhang, W. Yu, J. Su, and W. Li, "Design of license plate recognition system based on machine learning," in *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, pp. 518–522, Xiamen, China, 2019.

[10] G. Lin, B. Xue, B. Xu, and C. Chen, "License plate recognition based on mathematical morphology and template matching," in *2019 Chinese Automation Congress (CAC)*, pp. 405–410, Hangzhou, China, 2019.

[11] L. Du, L. Li, D. Wei, and J. Mao, "Saliency-guided single shot multibox detector for target detection in SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3366–3376, 2020.

[12] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," *Computer Vision and Pattern Recognition*, pp. 1–6, 2018.

[13] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "TOOD: task-aligned one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, QC, Canada, 2021.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[15] Y. Liu, J. Han, Q. Zhang, and L. Wang, "Salient object detection via two-stage graphs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 1023–1037, 2019.

[16] I. Kusumadewi, C. A. Sari, D. Setiadi, and E. H. Rachmawanto, "License number plate recognition using template matching and bounding box method," *Journal of Physics Conference Series*, vol. 1201, no. 1, article 012067, 2019.

[17] K. Abebe, R. Sharma, Y. K. Chung, and D. Zerihum, "Vehicle plate recognition for Ethiopian license plate: based sift feature," *International Journal of Engineering and Technology*, vol. 8, no. 6, pp. 568–570, 2020.

[18] S. Zherzdev and A. Gruzdev, "LPRNet: license plate recognition via deep neural networks," *Computer Vision and Pattern Recognition*, pp. 1–6, 2018.

[19] C. G. Xiu, X. Dan, and S. Zhang, "Research on mining platform scale license plate based on deep learning," *IOP Conference Series: Earth and Environmental Science*, vol. 242, no. 2, pp. 22048–22053, 2019.

[20] Z. Xu, W. Yang, A. Meng et al., "Towards end-to-end license plate detection and recognition: a large dataset and baseline," in *Computer Vision – ECCV 2018. ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11217 of Lecture Notes in Computer Science, pp. 255–271, Springer, Cham, 2018.

[21] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "Shufflenet v2: practical guidelines for efficient cnn architecture design," in *Computer Vision – ECCV 2018. ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11218 of Lecture Notes in Computer Science, pp. 116–131, Springer, Cham, 2018.

[22] D. V. Niture, V. Dhakane, P. Jawalkar, and A. Bamnote, "Smart transportation system using IOT," *International Journal of Engineering and Advanced Technology*, vol. 10, no. 5, pp. 434–438, 2021.

[23] X. Chen, "Research and application of license plate recognition system for property access control," *Modern Information Technology*, no. 2, pp. 1–3, 2018.

[24] R. Gomes, P. Rozario, and N. Adhikari, "Deep learning optimization in remote sensing image segmentation using dilated convolutions and ShuffleNet," in *2021 IEEE International Conference on Electro Information Technology(EIT)*, pp. 244–249, Mt. Pleasant, MI, USA, 2021.

[25] Z. Shen, L. Zhuang, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "DSOD: learning deeply supervised object detectors from scratch," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1919–1927, Venice, Italy, 2017.

[26] Y. Yang and H. Deng, "GC-YOLOv3: you only look once with global context block," *Electronics*, vol. 9, no. 8, p. 1235, 2020.

[27] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: faster and better learning for bounding box regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12993–13000, 2020.

[28] S. L. Tychsen and L. Petersson, "Improving object localization with fitness nms and bounded iou loss," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6877–6885, Salt Lake City, UT, USA, 2018.

[29] Y. Chen, H. Li, R. Gao, and D. Zhao, "Boost 3-D object detection via point clouds segmentation and fused 3-D GIoU-L_1 loss," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 762–773, 2020.

[30] X. Kang, H. Huang, Y. Hu, and Z. Huang, "Connectionist temporal classification loss for vector quantized variational

autoencoder in zero-shot voice conversion," *Digital Signal Processing*, vol. 116, article 103110, 2021.

[31] N. S. Hadjidimitriou, M. Lippi, M. Dell'Amico, and A. Skiera, "Machine learning for severity classification of accidents involving powered two wheelers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4308–4317, 2019.

[32] J. Huang, L. Tan, W. Li, and K. Yu, "RON-enhanced blockchain propagation mechanism for edge-enabled smart cities," *Journal of Information Security and Applications*, vol. 61, article 102936, 2021.

[33] K. Yu, L. Tan, S. Mumtaz et al., "Securing critical infrastructures: deep-learning-based threat detection in IIoT," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 76–82, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1238571, 8 pages
https://doi.org/10.1155/2022/1238571

Research Article

Count-Based Exploration via Embedded State Space for Deep Reinforcement Learning

Xinyue Liu,[1] **Qinghua Li**,[1] **and Yuangang Li**[2]

[1]*School of Software, Dalian University of Technology, Dalian 116620, China*
[2]*Faculty of Business Information, Shanghai Business School, Shanghai 200235, China*

Correspondence should be addressed to Yuangang Li; liyuangang76@163.com

Received 15 December 2021; Revised 17 March 2022; Accepted 19 April 2022; Published 17 May 2022

Academic Editor: Changqing Luo

Count-based exploration algorithms have shown to be effective in dealing with various deep reinforcement learning tasks. However, existing count-based exploration algorithms cannot work well in high-dimensional state space due to the complexity of state representation. In this paper, we propose a novel count-based exploration method, which can explore high-dimensional continuous state space and combine with any reinforcement learning algorithms. Specifically, by introducing the embedding network to encode the state space and to merge the states with similar key characteristics, we can compress the high-dimensional state space. By utilizing the state binary code to count the occurrence number of states, we generate additional rewards which can encourage the agent to explore the environment. Extensive experimental results on several commonly used environments show that our proposed method outperforms other strong baselines significantly.

1. Introduction

Reinforcement learning (RL), which was aimed at learning an optimal control strategy to maximize the reward from the environment, has achieved great success in various complex tasks, such as video games [1] and robot controlling [2]. One of the core problems of RL methods is how the agent should take trade-off decisions between the exploration of new actions and the selection of the best action based on existing knowledge. Although there are simple and theoretically guaranteed heuristic exploration methods for tabular RL algorithms, such as the ε-greedy strategy [3] and entropy regularization [4], these methods cannot be easily extended to a high-dimensional space environment due to the large state space and the complexity of state representation. Therefore, developing a common and simple exploration method is an important research direction.

In this paper, we propose a novel count-based exploration method via embedded state space for reinforcement learning. The core idea is to compress the high-dimensional state space by extracting the embedded representation of the state space and merging similar states. We use an action prediction model to train the state embedding network for obtaining a better state feature space. Take our human agent as an example. Assume someone is playing a racing game where the screen changes as the car goes ahead, as shown in Figure 1. The track changing in the yellow box will affect the next move, while the sky changing in the red box will not. We should focus on the state characteristics that affect the choice of actions. In summary, the contributions of this paper are as follows: (1) we propose a new count-based exploration method which is suitable for high-dimensional state space. And our method can be directly applied to most different RL algorithms. (2) We design a general mechanism for optimizing feature representations by introducing the embedding network and the action prediction model. (3) We conduct experiments on several games from the Atari [1] and achieve near state-of-the-art results, especially with fewer training epochs.

The rest of this article is organized as follows: in Section 2, we review the related work. In Section 3, we introduce some definitions and propose our novel exploration method. In Section 4, we show the experimental results on different kinds of environments. Finally in Section 5, we conclude the paper and point out the future work.

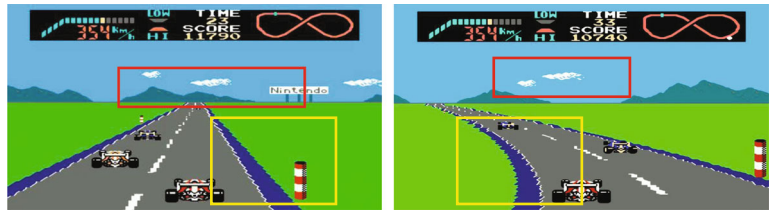


Figure 1: F1-race game screen. Red box line: state features that do not affect the choice of actions. Yellow box line: state features that affect the choice of actions.

2. Related Work

Many different approaches have been proposed in recent years to address the balance between exploration and exploitation. These methods can be divided into two types: count-based exploration methods and curiosity-based exploration methods. The former methods count the occurrence number of states and convert this number into a reward to encourage exploring states with higher rewards. One of the best-known approaches is the UCB bandit algorithm [5], which selects an action a_t at time t to maximize the upper confidence bound $\hat{r}(a_t) + \sqrt{2 \log t / n(a_t)}$, where $\hat{r}(a_t)$ is the estimated reward and $n(a_t)$ is the occurrence number of action a_t being previously chosen. Model-Based Interval Estimation-Exploration Bonus (MBIE-EB) of [6] has similar structure. It counts state-action pairs $n(s, a)$ with a table and adds a bonus reward of the form $\beta / \sqrt{n(s, a)}$ to encourage exploring less-visited pairs. It is proved by [7] that square root inverse correlation is optimal. Tang [8] uses hash functions to encode the state space, subsumes similar states into a single counter, and explores based on the counter's value. Martin et al. [9] generalize a probability density model by the characteristic representation of the state space and use this model to pseudocount. Curiosity-based exploration methods offer additional rewards based on the principle of optimism in the face of uncertainty [10]. These methods encourage the agent to choose actions that increase uncertainty about the value estimate. Classical examples utilize upper confidence bound [11] and Thompson sampling [12] for the stochastic sampling of actions. Recent algorithms combine these ideas with finer uncertainty, making them suitable for large state spaces that require deep exploration [13–15]. Dynamic autoencoder (Dynamic-AE) [16] is proposed to compress the state space. The distance between predicted state and real state is computed in this compressed state space. And intrinsic rewards are defined by this distance. Pathak et al. [17] use a self-supervised inverse dynamics model to predict the next state based on the current state-action pair. Then, they use the error between prediction and reality to generate curiosity. Savinov et al. [18] propose a new curiosity definition that marks the novelty of states by reachability. The episodic curiosity module (ECO) uses an episodic memory pool to store part of visited states. To compute the state novelty, ECO compares each state with states in memory. If the current state is far from the states contained in memory, the agent is rewarded an intrinsic reward.

Several recent studies have discussed the generalization of reinforcement learning and designed procedurally generated environments to test the generalization of reinforcement learning [19–21]. More recent papers show that traditional exploration methods fall short in procedurally generated environments and address this issue with new exploration methods [22, 23]. [24] proposes a new perspective of exploration bonus in episode-level data and achieves significantly SOTA performance on procedurally generated benchmarks. In the field of multiagent reinforcement learning (MARL), the study on exploration is roughly at the preliminary stage. Most of these exploration methods extend the ideas in the single-agent setting and propose different mechanisms by integrating the characteristics of deep MARL. Compared to the RL exploration, the dimensions of the state-action space increase rapidly as the number of agents increases in MARL. Zhou et al. [25] propose to treat the Q-function as a high-order high-dimensional tensor. Then, they approximate the Q-function with factorized pairwise interactions. [26] adopts a similar factorization approach in the search space to solve this problem.

3. The Proposed Method

In this work, we propose a novel count-based exploration method via embedded state space for deep reinforcement learning. Our method can be divided into two major parts: (1) Embedding Network and Action Prediction Model and (2) Count-Based Extra Bonus Generator. For different RL tasks, we first collect state information by agents randomly interacting with the environment. These data will be used to train the embedding network that can represent state features better. Then, we count the occurrence number of states based on the embedded feature representation. Finally, we generate extra bonus and add it to RL algorithms for training the agent.

3.1. Notations. Reinforcement learning (RL) [3] addresses the task of learning from interactions to achieve goals. It is usually formulated as an MDP $\langle S, A, P, R, \gamma \rangle$, where S is the set of states of the environment, A is the set of available actions, $P : (S \times A) \times S \longrightarrow [0, 1]$ is the state transition distribution, $R : (S \times A) \times S \longrightarrow R$ is the reward function, and γ is the discount factor. The agent is formally a policy $\pi : S \longrightarrow A$ that maps a state to an action. At timestep t, the agent is in a state $s_t \in S$, receives a reward r_t, and takes an action

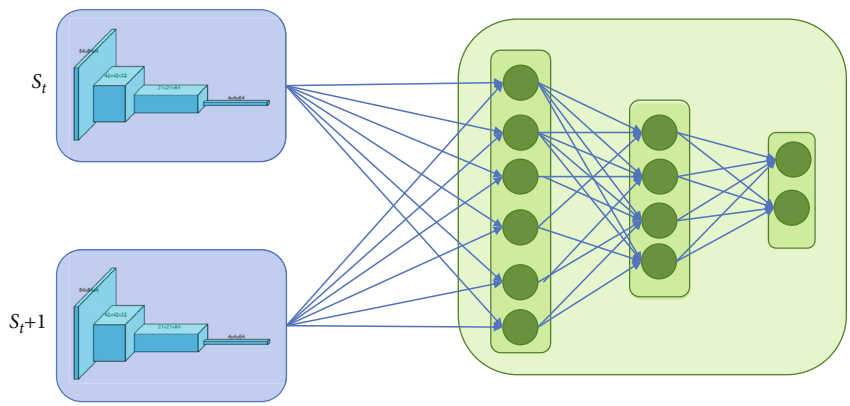


Figure 2: Left submodule encodes s_t into a feature $\phi(s_t)$. Right submodule predicts the action a_t.

$a_t \in A$. We seek a policy π that maximizes the expected sum of future rewards. The action-value $Q^\pi(s, a)$ of a state-action pair (s, a) under a policy π is the expected discounted sum of future rewards and follows π thereafter: $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a]$.

3.2. Embedding Network and Action Prediction Model. When MDP states have complex structures, as in the case of image observations, directly measuring their similarities in pixel space does not provide effective metric. Previous works in computer vision [27–29] introduce manually designed feature representations of images. These representations are suitable for semantic tasks including detection and classification. More recent methods learn complex features directly from data by training convolutional neural networks [30–32]. Considering these researches, it may be difficult to combine similar states using raw pixels or the general state space.

As mentioned in the previous F1 racing game example, some features in the state space are invalid for the agent, so we need to extract the valid features in the state space. To achieve this, we first divide the features of the state into three parts: (1) something that can be controlled by the agent (e.g., the car in the game), (2) things that the agent cannot control but that can affect the agent (e.g., the track), and (3) things out of the agent's control and not affecting the agent (e.g., the sky). We need to find a good feature space that includes the features of (1) and (2) and excludes the features of (3).

Our goal is to come up with a general mechanism for learning feature representations rather than manually designing for each environment. We propose that such a feature space can be learned by training a deep neural network with two submodules: the first submodule (embedding network) encodes the raw state s_t into a feature vector $\varphi(s_t)$. The second submodule (prediction network) takes the feature encoding $\varphi(s_t)$, $\varphi(s_{t+1})$ of two consequent states as inputs and predicts the action a_t taken by the agent to move from s_t to s_{t+1}. The whole model is illustrated in Figure 2. Training this neural network is equivalent to learning function f defined as

$$\widehat{a}_t = f(s_t, s_{t+1}; \theta_e), \tag{1}$$

where $\widehat{a}_t$ is predicted estimate of the action a_t and the neural network parameters θ_e are trained to optimize

$$\min_{\theta_e} L(\widehat{a}_t, a_t), \tag{2}$$

where L is the loss function that measures the discrepancy between the predicted and actual actions. In order to facilitate the subsequent processing of the state encoding, the embedding network takes the state s as the input and contains one special dense layer comprised of D sigmoid functions. By rounding the sigmoid activation $b(s)$ of this layer to their closest binary number $\lfloor b(s) \rceil \in \{0, 1\}^D$, any state s can be binarized. A problem with this architecture is that if $b(s_t)$ near 0.5 at a particular dimension, the error will increase while rounding. A solution is forcing the binary code layer to take on binary values. Therefore, we add another loss term into L, and the complete loss function is defined as

$$L(s_t, a_t, s_{t+1}) = \left[NLLLoss(\widehat{a}_t, a_t) + \lambda \sum_{i=1}^{D} \begin{pmatrix} \min\{(1 - b_i(s_t))^2, b_i(s_t)^2\} + \\ \min\{(1 - b_i(s_{t+1}))^2, b_i(s_{t+1})^2\} \end{pmatrix} \right]. \tag{3}$$

The tuple (s_t, a_t, s_{t+1}) is obtained while the agent interacts with the environment using its current policy $\pi(s)$.

3.3. Count-Based Extra Bonus Generator. We get embedded state space $\phi(s)$ through the embedding network. An exploration bonus $r^c : S \longrightarrow \mathbb{R}$ is added to the reward function, defined as

$$r^c(s) = \frac{\beta}{\sqrt{n(\phi(s))}}, \tag{4}$$

where $\beta \in \mathbb{R}_{\geq 0}$ is the bonus coefficient. Initially, the counts $n(\bullet)$ are set to 0 for the whole range of ϕ. For every state s_t encountered at time step t, $n(\phi(s_t))$ is increased by 1. The agent is trained with rewards $(r + r^c)$, while per performance is evaluated as the sum of rewards without bonuses.

We represent the policy $\pi(s_t; \theta_p)$ by a deep neural network with parameters θ_p. Given the agent in state s_t, it

Initialize $A \in R^{k \times D}$ with entries drawn i.i.d. from the standard Gaussian distribution $\mathrm{N}(0, 1)$;
Initialize a hash table with values $n(\cdot) \equiv 0$;
Initialize policy network with parameter θ_p and embedding network with parameter θ_c;
for each iteration j **do** {
  Collect a set of state-action samples $(s_t, a_t, s_{t+1})_{m=0}^{M}$ with policy π;
  Add the state samples to replay buffer;
  if $j \bmod j_{update} = 0$ **then** {
    Update the embedding network with loss function in Eq.(3). using samples drawn from the replay buffer;
  }
  Compute $\phi(s_t) = \lfloor b(s_t) \rceil$, the D-dim rounded hash code for s_t learned by the embedding network;
  Update the hash table counts $\forall m : 0 \le m \le M$ as $n(\phi'(s_m)) \longleftarrow n(\phi'(s_m)) + 1$;
  Update the policy π using rewards $\{r(s_m, a_m) + (\beta/\sqrt{n(\phi'(s_m))})\}_{m=0}^{M}$ with any RL algorithm;
}

Algorithm 1: Count-based exploration via embedded state space.

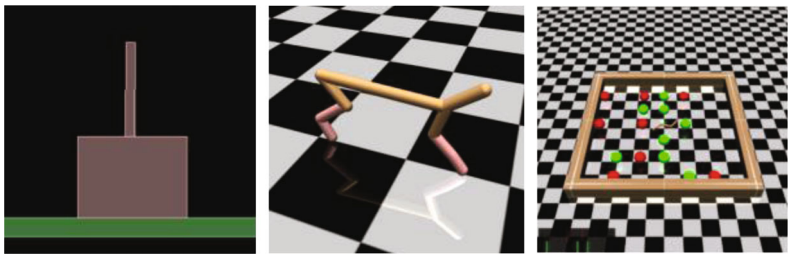


Figure 3: Illustrations of the Rllab task used in continuous control experiments, namely, CartPoleSwingup, HalfCheetah, and SwimmerGather.

executes the action $a_t \sim \pi(s_t; \theta_P)$ sampled from the policy. θ_P is optimized to maximize the expected sum of rewards:

$$\max_{\theta_p} \mathbb{E}_{\pi(s_t;\theta_p)} \left[\sum_t (r_t + r_t^c) \right]. \tag{5}$$

Because the code dimension often needs to be large for correctly predicting the action, we apply a downsampling procedure to the resulting binary code $\lfloor b(s) \rceil$, which can be done through random projection to a lower-dimensional space via SimHash:

$$\phi'(s) = \mathrm{sgn}\,(A\phi(s)) \in \{-1, 1\}^k, \tag{6}$$

where A is a $k \times D$ matrix drawn from a standard Gaussian distribution $\mathcal{N}(0, 1)$. The value for k controls the granularity: higher values lead to fewer collisions and clearly distinguish states. Algorithm 1 summarizes our method.

4. Experiments

4.1. Experimental Setup. We test our method in multiple environments from Rllab to Arcade Learning Environment (ALE) [19]. The experiments in Rllab verify that our method can be used in continuous control tasks. The experiments in ALE have recently become a standard high-dimensional benchmark for RL. The reward signal is computed from the game score. The raw state is a frame of video (a 160×210 array of 7-bit pixels). There are 18 available actions. The ALE is a particularly interesting testbed in our context, because the difficulty of exploration varies greatly among games. We choose six of these games where exploration is hard. Trust Region Policy Optimization is chosen as the RL algorithm for all experiments, because this algorithm can handle both discrete and continuous action spaces and is relatively insensitive to hyperparameter changes. The hyperparameter is $\beta = 0.01$ and $\lambda = 0.1$. All image curves are smoothed.

4.2. Rllab Environment. The Rllab benchmark consists of various control tasks to test deep RL algorithms. We selected several variants of the basic and locomotion tasks that use sparse rewards, as shown in Figure 3. These tasks are all highly difficult to solve with naive exploration strategies, such as adding Gaussian noise to the actions.

Figure 4 shows the results of TRPO (baseline), TRPO-SimHash [8], VIME [34], and our method on the classic tasks CartPoleSwingup, the locomotion task HalfCheetah, and the hierarchical task SwimmerGather. Using count-based exploration with embedded state space is capable of reaching the goal in all environments (which corresponds to a nonzero return), while baseline TRPO with Gaussian n control noise fails completely. Although our method picks up the sparse reward on HalfCheetah and receives better reward than other count-based exploration algorithm, it does not perform as well as VIME. In contrast, the

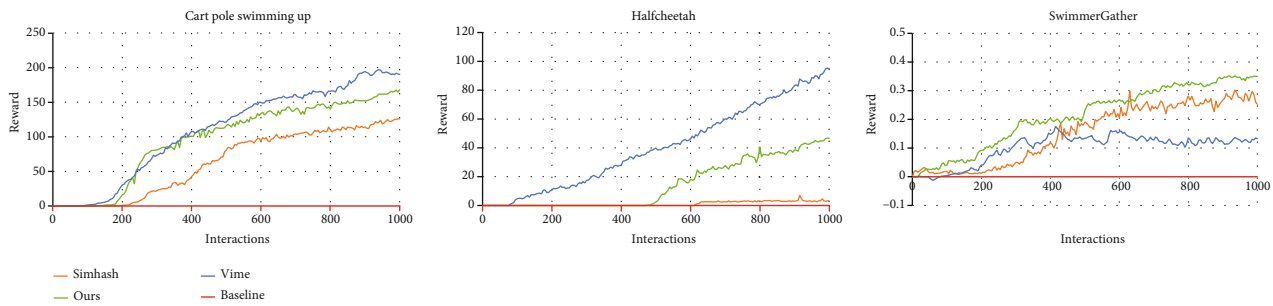


Figure 4: Mean average return of different algorithms on Rllab tasks with sparse rewards.

Table 1: Atari 2600: average total reward after training for 50 M time steps. Boldface numbers indicate best results. Italic numbers are the best among count-based exploration methods.

	Freeway	Frostbite	Gravitar	Montezuma	Solaris	Venture
TRPO (baseline)	16.5	2869	486	0	2758	121
Double-DQN	33.3	1683	412	0	3068	98
Dueling network	0	4672	588	0	2251	497
Gorila	11.7	605	**1054**	4	N/A	**1245**
DQN Pop-Art	33.4	3469	483	0	4544	1172
A3C+	27.3	507	246	142	2175	0
TRPO+AE	33.5	5214	482	75	4467	445
TRPO+BASS	28.4	3150	604	***238***	1201	616
TRPO+OURS	***34***	***5537***	*712*	196	***4860***	*983*

performance of ours is comparable with VIME on CartPoleSwingup, while it outperforms VIME on SwimmerGather.

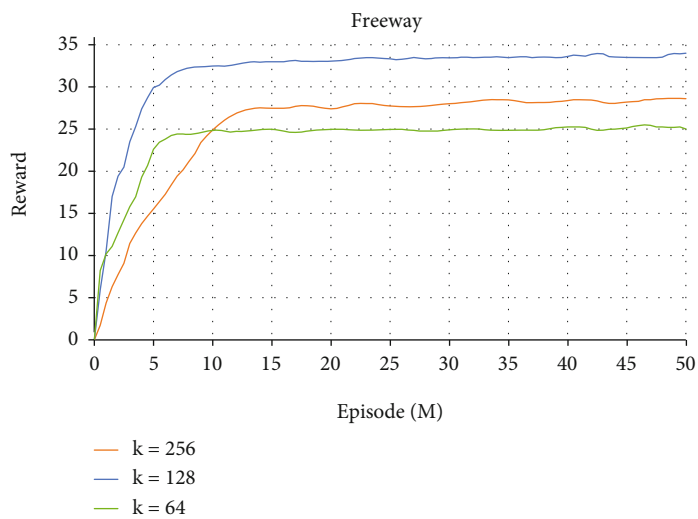
4.3. Arcade Learning Environment. The Arcade Learning Environment (ALE) [33], which consists of Atari 2600 video games, is an important benchmark for deep RL due to its high-dimensional state space and wide variety of games. In order to demonstrate the effectiveness of the proposed exploration strategy, six games are selected featuring long horizons while requiring significant exploration: Freeway, Frostbite, Gravitar, Montezuma's Revenge, Solaris, and Venture. The agent is trained for 500 iterations in all experiments, with each iteration consisting of 0.1 M steps (the TRPO batch size corresponds to 0.4 M frames).

We compare our results to double DQN [35], dueling network [36], A3C+ [37], double DQN with pseudocounts [37], Gorila [38], DQN Pop-Art [39], and TRPO-SimHash [8] the "null op" metric. We summarize all results in Table 1.

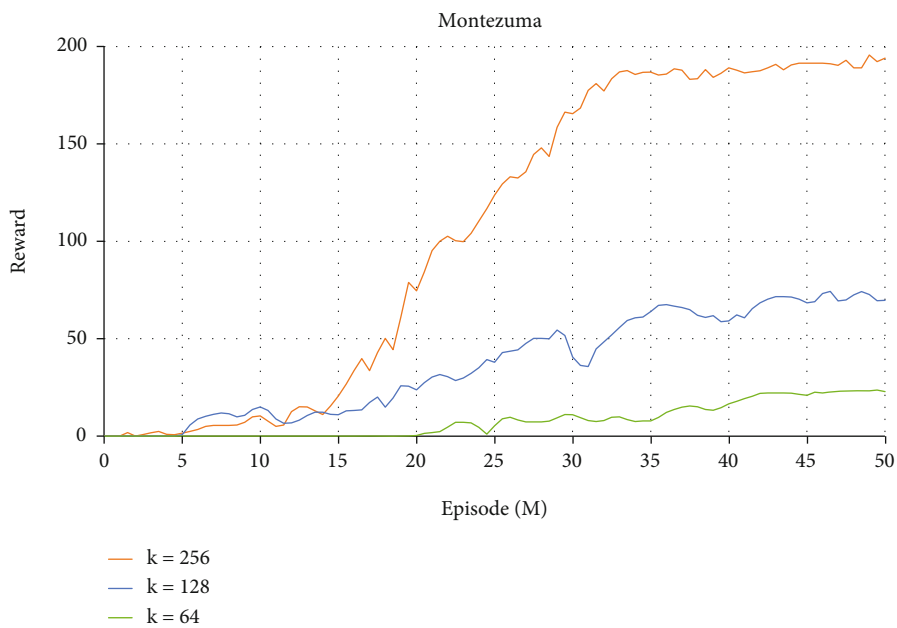
As observed in Table 1, our approach has performed better on most of the games compared to similar count-based exploration methods. It means the embedded state space after feature extraction by using action predict network can select the part which has more important influence on the decision-making of the agent. Our method achieves near state-of-the-art performance on Freeway, Frostbite, and Solaris. A reason why TRPO+BASS is better than ours on Montezuma is that BASS is a hand-designed feature transformation for images in Atari 2600 games. The hash codes generated by our method distinguish between visually different states but fail to emphasize that the agent needs to explore different rooms. But the hand-designed feature transformation can clearly describe this key information.

4.4. Downsampling. We apply a downsampling process to the generated binary code in equation (6), which can be done using SimHash's random projection to a lower-dimensional space. However, there may be states that are distinct but fall into the same group after downsampling. Moreover, different downsampling dimensions have different effects on the final experimental results. We conduct more experiments in different game environments. Figure 5 and Table 2 show an overview of the results.

As observed in the results, in the case of low-latitude downsampling, the agent reaches the plateau fastest, but the reward is relatively lower. Conversely, in the case of high-latitude downsampling, the speed of convergence is slower, but the reward obtained is higher. This phenomenon is well explained because downsampling at low latitudes greatly compresses the state space, and the novelty to the agent disappears quickly. And because many different states are assigned to the same state, the agent's exploration ability is greatly affected, and more effective rewards cannot be obtained. And we found that there is a certain correlation between the complexity of the environment and the optimal dimension of downsampling. For complex games such as Montezuma, a higher-dimensional downsampling representation can better distinguish different states and achieve excellent returns. For games with relatively low complexity, such as Freeway, high-dimensional downsampling will cause overfitting. Overdistinguishing similar states will lead to lower overall rewards for the agent.



(a)



(b)

Figure 5: The influence of different downsampling dimensions on experimental results.

Table 2: Downsampling experiments in Atari 2600. K represents the dimension of downsampling. Boldface numbers indicate best results.

	$K = 64$	$K = 128$	$K = 256$
Freeway	25.1	**34**	28.3
Venture	479	677	**983**
Montezuma	24	74	**196**
Frostbite	3043	**5537**	4983

5. Conclusion

In this paper, we propose a novel count-based exploration method for deep reinforcement learning. By introducing the embedding network and the action prediction model, the proposed method tends to extract the state features that have positive impacts on the agent and encourage the agent to explore states with higher rewards. Extensive experiments demonstrate that our proposed method can achieve promising performance on different tasks. In future work, we plan to optimize the representation of state features and attempt to apply the state feature extraction framework to other kinds of reinforcement learning exploration methods.

Data Availability

Previously reported environment data were used to support this study and are available at 10.1613/jair.3912. These prior studies (and datasets) are cited at relevant places within the text as references. The experiment data used to support the

findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. 61972065).

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, 2015.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction[M]*, MIT press, 2018.

[4] V. Mnih, A. P. Badia, M. Mirza et al., "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, pp. 1928–1937, 2016.

[5] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[6] A. L. Strehl and M. L. Littman, "An analysis of model-based interval estimation for Markov decision processes," *Journal of Computer and System Sciences*, vol. 74, no. 8, pp. 1309–1331, 2008.

[7] J. Z. Kolter and A. Y. Ng, "Near-Bayesian exploration in polynomial time," in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pp. 513–520, 2009.

[8] H. Tang, *Towards Informed Exploration for Deep Reinforcement Learning*, University of California, Berkeley, 2019.

[9] J. Martin, S. N. Sasikumar, T. Everitt, and M. Hutter, "Count-based exploration in feature space for reinforcement learning," 2017, http://arxiv.org/abs/1706.08090.

[10] J. Schmidhuber, "A possibility for implementing curiosity and boredom in model-building neural controllers," in *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, 1991.

[11] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2/3, pp. 235–256, 2002.

[12] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3-4, pp. 285–294, 1933.

[13] I. Osband, C. Blundell, A. Pritzel, and B. V. Roy, "Deep exploration via bootstrapped DQN," *Advances in neural information processing systems*, vol. 29, 2016.

[14] B. O'Donoghue, I. Osband, R. Munos, and V. Mnih, "The uncertainty bellman equation and exploration," in *International Conference on Machine Learning*, pp. 3836–3845, 2018.

[15] M. Fortunato, M. G. Azar, B. Piot et al., "Noisy networks for exploration," in *ICLR*, 2018.

[16] B. C. Stadie, S. Levine, and P. Abbeel, "Incentivizing exploration in reinforcement learning with deep predictive models," 2015, http://arxiv.org/abs/1507.00814.

[17] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16-17, 2017.

[18] N. Savinov, A. Raichuk, D. Vincent, M. Pollefeys, T. Lillicrap, and S. Gelly, "Episodic curiosity through reachability," in *International Conference on Learning Representations*, 2018.

[19] A. Rajeswaran, K. Lowrey, E. V. Todorov, and S. M. Kakade, "Towards generalization and simplicity in continuous control," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[20] A. Zhang, N. Ballas, and J. Pineau, "A dissection of overfitting and generalization in continuous reinforcement learning," 2018, http://arxiv.org/abs/1806.07937.

[21] J. Choi, Y. Guo, M. Moczulski et al., "Contingency-aware exploration in reinforcement learning," in *International Conference on Learning Representations*, 2018.

[22] R. Raileanu and T. Rocktaschel, "Ride: rewarding impact-driven exploration for procedurally-generated environments," in *International Conference on Learning Representations*, 2020.

[23] A. Campero, R. Raileanu, H. Küttler, J. B. Tenenbaum, T. Rocktäschel, and E. Grefenstette, "Learning with amigo: Adversarially motivated intrinsic goals," 2020, http://arxiv.org/abs/2006.12122.

[24] D. Zha, W. Ma, L. Yuan, X. Hu, and J. Liu, "Rank the episodes: a simple approach for exploration in procedurally-generated environments," 2021, http://arxiv.org/abs/2101.08152.

[25] M. Zhou, Y. Chen, Y. Wen et al., "Factorized q-learning for large-scale multi-agent systems," in *Proceedings of the First International Conference on Distributed Artificial Intelligence*, Beijing China, 2019.

[26] B. H. Abed-Alguni, S. K. Chalup, F. A. Henskens, and D. J. Paul, "A multi-agent cooperative reinforcement learning model using a hierarchy of consultants, tutors and workers," *Vietnam Journal of Computer Science*, vol. 2, no. 4, pp. 213–226, 2015.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, 2005.

[28] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, IEEE, 1999.

[29] E. Tola, V. Lepetit, and P. Fua, "Daisy: an efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, http://arxiv.org/abs/1409.1556.

[33] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: an evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.

[34] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "Vime: variational information maximizing exploration," 2016, http://arxiv.org/abs/1605.09674.

[35] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, Phoenix, Arizona USA, 2016.

[36] Z. Wang, T. Schaul, M. Hessel, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*, pp. 1995–2003, PMLR, 2016.

[37] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos, "Unifying count-based exploration and intrinsic motivation," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[38] A. Nair, P. Srinivasan, S. Blackwell et al., "Massively parallel methods for deep reinforcement learning," 2015, http://arxiv.org/abs/1507.04296.

[39] H. P. van Hasselt, A. Guez, M. Hessel, V. Mnih, and D. Silver, "Learning values across many orders of magnitude," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4147373, 14 pages
https://doi.org/10.1155/2022/4147373

Research Article

The Optimal Packets Scheduling for Buffer-Aid Energy Harvesting RSUs in Cooperative Vehicle Infrastructure System

Ying Shi,[1,2] Chi Feng,[1] Tong Wang,[1] Qingjiang Yang,[2] and Yiping Chen[2]

[1]*College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China*
[2]*School of Electric and Information Engineering, Heilongjiang University of Science and Technology, Harbin 150022, China*

Correspondence should be addressed to Yiping Chen; sy_hit2004@163.com

Received 10 February 2022; Accepted 1 April 2022; Published 26 April 2022

Academic Editor: Han Liu

In this paper, we investigate the optimal packet scheduling for RSU-to-vehicle downlink communication in Cooperative Vehicle Infrastructure System (CVIS). The RSU is powered by a capacity-limited battery storing harvested energy with reliable energy source as backup, which meets requirement of renewable energy and ensures sustainability of communication system. We aim to find the optimal packet scheduling policy that minimizes the data queuing delay under a given the reliable power constraint and downlink availability rate. According to probabilistic scheduling policy, we adopt data buffer queue and harvesting energy queue as the state-space, and determine the transition action by the joint state of packet arrivals and channel state to establish a two-dimensional Markov chain. Based on this, the optimization problem is formulated. By solving the problem and analysing its solution, we derive the optimal transmission parameters and the optimal scheduling policy which turns out to be threshold-based. Simulations are demonstrated to verify the accuracy of the theoretical derivation results.

1. Introduction

1.1. Motivations. Internet of Vehicles (IoV) is the application of Internet of Things (IoT) in urban traffic, which promotes the development of urban intelligent transportation system (ITS). At present, the Cooperative Vehicle Infrastructure System (CVIS) has become one of the most promising and fastest growing research in ITS. In CVIS, accurate and timely information interaction between vehicles and infrastructure can contribute to solve intelligent information management and decision-making problems, such as traffic congestion and road safety problems [1, 2]. Data transmission is completed in the way of "store-carry-forward" through passing vehicles within roadside units (RSUs) coverage. Therefore, the RSUs are not only important network access equipment with providing high-speed message forwarding services for vehicles over high bandwidth links, but also gateway nodes of the surrounding environmental monitoring sensor network, which undertakes the function of transmitting monitoring data to roadside units connected with the Internet [3, 4]. Nevertheless, the high-speed and high bandwidth forwarding services require high power consumption [5]. In addition, the RSUs are powered by the electrical grid instead of no-renewable energy, which costs too much, and increased usage of such energy resources will lead to increase carbon dioxide emission. Thereby, renewable and harvested energy resources, such as solar energy, wind energy and RF energy, are generally considered as substitute for conventional grid power [6].

However, harvested energy arrives randomly and sporadically due to environmental influence, such as weather and geographic position, etc. To avoid excessive energy and save for future use, battery is used to save the collected energy, but the capacity is usually limited and inconsistent availability [7]. It is possible that RSUs powered by harvested energy alone may not guarantee the quality of service. Thus, the use of conventional grid power cannot be eliminated. When renewable energy sources are not available, using grid energy as a substitute power supply can minimize grid consumption, which will help in reducing energy cost and harmful emission. Hence, there is a hybrid energy supplies formed by harvesting energy and reliable power (for

example, grid power) to ensure quality of service and the communication sustainability [8, 9]. In fact, the RSU is mainly powered by harvested energy and the grid is used as a backup.

In the above scenario, guaranteeing data timeliness under the constraint of reliable power consumption is a most important metrics when dealing with the packet scheduling optimization of each RSU. Nevertheless, this is not easy. First, in order to achieve efficient data dissemination, there is usually the cooperation of hybrid infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communication in CVIS, which makes the data conflicts in the RSU coverage area, causing the interference in the establishment of data communication links, and directly affecting the data transmission timeliness [10, 11]. Second, the RSU is encouraged to use the free and renewable energy whenever available, but the waiting time could be undesirably long if it only relies on the harvested energy. Third, even if a reliable energy is available when renewable energy is unavailable, power consumption limit makes it impossible to use indefinitely, and it will increase the waiting time. To address the above issues, it is necessary to jointly research data transmission delay-minimal and power-constrained trade-off problem to ensure the RSU-to-vehicle downlink data communication timeliness with using grid power as little as possible.

1.2. Related Works. In recent years, the use of energy harvesting RSUs in vehicular environment has gradually received the attention in the research literature. In [12], the design of a solar energy harvesting circuit RSUs is presented, and the usability of energy harvesting systems in vehicular networks is investigated in [13]. In [14], the authors use a smart scheduling to reduce RSU energy costs in green vehicular roadside infrastructure, and the authors of [8] assumed that RSUs are powered by both grid source and renewable energy, and proposed an efficient sleep scheduling method to an overall decrease of energy cost. Several works have been carried out recently for optimal scheduling in RSU-to-vehicle communication. The main contribution of [15] is to investigate the problem of scheduling the downlink communication from renewable energy-powered RSUs toward vehicles, with the objective of maximizing the number of served vehicles. For the purpose of optimizing the RSU's downlink traffic scheduling, a protocol for energy-efficient adaptive scheduling using reinforcement learning is proposed, focusing on guaranteeing the operation of the vehicular network [16]. Regarding to energy consumption and time delay minimal in the CVIS, packet scheduling optimization strategy for energy-delay trade-off in self-powered RSUs is proposed [17, 18]. And a rudimentary online scheduling algorithm for estimating the energy consumption, average packet delay and required battery capacity is proposed [19].

In the process of RSU-to-vehicle communication, it involves data arrival, queuing behaviour, power allocation and data transmission in RSUs, corresponding to the network layer, the data link layer and physical layer, respectively. The cross-layer has been proved as one of the most efficient solutions for wireless communications involving multiple layers [20–23]. The cross-layer method was firstly studied in [20], where the scheduling policy is studied for transmission under a time-varying channel with delay constraint. In [21], the sensing-throughput trade-off crosses physical layer and MAC layer in multi-channel cognitive radio networks. In [22], the authors achieved energy efficient and delay trade-off using cross-layer stochastic optimization approach. In addition, it is well applied to green vehicular networks [23].

1.3. Contributions. In this work, we address an optimal resource scheduling for the RSU powered by both grid-powered and energy harvesting in RSU-to-vehicle downlink communication. The objective is to minimize the packet queuing delay given the available power constraint with considering the connectivity probability between RSU and vehicle. To analyse the proposed scheme, we adopt the cross-layer framework by combining the data packet arrival in the network layer, queuing state in the MAC layer, and transmission over the wireless channel as well as the energy harvesting behaviour in the physical layer. Then, we formulate a probability scheduling, considering the joint state of random energy arrivals, random data packet arrivals and time-varying channel states. Based on the scheduling, we draw up a two-dimensional Markov chain by using both data queue and harvested energy queue as state variables, and formulate an optimal scheduling problem to describe the delay minimal and reliable power consumption trade-off. With the solution to the optimization problem, the optimal scheduling policy can be revealed as a threshold-based policy. At the end, our theoretical analysis is verified by simulations.

2. System Model

We consider a packet transmission scenario between RSUs relayed by vehicle is shown in Figure 1. For sake of reducing environmental pollution, the RSU is powered by two sources of energy namely, a fixed power line using the conventional grid energy and Photo Voltaic cells to utilize the solar energy, but grid power is only used to ensure that the RSU is always operational when necessary [8]. The collected data (such as surrounding traffic, monitoring information) is stored in the data buffer of RSU, waiting to be sent to another RSU with data aggregation centre in form of packet queuing through mobile vehicles transmission. To improve the connectivity probability of vehicular communication, vehicle may complete the whole data transmission through a wireless multi-hop path with V2V communication [24]. However, in this paper, we only focus on the downlink communication from RSU-to-vehicle. Considering the discrete time-slotted system, in a certain time slot, if RSU does not establish a downlink communication link with passing vehicles, the data packet will wait in the buffer. However, if the downlink link is available, RSU determines whether to send packets to passing vehicles and the corresponding power consumption according to the packet scheduling strategy.

In Figure 2, we formulate a cross-layer system model. At the RSU, the data packets (such as monitoring data)

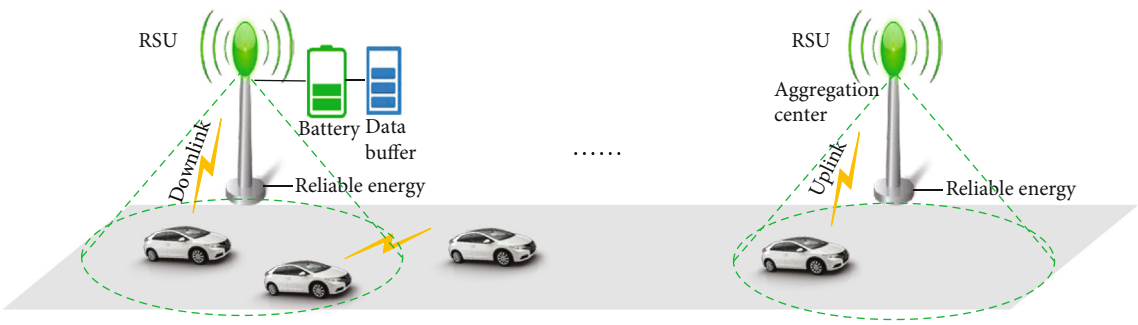


Figure 1: Schematic diagram of RSU-to-vehicle scene.

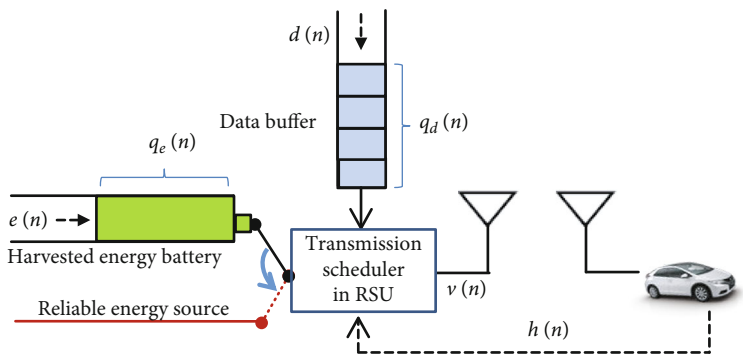


Figure 2: A cross-layer system model.

generated by the upper layers arrive at the MAC layer and are stored at a data buffer. A battery storing the free and renewable energy harvested from the environment is represented in a virtual queue. If a downlink link has been established in the slot, the RSU will combine the data buffer status, energy queue status and new packet arrival to determine whether to send data packet to the vehicle according to the packet scheduling strategy, so as to control the reliable energy consumption and queuing delay of data.

2.1. Packet Arrival Model. In discrete-time slotted system, assuming that packets are generated randomly, following the Bernoulli process, we can define that the data packets randomly arrive with rate η_d. Each data arrival contains $k_d \in \mathbb{N}^+$ packets. Let $d[n]$ denotes the number of packets arriving at the RSU at the beginning of the n-th timeslot. The mass probability function of $d[n]$ is given by

$$\begin{aligned} \Pr\{d[n] = k_d\} &= \eta_d \\ \Pr\{d[n] = 0\} &= \bar{\eta}_d, \end{aligned} \tag{1}$$

where $\eta_d \in [0, 1]$ and $\bar{\eta}_d = 1 - \eta_d$. At the RSU, a buffer is adopted to store the packets that cannot be sent immediately, and its capacity is expressed by Q_d. Let $v_d[n]$ and $q_d[n]$ note the number of packets transmitted in the n-th slot and the queue length at the end of n-th timeslot, respectively. Thus, the update of the data packet queue can be obtained as

$$q_d[n] = \min\{q_d[n-1] + d[n], Q_d\} - v_d[n] \tag{2}$$

In Figure 2, the RSU can harvest energy from the environment. The energy storage and consumption processes of the battery can be described as a queuing system. In particular, the harvested energy packets arrive randomly following a certain distribution, and leave the queue when the scheduler spends a few energy packets for data packet transmission. There also exists an upper limit Q_e for the virtual queue corresponding to the storage capacity of the battery. This means that the battery can store at most Q_e energy packets, each of which contains a certain amount of energy. $q_e[n]$is used to measure the length of the energy queue, i.e., the remaining energy in the battery. We use $e[n]$ describe the process of energy harvesting. Suppose that a certain amount of energy is harvested each time. Accordingly, $k_e \in \mathbb{N}^+$ energy packets arrive at the battery with probability η_e. Therefore, the mass probability function of $e[n]$ is given by

$$\begin{aligned} \Pr\{e[n] = k_e\} &= \eta_e \\ \Pr\{e[n] = 0\} &= \bar{\eta}_e, \end{aligned} \tag{3}$$

where $\eta_e \in [0, 1]$means the energy arrival rate, and $\bar{\eta}_e = 1 - \eta_e$. Let $v_e[n]$ notes number of energy packets consumed in n-th timeslot. Similarly, the update of the energy queue is characterized as

$$q_e[n] = \min\{q_e[n-1] + e[n], Q_e\} - v_e[n] \tag{4}$$

2.2. Transmission Channel Model. In Figure 1, all the data packets in the RSU will be transmitted to the vehicle through wireless channel, which is supported by dedicated short range communication (DSRC) protocol. However, in the

CVIS, the vehicle can only establish communication connection with RSU within its coverage. When the vehicle drives out the coverage of RSU, the connection will be disconnected. Therefore, there are two communication channel states of "*on*/*off*" between RSU and vehicle, which represent the working state and offline state between vehicle and RSU. Moreover, within the coverage aera of RSU, there are not only RSU-to-vehicle downlink transmission channels, but also vehicle-to-vehicle transmission channels, which may interfere with each other. It cannot be guaranteed that the downlink communication channels between RSU and vehicle are available in each time slot. So, in this paper, we use "*on*" and "*off*" to describe the channel state, which is denoted by $h[n]$ in the n-th timeslot. The mass distribution function of the state is given by

$$\begin{aligned} \Pr\left\{h[n] = {}'on'\right\} &= \alpha \\ \Pr\left\{h[n] = {}'off'\right\} &= \bar{\alpha}, \end{aligned} \tag{5}$$

where α represents the probability that the downlink channel remains working, namely "*on*," while $\bar{\alpha}$ represents the probability that the downlink channel remains offline, namely "*off*," and $\alpha + \bar{\alpha} = 1$ in n-th timeslot.

2.3. Probabilistic Scheduling Policy. For convenience of analyse, the RSU is supposed to successfully deliver one data packet with one energy packet from the energy queue or with grid power in watts when the channel state is "*on*." Usually, the scheduler prefers to use the harvested energy because it is free and renewable, which is also desirable. However, due to the uncertainty of external environment, it may have to wait a long time to harvest sufficient power for next packet transmission, which is intolerable for latency-sensitive application. In this situation, reliable power source should be adopted instead. However, if the RSU relies too much on the reliable power source, the communication cost will increase inevitably, which is inconsistent with the original intention. Thus, there exists a trade-off between the queuing delay and the transmission power drawn from the reliable energy source. In this work, the scheduler aims to find an optimal scheduling strategy to minimize the average queuing delay constrained by the grid power.

Let $\boldsymbol{v}[n] = (v_d[n], v_e[n])$ note the actions taken in the n-th slot, namely, transmitting $v_d[n]$data packets with $v_e[n]$ energy packets. The transmission channel state plays a key role, because it be possible to transmit data packets only when $h[n] = {}'on'$. Specifically, when the harvested energy is enough for one transmission, one data packet should always be transmitted to decrease the buffer occupation. Otherwise, the scheduler must decide whether to transmit with a certain grid power in current timeslot. Simultaneously, we define the joint state vector $\boldsymbol{s}[n] = (d[n], e[n], h[n])$ and the queue state vector $\boldsymbol{q}[n] = (q_d[n], q_e[n])$. Based on the packet arrival and queuing states, the scheduling policy can be described below in three cases.

Case 1. when $\boldsymbol{q}[n-1] = (i, 0), (i \geq 0)$,

$$\boldsymbol{v}[n] = \begin{cases} (1,1)\ w.p.1 & \boldsymbol{s}[n] = \left(\cdot, k_e, {}'on'\right) \\ (1,0)\ w.p.g_i & \boldsymbol{s}[n] = \left(k_d, 0, {}'on'\right) \\ (1,0)\ w.p.f_i & \boldsymbol{s}[n] = \left(0, 0, {}'on'\right) \\ (0,0) & otherwise \end{cases} \tag{6}$$

where "*w.p.*" is short for "with probability of" and the symbol "·" in the expression of $\boldsymbol{s}[n]$ means no matter how many packets (data packet or energy packet) arrive in this slot. Let $\{g_i\} \in [0,1]$ and $\{f_i\} \in [0,1]$ be the probabilities of packet transmission with or without packet arrivals in this time slot, respectively. Accordingly, it is also possible for the RSU to remain silent with probabilities $1 - g_i$ and $1 - f_i$, respectively. In this case, there are data packets waiting in the data buffer but no renewable energy packet can be used for transmitting in RSU. Taking the first term in (6) for example, when the channel is "*on*," regardless of the data packet arrival state, one packet will be delivered $v_d[n]$ at the cost of one energy packet $v_e[n]$ if k_e energy packets being harvested. As for the second term, there is no energy packet newly harvested. Thus, the scheduler decides to transmit one data packet $v_d[n] = 1$ with probability $g_i (i \geq 0)$ using paid reliable power $v_e[n] = 1$, otherwise, remain silent $v_d[n] = 0$ with probability $1 - g_i$. Similarly, when no data packets newly arrive, the RSU would transmit one packet using the paid power with probability $f_i (i > 0)$ or remain silent $v_d[n] = 0$ with probability $1 - f_i$, respectively. Whether the packets are reached or not, only if the channel state is offline, then $\boldsymbol{v}[n] = (0, 0)$.

Case 2. when $\boldsymbol{q}[n-1] = (i, j), (i, j > 0)$,

$$\boldsymbol{v}[n] = \begin{cases} (1,1)\ w.p.1 & \boldsymbol{s}[n] = \left(\cdot, \cdot, {}'on'\right) \\ (0,0)\ w.p.1 & \boldsymbol{s}[n] = \left(\cdot, \cdot, {}'off'\right) \end{cases} \tag{7}$$

In this case, both data buffer and energy queue are not empty. Then, no matter whether the new data and harvested energy packets arrive in n-th slot, a data packet will be delivered with an energy packet if the downlink channel is available.

Case 3. when $\boldsymbol{q}[n-1] = (0, j), (j > 0)$,

$$\boldsymbol{v}[n] = \begin{cases} (1,1)\ w.p.1 & \boldsymbol{s}[n] = \left(k_d, \cdot, {}'on'\right) \\ (0,0)\ w.p.1 & \boldsymbol{s}[n] = \left(0, \cdot, {}'on'\right) \\ (0,0)\ w.p.1 & \boldsymbol{s}[n] = \left(\cdot, \cdot, {}'off'\right) \end{cases} \tag{8}$$

In this case, there are enough harvested energy storing at the battery but the data queue is zero. Then, one energy

Table 1: The one-step transition probabilities $\Pr\{\boldsymbol{q}[n]|\boldsymbol{q}[n-1]\}$.

$\boldsymbol{q}[n\text{-}1]$	$\boldsymbol{q}[n]$	$\Pr\{\boldsymbol{q}[n]\|\boldsymbol{q}[n-1]\}$	$\boldsymbol{q}[n\text{-}1]$	$\boldsymbol{q}[n]$	$\Pr\{\boldsymbol{q}[n]\|\boldsymbol{q}[n-1]\}$
(i, j)	$(i\text{-}1, (j+k_e)\vee Q_e\text{-}1)$	$\rho_0\alpha$	(i, j)	$(i, j+k_e)$	$\rho_0\bar{\alpha}$
(i, j)	$(i\text{-}1, j\text{-}1)$	$\rho_1\alpha$	(i, j)	(i, j)	$\rho_1\bar{\alpha}$
(i, j)	$(i+k_d\text{ -}1, j\text{-}1)$	$\rho_2\alpha$	(i, j)	$(i+k_d, j)$	$\rho_2\bar{\alpha}$
(i, j)	$(i+k_d\text{ -}1, (j+k_e)\vee Q_e\text{-}1)$	$\rho_3\alpha$	(i, j)	$(i+k_d, j+k_e)$	$\rho_3\bar{\alpha}$
$(i, 0)$	$(i+k_d\text{ -}1, 0)$	$\rho_2\alpha g_i$	$(i, 0)$	$(i+k_d, 0)$	$\rho_2\alpha(1-g_i)+\rho_2\bar{\alpha}$
$(i, 0)$	$(i\text{-}1, 0)$	$\rho_1\alpha f_i$	$(i, 0)$	$(i, 0)$	$\rho_1\alpha(1-f_i)+\rho_1\bar{\alpha}$
$(0, j)$	$(0, (j+k_e)\vee Q_e)$	ρ_0	$(0, j)$	$(0, j)$	ρ_1
(i, Q_e)	$(i\text{-}1, Q_e\text{-}1)$	$\bar{\eta}_d\alpha$	(i, Q_e)	$(i+k_d\text{-}1, Q_e\text{-}1)$	$\eta_d\alpha$
(i, Q_e)	(i, Q_e)	$\bar{\eta}_d\bar{\alpha}$	(i, Q_e)	$(i+k_d, Q_e)$	$\eta_d\bar{\alpha}$
$(i, Q_e\text{-}1)$	(i, Q_e)	$\rho_0\bar{\alpha}$	$(0, Q_e)$	$(0, Q_e)$	$\bar{\eta}_d$
$(0, Q_e\text{-}1)$	$(0, Q_e)$	ρ_0	$(0, 0)$	$(0, 0)$	ρ_1

packet is spent for delivering one data packet newly arriving. And if no packets arrive at n-th slot, or the channel is offline, there is no energy consumption.

2.4. Markov Chain Model. Based on the probabilistic scheduling policy, we know that the transmission decision is made only in each time slot based on the queuing state $\boldsymbol{q}[n-1]$ and the joint state$\boldsymbol{s}[n]$, Hence, the queuing system can be characterized by a two-dimensional Markov chain with $\boldsymbol{q}[n]$ being the state variables $\{(i,j)|i=0,\cdots,Q_d\,;j=0,\cdots,Q_e\}$. For simplicity, we firstly define the following four joint probability parameters about packets arrival, shown as

$$\begin{cases} \rho_0 = \Pr\{d[n]=0, e[n]=k_e\} = \bar{\eta}_d\eta_e \\ \rho_1 = \Pr\{d[n]=0, e[n]=0\} = \bar{\eta}_d\bar{\eta}_e \\ \rho_2 = \Pr\{d[n]=k_d, e[n]=0\} = \eta_d\bar{\eta}_e \\ \rho_3 = \Pr\{d[n]=k_d, e[n]=k_e\} = \eta_d\eta_e \end{cases} \tag{9}$$

According to the scheduling scheme defined by the formulas (6)-(8), we focus on deriving the one-step state transition probability $\Pr\{\boldsymbol{q}[n]|\boldsymbol{q}[n-1]\}$ between states, which are listed in Table 1 in detail. In Table 1, the $(j+k_e)\vee Q_e$ denotes $\min\{(j+k_e), Q_e\}$, and the state transition probabilities locating below dashed line are the special case what i=0 or $j=\{0, Q_e-1, Q_e\}$. Except for the special states, the rest state transition probabilities $\Pr\{\boldsymbol{q}[n]|(i,j)\}$ are exhibited above the dashed line. In particular, the case $j=Q_e$ requires special processing, because the battery is full and the newly harvested energy must be discarded. Besides, the case $j=Q_e-1$ also discard the remaining packets when the packet number of newly harvested energy exceeds 1.

Based on Table 1, we construct a 2D Markov chain model shown in Figure 3, where solid lines present the fixed state transitions while all dotted lines indicate state transitions that vary with different k_d and k_e. To check the transition probability given in Table 1, assume $k_d=k_e=2$in Figure 3. Let $m, n\in\{0,1,2,3\}$, we modify the definition of variable α as, $\alpha_m=\rho_m\alpha$, $\bar{\alpha}_m=\rho_m\bar{\alpha}$ in Figure 3. Moreover, the scheduler makes a decision of transmission with a probability only under the situation that there are no harvested energy packets can be used, that is the state transition between states $(i,0)$. We define four variables $\lambda_{1,i}=\rho_2\alpha g_i$, $\lambda'_{1,i}=\rho_2\alpha(1-g_i)+\rho_2\bar{\alpha}$, $\mu_{1,i}=\rho_1\alpha f_i$, $\mu'_{1,i}=\rho_1\alpha(1-f_i)+\rho_1\bar{\alpha}$ on the first column in Figure 3.

For ease of discussion, we give the simplified Markov chain which is shown in Figure 4, where the transition probabilities are obtained from the substitution of $k_d=1$, $k_e=1$ in Figure 3. For example, the transition from (i,j) to $(i+k_d-1, j)$ in Figure 3 becomes that from (i,j) to (i,j) in Figure 4, for $i,j>0$. For this state transition, one is that both data and energy packet arrive and the transmission link is available so that a data is delivered with the corresponding probability of α_3, the other is neither data nor energy packet arrives and the transmission link is unavailable, so that no backlogged data packet is transmitted with the corresponding probability $\bar{\alpha}_1$. Therefore, a new notation is needed for the transition from (i,j) to (i,j), which $\bar{\alpha}_1+\alpha_3$, namely, α', for $i,j>0$. The transition between these states in Figure 4 can easily derived based on the dynamic change of the data buffer queue and the virtual energy queue as given in (2) and (4).

Denote by $\boldsymbol{\pi}_i$, $\boldsymbol{\pi}_j$ and $\pi_{(i,j)}$ the steady-state probabilities of the data queue state, the energy queue state and the joint queue state, respectively. The relationship between the steady-state probabilities of the 2D Markov chain can be represented in the following theorem.

Theorem 1. *Define $\tilde{\mu}_{1,i}=\rho_1\alpha f_i+\alpha_0$, then the local balance equations of the Markov chain is obtained as*

$$\begin{aligned} &\tilde{\mu}_{1,i}\pi_{(i,0)}+\alpha_0\sum_{j=1}^{Q_e}\pi_{(i,j)}+\alpha_1\sum_{j=1}^{Q_e}\pi_{(i,j)} \\ &\quad=\pi_{(i-1,0)}\lambda'_{1,i-1}+\bar{\alpha}_2\sum_{j=1}^{Q_e}\pi_{(i-1,j)}+\bar{\alpha}_3\sum_{j=0}^{Q_e}\pi_{(i-1,j)} \end{aligned} \tag{10}$$

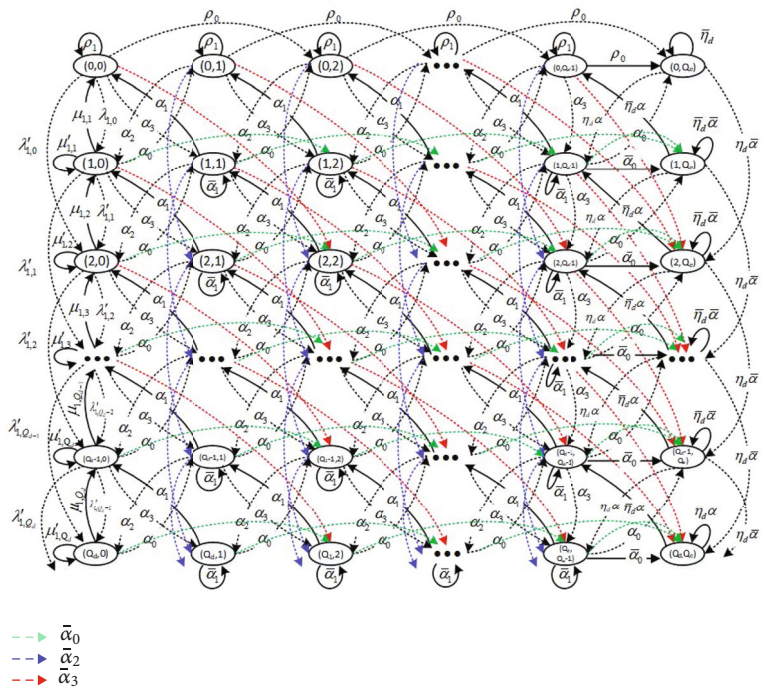


Figure 3: The two-dimensional Markov chain model.

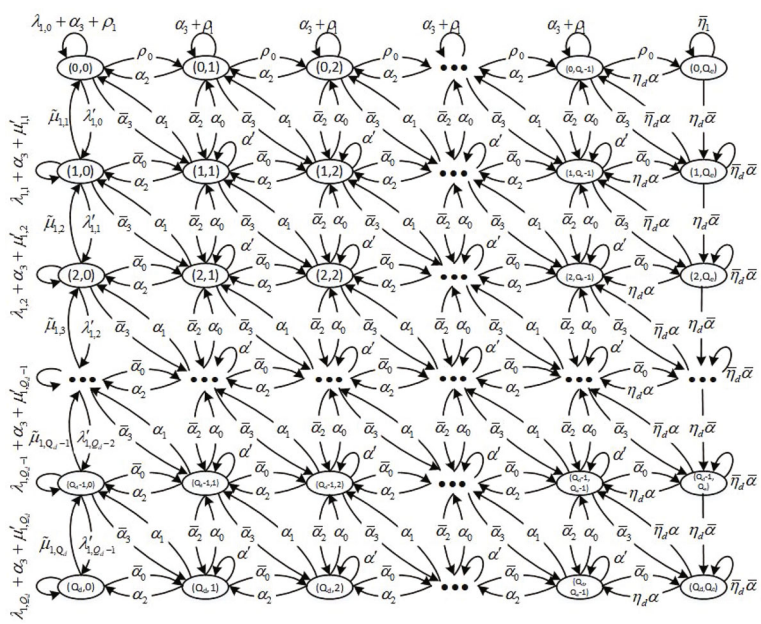


Figure 4: The two-dimensional Markov chain model.

The expression of steady-state probabilities of the data queue $\boldsymbol{\pi}_i$ can be derived as

$$\boldsymbol{\pi}_i = \sum_{j=0}^{Q_e} \pi_{(i,j)} \tag{11}$$

With the definition of $\boldsymbol{\pi}_i$, $\alpha_0 + \alpha_1 = \bar{\eta}_d \alpha$, and $\bar{\alpha}_2 + \bar{\alpha}_3 = \eta_d \bar{\alpha}$, we can then reformulate (10), as following

$$\bar{\eta}_d \alpha \boldsymbol{\pi}_i - \eta_d \bar{\alpha} \boldsymbol{\pi}_{i-1} = \left(\lambda'_{1,i-1} - \bar{\alpha}_2\right) \pi_{(i-1,0)} + \left(\bar{\eta}_d \alpha - \tilde{\mu}_{1,i}\right) \pi_{(i,0)} \tag{12}$$

Let $\boldsymbol{\Lambda}$ denote the transition probability matrix, we know that the steady-state probability $\pi_{(i,j)}$ should meet constraint $\boldsymbol{\Lambda}\boldsymbol{\pi} = \boldsymbol{\pi}$, where vector $\boldsymbol{\pi}$ is defined as $[\pi_{(0,0)}, \pi_{(0,1)} \cdots \pi_{(0,Q_e)}, \pi_{(1,0)}, \pi_{(1,1)} \cdots \pi_{(1,Q_e)}, \cdots, \pi_{(Q_d,0)}, \pi_{(Q_d,1)} \cdots \pi_{(Q_d,Q_e)}]$. Also, $\pi_{(i,j)}$

should meet the normalization constraint, namely, $\sum_{i=0}^{Q_d}\sum_{j=0}^{Q_e}\pi_{(i,j)} = 1$.

3. Establishment of Optimization Problem

At the above section, we have illustrated the fundamental trade-off between queuing delay and transmission power paid at cost in the RSU. It is of great significance to derive the average queuing delay and the power consumption mathematically. In this section, we will discuss the delay and power metrics based on the properties of two-dimensional Markov chain model in Figure 4.

3.1. The Delay and Power Metrics. Since the average queuing delay is associated with the occupancy rate of the data queue, combined with the steady-state probabilities of the data queue, the average occupancy rate of the data queue can be obtained as $\sum_{i=1}^{Q_d} i\boldsymbol{\pi}_i$. The average packet arrival rate is equal to $k_d\eta_d$ according to (1). Thus, the average queuing delay is derived based on Little's Law [25] as

$$\boldsymbol{D}_{av} = \frac{1}{k_d\eta_d}\sum_{i=1}^{Q_d} i\boldsymbol{\pi}_i \tag{13}$$

Next, we discuss the analytical expressions of the average power consumption. Notice that only when no harvested energy but data packets waiting for transmission, the reliable energy source will be possibly used if the transmission channel is available. Thus, the power consumption from grid-powered only depends on two probabilities $\lambda_{1,i}$ and $\mu_{1,i}$.

Theorem 2. *The average power consumption $\boldsymbol{P}_{av}$ drawn from the reliable energy source is obtained as*

$$\boldsymbol{P}_{av} = \sum_{i=0}^{Q_d}\xi_i\pi_{(i,0)} - \sum_{i=0}^{Q_d}\zeta_i\pi_{(i,1)}, \tag{14}$$

where

$$\begin{aligned}\xi_0 &= (\rho_0 + \bar{\alpha}_3)Q_d + \rho_2\\ \xi_i &= \eta_e\bar{\alpha}(Q_d - i) + \rho_2 - \alpha_0\\ \zeta_0 &= \alpha_2 Q_d\\ \zeta_i &= \bar{\eta}_e\alpha(Q_d - i) + \alpha_1\end{aligned} \tag{15}$$

for $i \in \{0, 1, \cdots, Q_d\}$.

Proof. As discussed above, the normalized average power consumption from grid-powered can be obtained as

$$\boldsymbol{P}_{av} = \sum_{i=0}^{Q_d}\pi_{(i,0)}\rho_2\alpha g_i - \sum_{i=1}^{Q_d}\pi_{(i,0)}\rho_1\alpha f_i \tag{16}$$

By inserting $\rho_2\alpha g_i = \rho_2 - \lambda'_{1,i-1}$ and $\rho_1\alpha f_i = \tilde{\mu}_{1,i} - \alpha_0$ into (16) and extracting $\boldsymbol{P}_{av}$, we arrive at

$$\boldsymbol{P}_{av} = \rho_2\sum_{i=0}^{Q_d}\pi_{(i,0)} - \alpha_0\sum_{i=1}^{Q_d}\pi_{(i,0)} + \sum_{i=1}^{Q_d}\left(\pi_{(i,0)}\tilde{\mu}_{1,i} - \pi_{(i-1,0)}\lambda'_{1,i-1}\right). \tag{17}$$

From the Markov chain shown in Figure 4, the local balance equation at state $(i,0)$ can be expressed as

$$\pi_{(i+1,0)}\tilde{\mu}_{1,i+1} - \pi_{(i,0)}\lambda'_{1,i} = \begin{cases}(\rho_0 + \bar{\alpha}_3)\pi_{(i,0)} - \alpha_2\pi_{(i,1)} - \alpha_1\pi_{(i+1,1)} & i = 0\\ (\tilde{\mu}_{1,i} + \eta_e\bar{\alpha})\pi_{(i,0)} - \alpha_2\pi_{(i,1)} & \\ -\pi_{(i-1,0)}\lambda'_{1,i-1} - \alpha_1\pi_{(i+1,1)} & i \geq 1\end{cases} \tag{18}$$

For the partition function, we rewrite the second section and obtain

$$\begin{aligned}\pi_{(i+1,0)}\tilde{\mu}_{1,i+1} - \pi_{(i,0)}\lambda'_{1,i} &= \left(\pi_{(i,0)}\tilde{\mu}_{1,i} - \pi_{(i-1,0)}\lambda'_{1,i-1}\right)\\ &\quad + \eta_e\bar{\alpha}\pi_{(i,0)} - \alpha_2\pi_{(i,1)} - \alpha_1\pi_{(i+1,1)}\end{aligned} \tag{19}$$

for all $i \geq 1$. Through recursion of (19) and by inserting $\pi_{(1,0)}\tilde{\mu}_{1,1} - \pi_{(0,0)}\lambda'_{1,0}$ given by (18), we can further get

$$\begin{aligned}\pi_{(i+1,0)}\tilde{\mu}_{1,i+1} - \pi_{(i,0)}\lambda'_{1,i} &= \eta_e\bar{\alpha}\sum_{m=1}^{i}\pi_{(m,0)} - \alpha_2\sum_{m=0}^{i}\pi_{(m,1)}\\ &\quad - \alpha_1\sum_{m=1}^{i}\pi_{(m,1)} + (\rho_0 + \bar{\alpha}_3)\pi_{(0,0)}\end{aligned} \tag{20}$$

By substituting (20) into (17), we can express the average power $\boldsymbol{P}_{av}$ as

$$\begin{aligned}\boldsymbol{P}_{av} &= ((\rho_0 + \bar{\alpha}_3)Q_d + \rho_2)\pi_{(0,0)} - \alpha_2 Q_d\pi_{(0,1)}\\ &\quad + \sum_{i=1}^{Q_d}(\eta_e\bar{\alpha}(Q_d - i) + \rho_2 - \alpha_0)\pi_{(i,0)}\\ &\quad - \sum_{i=1}^{Q_d}(\bar{\eta}_e\alpha(Q_d - i) + \alpha_1)\pi_{(i,1)}\end{aligned} \tag{21}$$

which perfectly corresponds to the (14) and (15). □

Therefore, by eliminating the dependence of $\boldsymbol{P}_{av}$ on the parameters g_i and f_i, the $\boldsymbol{P}_{av}$ have converted into a linear function of the steady-state probabilities $\pi_{(i,j)}$. Since the analytical expressions of both average queuing delay and average power consumption are determined by steady-state probability, we can achieve the excepted scheduling strategy by optimizing $\{\pi_{(i,j)}\}$, which is debated in Section 4.

3.2. Delay Minimization Problem under Power Constrained. We can find that the average delay is a linear combination of the steady-state probabilities in (13), so it can be minimized by restricting the steady-state probabilities. In addition, the

average power consumption is also a linear function of the steady-state probabilities and coefficients are related to the transition probabilities in (21). Furthermore, the relationship between power consumption limitation and delay optimization can be established through the steady-state probabilities. Meanwhile, some transition probabilities can be expressed by the transmission parameters g_i and f_i, so we can get several constraints.

Due to $g_i \in [0, 1]$ and $f_i \in [0, 1]$, the transition probabilities $\lambda'_{1,i} = \rho_2\alpha(1 - g_i) + \rho_2\bar{\alpha}$ and $\tilde{\mu}_{1,i} = \rho_1\alpha f_i + \alpha_0$ should satisfy the following inequalities

$$\rho_2\bar{\alpha} \le \lambda'_{1,i} \le \rho_2 \tag{22}$$

$$\alpha_0 \le \tilde{\mu}_{1,i} \le \bar{\eta}_d\alpha \tag{23}$$

In (22), we get the minimum of $\lambda'_{1,i}$ called $\rho_2\bar{\alpha}$ when $g_i = 1$ and the maximum called ρ_2 when$g_i = 0$. However, we get the minimum of $\tilde{\mu}_{1,i}$ called α_0 when $f_i = 0$ and the maximum called $\bar{\eta}_d\alpha$ when $f_i = 1$ in (23). Consequently, recalling the local balance equation (12), we can obtain the following the steady-state probabilities constraint

$$\gamma\boldsymbol{\pi}_{i-1} \le \boldsymbol{\pi}_i \le \gamma\boldsymbol{\pi}_{i-1} + \bar{\eta}_e\pi_{(i,0)} + \delta\bar{\eta}_e\pi_{(i-1,0)} \tag{24}$$

where $\delta = \eta_d/\bar{\eta}_d$ and $\gamma = \bar{\alpha}\eta_d/\alpha\bar{\eta}_d$. Let $\boldsymbol{P}_{\max}$ denotes the upper bound of allowable power from reliable energy source. Based on the above constraints and the scheduling probability as the optimization variable, the average delay minimum problem under given power restriction is established as follows

$$\min_{\pi_{(i,j)}} \boldsymbol{D}_{av} = \frac{1}{k_d\eta_d}\sum_{i=1}^{Q_d} i\boldsymbol{\pi}_i \qquad (a)$$

$$s.t.\begin{cases} \sum_{i=0}^{Q_d}\xi_i\pi_{(i,0)} - \sum_{i=0}^{Q_d}\zeta_i\pi_{(i,1)} \le \boldsymbol{P}_{\max} & (b) \\ \gamma\boldsymbol{\pi}_{i-1} \le \boldsymbol{\pi}_i \le \gamma\boldsymbol{\pi}_{i-1} + \bar{\eta}_e\pi_{(i,0)} + \delta\bar{\eta}_e\pi_{(i-1,0)} & (c) \\ \pi_{(i,j)} \ge 0 & (d) \\ \sum_{i=0}^{Q_d}\sum_{j=0}^{Q_e}\pi_{(i,j)} = 1 & (e) \\ \boldsymbol{\Lambda\pi} - \boldsymbol{\pi} = \boldsymbol{0} & (f) \end{cases} \tag{25}$$

In problem (25), the objective and the first constraint are exactly the average queuing delay and the paid power consumption. Constraint (25.c) represents the mapping from the scheduling probabilities to the steady-state probabilities. Constraints (25.d)- (25.f) indicate the properties of the Markov chain. Problem (25) is a linear programming (LP) problem with the steady-state probabilities being the variables. With the optimal solution$\pi^*_{(i,j)}$, we can obtain the optimal scheduling probabilities g_i^* and f_i^* as shown in the sequel.

4. Solution of Optimization Problem

In this subsection, we will discuss how to seek the optimal solution $\pi^*_{(i,j)}$ of the 2-D Markov chain to attempt solving the optimization problem (25), and how to further derive all optimal transmission parameters $\{g_i^*\}$ and $\{f_i^*\}$.

4.1. The Optimal Steady-State Probability. The (25) is a formulation of the LP problem with the stead-state probabilities being the variable. However, due to the complexity of 2-D state transitions, it is difficult to derive a closed-form optimal solution of the corresponding LP problem.

For the purpose of reducing latency, the scheduling policy should send data packets continually, which makes the queue of data packet not too large. On the other side, for the purpose of cutting down the average power consumption from the reliable energy source, the scheduling policy prefers to wait for the harvested energy when the energy queue is none, which leads to an increase in the length of data queue. As a result, there exists an optimal threshold imposed on the data queue length, $i^* \in \{0, 1,\cdots,Q_d\}$. When the data queue length exceeds the threshold, the source transmits using the reliable energy, otherwise it waits for the harvested energy. Therefore, the transmission could consume more energy from the reliable energy source with the threshold decreasing. It further shows that the optimal threshold is determined by the power constraint $\boldsymbol{P}_{\max}$ from the reliable energy source.

Given all of that, combining with (24), we can obtain the optimal solution $\pi^*_{(i,j)}$ corresponding to the threshold i^* and $\boldsymbol{P}_{\max}$ based data transmission scheme [9], as follows

$$\begin{gathered} ((\rho_0 + \bar{\alpha}_3)Q_d + \rho_2)\pi_{(0,0)} - \alpha_2 Q_d\pi_{(0,1)} = \boldsymbol{P}_{\max}, \\ \gamma\sum_{j=0}^{Q_e}\pi^*_{(i-1,j)} - \sum_{j=0}^{Q_e}\pi^*_{(i,j)} = 0, (1 \le i \le i^* - 1), \\ \gamma\sum_{j=0}^{Q_e}\pi^*_{(i-1,j)} + \bar{\eta}_e\left(\pi^*_{(i,0)} + \delta\pi^*_{(i-1,0)}\right) - \sum_{j=0}^{Q_e}\pi^*_{(i,j)} = 0,\ (i^* + 1 \le i \le Q_d)\,. \end{gathered} \tag{26}$$

In order to derive all $\{\pi^*_{(i,j)}, 0 \le i \le Q_d, 0 \le j \le Q_e\}$ from (26), a multidimensional matrix calculation is constructed by adding two constraints (25.e) and (25.f)

$$\boldsymbol{\Gamma}_{N\times N} \cdot \boldsymbol{\pi}^*_{N\times 1} = [\boldsymbol{P}_{\max}, 0,\cdots,1]^T_{1\times N}, \tag{27}$$

where $\boldsymbol{\Gamma}$ is a $N \times N$ coefficient matrix, and $N = (1 + Q_d)(1 + Q_e)$. Through (27), we can calculate all optimal steady-state probabilities $\{\pi^*_{(i,j)}\}$ by solving N independent linear equations.

4.2. The Optimal Transmission Parameters. For ease of discussion, we suppose $\boldsymbol{P}_{con}$ is the power consumption when the RSU always use the reliable power for transmitting. Due to the limit of $\boldsymbol{P}_{\max}$, there are two different situations [9], as following.

Situation 1: $\boldsymbol{P}_{con} \le \boldsymbol{P}_{\max}$, which means there are enough reliable power can be used for transmitting when it needs. In this case, the optimal threshold satisfies $i^* = 0$. It presents that the RSU delivers one packet if the data buffer is non-empty and the transmission channel is "*on*," without concern for whether the harvested energy queue is empty. Furthermore, the optimal transmission parameters are given by $g_i^* = 1(i^* \ge 0)$ and $f_i^* = 1(i^* > 0)$.

Situation 2: $\boldsymbol{P}_{con} > \boldsymbol{P}_{\max}$, which means the reliable power is limit. Then, it is necessary to balance the use of energy from the reliable energy source to achieve the optimal problem (25). In this case, the utilization of power from the reliable energy source could happen only in the case $\boldsymbol{q}[n-1] = (i, 0)$ when new data packets arrive and the transmission channel is available but the harvested energy queue is empty. According to the above threshold strategy, if $(q_d[n-1] + d[n]) > i^*$ and $\boldsymbol{s}[n] = (\cdot, \cdot, 'on')$, the source should transmit data packet with probability "1" even though no harvested energy. So, the optimal transmission parameters g_i^* and f_i^* are set to 1. In contrast, if $(q_d[n-1] + d[n]) < i^*$, the source need to silently wait for the harvested energy whether new data arrives or not. Here, g_i^* andf_i^* are set to 0. But, when $(q_d[n-1] + d[n]) = i^*$, the source transmits using the reliable power with $g_i^* < 1$ and $f_i^* < 1$ in order to obtain the minimum average delay $\boldsymbol{D}_{av}^*$ under the power constraint $\boldsymbol{P}_{\max}$.

In order to obtain g_i^* and f_i^* at the optimal threshold, we recall the local balance equation with the optimal steady-state probabilities $\pi_{(i,j)}^*$ at state (i, j), and have

$$\bar{\eta}_d \alpha \boldsymbol{\pi}_{i^*}^* - \eta_d \bar{\alpha} \boldsymbol{\pi}_{i^*-1}^* = \left(\lambda_{1,i^*-1}' - \bar{\alpha}_2\right) \pi_{(i^*-1,0)}^* + \left(\bar{\eta}_d \alpha - \tilde{\mu}_{1,i^*}\right) \pi_{(i^*,0)}^* \tag{28}$$

By substituting $\lambda_{1,i^*-1}' = \rho_2(1 - \alpha g_{i^*-1}^*)$ and $\tilde{\mu}_{1,i^*} = \rho_1 \alpha f_{i^*}^* + \alpha_0$ into (28), we can get

$$\rho_2(1 - \alpha g_{i^*-1}^*)\pi_{(i^*-1,0)}^* = \bar{\eta}_d \alpha \boldsymbol{\pi}_{i^*}^* - \eta_d \bar{\alpha} \boldsymbol{\pi}_{i^*-1}^* - \alpha_1(1 - f_{i^*}^*)\pi_{(i^*,0)}^* + \bar{\alpha}_2 \pi_{(i^*-1,0)}^* \tag{29}$$

In (29), $f_{i^*}^*$ denotes the optimal transmission probability when $\boldsymbol{q}[n-1] = (i^*, 0)$ and $\boldsymbol{s}[n] = (\cdot, \cdot, 'on')$. Although the data queue length has reached the threshold and no harvested energy is available at the end of last timeslot, no new data packet arrives at the n-th timeslot. For the convenience, $f_{i^*}^* = 0$ is acceptable and reasonable. So that, we can get all transmission parameters as follow

$$g_i^* = \begin{cases} 0 & i < i^* - 1 \\ 1 - \dfrac{\bar{\eta}_d \alpha \boldsymbol{\pi}_i^* - \eta_d \bar{\alpha} \boldsymbol{\pi}_{i-1}^* - \alpha_1 \pi_{(i,0)}^*}{\alpha_2 \pi_{(i-1,0)}^*} & i = i^* - 1 \\ 1 & i > i^* - 1 \end{cases} \tag{30}$$

$$f_i^* = \begin{cases} 0 & i \le i^* \\ 1 & i > i^* \end{cases} \tag{31}$$

From (30) and (31), the transmit probabilities are only determined by the comparison between the data queue length and the threshold value. Due to the limit of reliable power consumption, RSU waits for the harvested energy when the data queue length is less than or equal to the threshold, but when the data queue length exceeds the threshold, one packet is sent by the reliable power over downlink wireless channel. Therefore, the optimal scheduling policy can be described as a threshold-based policy.

5. Simulation Results

In simulations, the data packet and energy packet arrival processes in the RSU are modelled by generating two Bernoulli random variables with the parameters η_d and η_e at the beginning of each time slot, respectively. The number of newly arrived data and energy packets is separately less than 2 in each time slot. We adopt two-state "*on/off*" channel model to describe the RSU-to-vehicle downlink communication link state. The packet transmission is scheduled according to the optimal transmission parameters $\{g_i^*\}$ and $\{f_i^*\}$ in (30) and (31). Each simulation runs over 10^6 time slots.

We simulate and discuss the optimal queuing delay and reliable power consumption trade-off performance in terms of battery capacity, packet arrival rate and channel availability rate. The results are shown in Figures 5–7. Figure 5 plots the trade-off under different queue length of harvested energy Q_e, where the data packet and energy arrival rates in RSU are same, as $\eta_d = \eta_e = 0.3$. In Figure 5, the theoretical results are presented as the lines, while the simulation results are marked by symbol "∘." In addition, we use the Linprog optimization tool to solve the LP problem (25) in order to further verify the correctness of the theoretical solution, and use the maker "+" to demonstrate the Linprog results. We can see that theoretical results are in good agreement with the simulation and Linprog results from Figure 5. It is observed that the average minimum queuing delay decreases when the average power increases, and when the available power decreases and approaches a critical value, the queuing delay will increase dramatically and grow to infinity. Besides, the decreasing rate grows with the increase of the battery capacity Q_e. This means that a larger Q_e leads to a much smaller queuing delay, since less harvested energy is wasted due to the limitation of the battery capacity. For the sake of brevity, the Linprog results will be omitted in the following simulation diagrams.

Figure 6 focuses on the impact of different energy arrival rates η_e on the trade-off curve, where the data packet arrival rates is set to $\eta_d = 0.3$. From Figure 6, we show that the comparison between the energy arrival rate and the data package arrival rate has a great influence on the whole delay-power trade-off curve. Given the identical queue length and data packet arrival rate, a greater average power will be induced for lower harvested energy arrival rate, besides that the

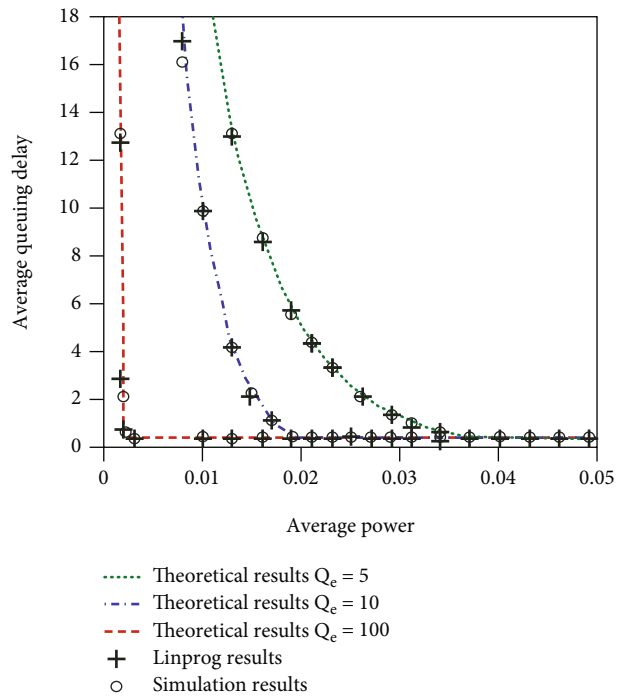


Figure 5: The optimal delay-power curves for different Q_e. ($\eta_d = \eta_e = 0.3$, $Q_d = 45$).

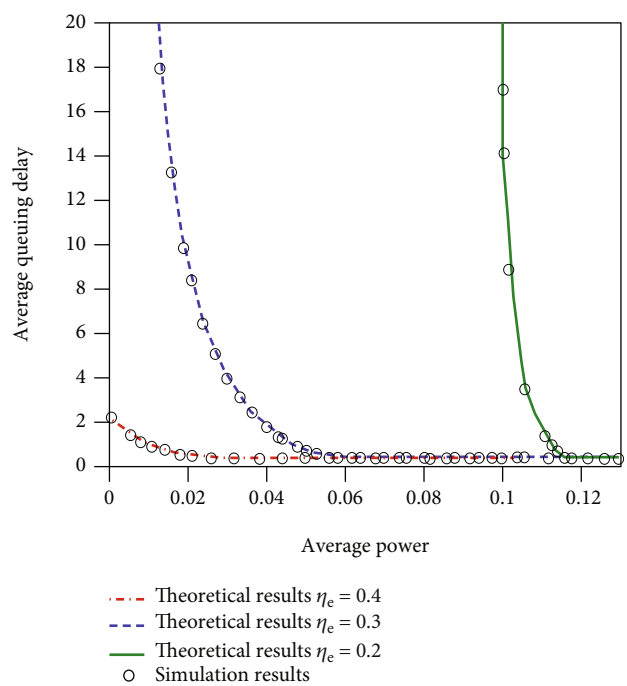


Figure 6: The optimal delay-power curves for different energy harvesting rate η_e ($\eta_d = 0.3$, $Q_d = 45$, $Q_e = 5$).

delay-power curves change steeper. It is because that the RSU must exploit more extra reliable energy source power to transmit backlogged packets waiting when the energy arrival rate is less than the data package arrival rate $\eta_e < \eta_d$. Conversely, the RSU has enough harvested energy to transmit data packets when $\eta_e > \eta_d$. Moreover, there has the less average delay even if $\boldsymbol{P}_{\max} = 0$.

Figure 7 shows the optimal queuing delay and reliable power consumption trade-off under different probabilities of RSU-to-vehicle downlink availability rate, α =0.5, 0.6, 0.8 and 1. The average queuing delay decreases with the average power increasing, as expected. For each numerical curve, when the reliable power become large enough which means there always exists energy for transmitting if needed,

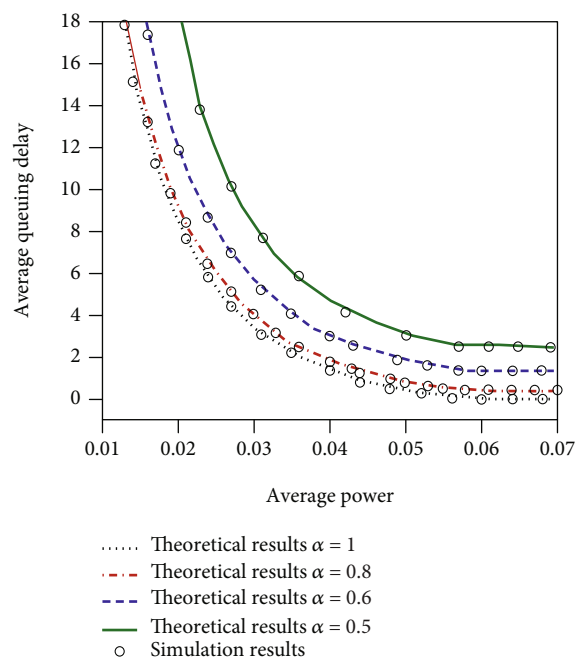


Figure 7: The delay-power curve for different α. ($\eta_d = \eta_e = 0.3$).

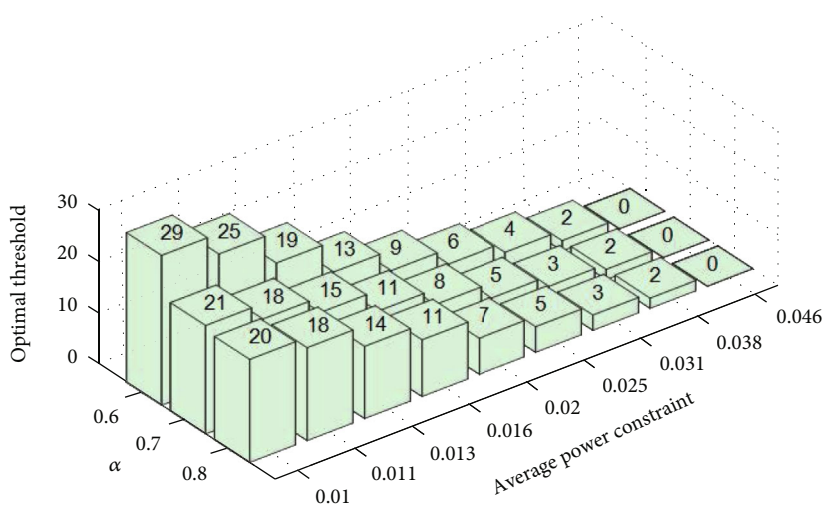
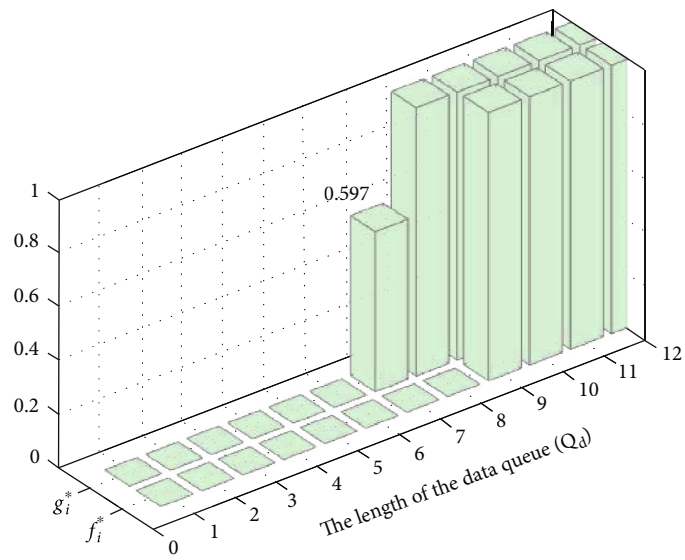


Figure 8: The threshold structure of optimal scheduling policy.

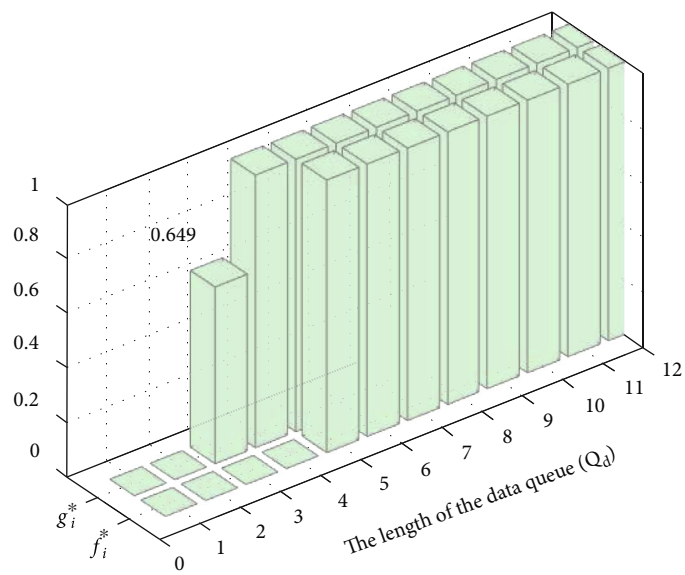
the average queuing delay of RSU only depends on the data packets arrival and the available rate of the channel and remains same. Meanwhile, we can see that, the smaller the availability rate of the channel is, the greater the average queuing delay is. This is because more timeslots will stay in the "*on*" state and can be used for transmitting.

Next, we verify that the optimal scheduling policy is determined by the data queue threshold i^* and the transmission parameters $\{g_i^*\}$ and $\{f_i^*\}$. Figure 8 shows the optimal threshold i^* value responding to the assigned average reliable power constraint $\boldsymbol{P}_{\max}$ when the availability rate of downlink channel α=0.6, 0.7 and 0.8, respectively. The threshold-power curve shows ladder-type decline in whichever given α. With the increasing of the disposable power constraint, the optimal threshold decreases slowly since the RSU could afford to transmit more frequently without harvested energy when the channel is "*on*" state. Furthermore, as the channel availability rate increases, the optimal threshold i^* decreases gradually under the same reliable power limit because the RSU has more opportunities to send queuing data to passing vehicle within the coverage of RSU.

In Figure 9, we demonstrate the theoretical results to confirm the threshold-based structure of the optimal scheduling policy expressed by (30) and (31), where $\eta_e = \eta_d = 0.4$ and $\alpha = 0.7$. What is more, the transmission parameters bring to light a threshold-based structure imposed on the data queue length under reliable power constraints. From Figure 9(a), we can find that the optimal threshold is queue length "8," and transmission parameters are set as $g_7^* = 0.597$, $g_i^* = f_i^* = 1 (i \geq 8)$, which is the idea of threshold-



(a)



(b)

Figure 9: The threshold-based scheduling policy. (*a*) $\boldsymbol{P}_{\max} = 0.02$, (*b*) $\boldsymbol{P}_{\max} = 0.03$.

based policy. However, it is queue length "4" and $g_3^* = 0.649$, $g_i^* = f_i^* = 1 (i \geq 4)$in Figure 9(b). Though it is relatively easier to dispatch data packets when the communication channel is usable in Figure 9(b), it leads to a higher reliable power consumption in RSU at cost. In other words, the RSU makes good use of the power resource from reliable energy source by regulating the optimal threshold point for different power constraints.

6. Conclusions

In this paper, we investigated the delay-minimal scheduling problem with the supply of hybrid energy sources for RSU-to-vehicle downlink communication in CVIS. The RSU is powered by a limited capacity energy harvesting battery, coupled with a reliable energy constrained by maximum power consumption. By establishing probabilistic scheduling policy and modelling the two-dimensional Markov chain, we formulated the expressions of delay and reliable power consumption, and construct an optimization LP problem with the steady-state probability as variables. For solving the steady-state probability problem caused by the complexity of local equations, the analytical solution of the optimal transmission parameters can be obtained by threshold piecewise optimization. It has been proved that the optimal scheduling policy is threshold-based, that is, the use of reliable power resource for transmission should only occur when the data queue length exceeds optimal threshold while no harvested energy can be resorted if wireless channel is available. The optimal threshold is jointly decided by packet arrivals, downlink availability rate as well as the maximum reliable power constraint. At the end, simulation results

confirmed our theoretical analysis. It was shown that there always exists an optimal delay and reliable power limit trade-off in RSU and its decreasing rate depends on the harvested energy arrival rate and the battery capacity. Moreover, the downlink availability rate increases, the optimal threshold decreases gradually under the same power consumption limit.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

The authors would like to thank the supports of Scientific and Technological Key Project of "Revealing the List and Taking Command" in Heilongjiang Province (2021ZXJ02A02), the National Natural Science Foundation (61102105), and the Harbin Artificial Intelligence Association Key Project (ZKYLX004).

References

[1] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[2] X. Fan, C. Huang, J. Zhu, and B. Fu, "Replication-Based Data Dissemination in Connected Internet of Vehicles," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 2150524, 16 pages, 2019.

[3] L. Huang, H. Jiang, Z. Zhang, Z. Yan, and H. Guo, "Efficient data traffic forwarding for infrastructure-to-infrastructure communications in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 839–853, 2018.

[4] C. Wang, J. Li, Y. He, K. Xiao, and H. Zhang, "Destination prediction-based scheduling algorithms for message delivery in iovs," *IEEE Access*, vol. 8, pp. 14965–14976, 2020.

[5] M. M. Najm, M. Patra, and T. Venkatesh, "Cost-and-Delay aware dynamic resource allocation in federated vehicular clouds," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6159–6171, 2021.

[6] J. Li, C. Wang, D. Seo et al., "Deep Learning-Based Service Scheduling Mechanism for GreenRSUs in the IoVs," *Wireless Communications and Mobile Computing (WCMC)*, vol. 2021, article 7018486, pp. 1–15, 2021.

[7] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 1180–1189, 2012.

[8] M. Patra and C. S. R. Murthy, "Performance evaluation of joint placement and sleep scheduling of grid-connected solar powered road side units in vehicular networks," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 4, pp. 1197–1209, 2018.

[9] J. Liu, H. Dai, and W. Chen, "Delay optimal scheduling for energy harvesting based communications," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 452–466, 2015.

[10] B. Ko, K. Liu, S. H. Son, and K. -J. Park, "RSU-assisted adaptive scheduling for vehicle-to-vehicle data sharing in bidirectional road scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 977–989, 2021.

[11] K. Liu, J. K. Ng, V. C. Lee, S. H. Son, and I. Stojmenovic, "Cooperative Data Scheduling in Hybrid Vehicular Ad Hoc Networks: VANET as a Software Defined Network," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1759–1773, 2016.

[12] Q. Ibrahim, "Design, implementation and optimisation of an energy harvesting system for vehicular ad hoc networks' road side units," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 298–307, 2014.

[13] R. Atallah, M. Khabbaz, and C. Assi, "Energy harvesting in vehicular networks: a contemporary survey," *IEEE Wireless Communications*, vol. 23, no. 2, pp. 70–77, 2016.

[14] A. A. Hammad, G. H. Badawy, T. D. Todd, A. A. Sayegh, and D. Zhao, "Traffic scheduling for energy sustainable vehicular infrastructure," in *IEEE GLOBECOM*, pp. 1–6, Miami, FL, USA, 2010.

[15] W. S. Atoui, W. Ajib, and M. Boukadoum, "Offline and online scheduling algorithms for energy harvesting RSUs in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6370–6382, 2018.

[16] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4592–4601, 2017.

[17] L. Dai, T. Chen, Y. Zhai, and G. Wang, "Energy-Efficient Distributed Packet Scheduling Optimization Strategy in Cooperative Vehicle Infrastructure Systems," *Wireless Communications and Mobile Computing (WCMC)*, vol. 2021, article 6661623, pp. 1–11, 2021.

[18] L. Dai, Y. Zhai, and G. Wang, "Packet scheduling scheme for energy-delay tradeoff in self-powered roadside units," *Journal of Traffic and Transportation Engineering*, vol. 20, no. 2, pp. 161–171, 2020.

[19] A. Muhtar, B. Qazi, S. Bhattacharya, and J. Elmirghani, "Greening vehicular networks with standalone wind powered RSUs: a performance case study," in *Proceedings of the IEEE International Conference on Communications*, pp. 4437–4442, Budapest, Hungary, 2013.

[20] B. Collins and R. L. Cruz, "Transmission policies for time varying channels with average delay constraints," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, pp. 709–717, Monticello, IL, 1999.

[21] S. Zhang, A. Hafid, Z. Wang, H. Zhao, and S. Wang, "A cross-layer aware sensing-throughput tradeoff for Multi-Channel cognitive radio networks," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–5, Glasgow, UK, 2015.

[22] Q. Shi, L. Zhao, Y. Zhang, G. Zheng, F. R. Yu, and H. -H. Chen, "Energy-efficiency versus delay tradeoff in wireless networks virtualization," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 837–841, 2018.

[23] X. Liu, K. Hua, Z. Chen, A. S. Alghamdi, and M. N. Ali, "An Efficient Cross-layer Approach for Throughput-Maximal and Delay-Minimal Green Vehicular Networks," in *2018*

International Conference on Computing, Networking and Communications (ICNC), pp. 652–658, Maui, HI, USA, 2018.

[24] J. Zhao, Y. Chen, and Y. Gong, "Study of connectivity probability of vehicle-to-vehicle and vehicle-to-infrastructure communication systems," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pp. 1–4, Nanjing, China, 2016.

[25] L. Kleinrock, *Queueing Systems, Theory*, vol. 1, Wiley, Hoboken, NJ, USA, 1975.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5430622, 10 pages
https://doi.org/10.1155/2022/5430622

Research Article

Hybrid Model of Machine Learning Refractory Data Prediction Based on IoT Smart Cities

Xuewei Li, Kai Huang, and Lei Xu

School of Civil Engineering, Dalian Minzu University, Dalian, 116650 Liaoning, China

Correspondence should be addressed to Lei Xu; xulgb@163.com

Received 18 August 2021; Revised 5 November 2021; Accepted 2 December 2021; Published 23 February 2022

Academic Editor: Han Liu

With the advent of the digital age in recent years, the application of artificial intelligence in urban Internet of Things (IoT) systems has become increasingly important. The concept of smart cities has gradually formed, and smart firefighting under the smart city system has also become important. The method of machine learning is now applied in various fields, but seldom to the data prediction of smart firefighting. Various types of applications including data applications of machine learning algorithms in smart firefighting have yet to be explored. In this article, we propose using machine learning algorithms to predict building fire-resistance data, aiming to provide more theoretical and technical support for IoT smart cities. This article adopts the fire-resistance data of building beam components in a real fire environment, using three integrated machine learning algorithms, Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), and the grey wolf optimization algorithm to optimize. We improve the grey wolf algorithm and combine the grey wolf algorithm with the machine learning model. The algorithm constitutes three machine learning hybrid models: GWO-ET, GWO-AdaBoost, and GWO-GBM. Compared with traditional grid tuning, particle swarm optimization (PSO), and genetic algorithm (GA) optimization, the robustness and accuracy of the three optimization algorithms and the machine learning hybrid algorithm on the data set are compared and analyzed. Performance is measured through various performance comparisons and experimental result comparisons. For various building beam component data sets under real fires, the optimization and comparison show that the mean square error (MSE) of the proposed algorithm is extremely small. The results indicate that the GWO machine learning hybrid model is superior to other models and has a smaller prediction error.

1. Introduction

With the continuous development of modern smart cities, flammable building materials have increased the risk of fires that threaten lives and livelihoods, so the application of smart firefighting in urban Internet of Things (IoT) artificial intelligence has become more important. Machine learning technology is widely used in various fields, such as data mining, image processing, intelligent transportation, smart cities, medical health, intelligent prediction, and the IoT. Machine learning prediction is used in many fields, such as the prediction of stocks in the financial sector; the prediction of biological information, such as breast cancer classification and image recognition; the use of big data to mine social media data; and recommendation prediction in e-commerce. With the rapid development of artificial intelligence, the use of algorithms to predict related data has also become the top priority of smart fire protection and related fields under the urban IoT.

In [1, 2], the fire-resistance performance of most components in a fire is calculated by numerical, empirical, and computational analysis methods. In [3, 4], the fire-resistance limit is determined by the degree of damage deflection of the component. In this article, we use numerical analysis of fire-resistance data to conduct supervised learning to predict the fire-resistance limit of components. In previous studies, machine learning algorithms have been used to predict the fire-resistance data of building components. In [5], the artificial neural network (ANN) was used to predict the fire-resistance performance of concrete-clad steel composite columns. The predictive ability of the ANN was reasonable. The proposed ANN model was also better

than analytical equations in terms of prediction accuracy, indicating that the application of machine learning algorithms to the prediction of fire-resistance limit data for smart fire protection has a certain theoretical basis and higher accuracy. In [6], an integrated model was proposed to predict the bearing capacity of composite honeycomb steel beams in a fire, using gene expression programming (GEP), multiple linear regression (MLR), and principal component regression (PCR). The experimental results show that the combination of GEP and MLR was considered to predict the best model of CCSB carrying capacity. The above prediction was based on the algorithm and used the ambient temperature of the international standard temperature. In this article, the real fire environment temperature simulated by FDS is used, and the data are simulated and calculated using the ABAQUS software.

In [7], a pavement technical condition index attenuation prediction model based on LightGBM (Light Gradient Boost Machine) was proposed, which proved to be reliable and practical under the conditions of highway pavement. In [8], the ultimate gradient boosting (XGBoost) algorithm was used to predict the bearing capacity of fiber reinforced polymer (FRP) reinforced concrete (RC) columns. The prediction error was relatively low, but the network search method was used for optimization. We try to use the grey wolf optimizer (GWO) optimization method for Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), three algorithms used to predict refractory data. In [9], a decomposition method based on the long short-term memory (LSTM) network and GWO was proposed to develop a new wind speed prediction hybrid model. In [10], the Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) was proposed for use to accurately predict the output power of photovoltaic systems. In [11], it was proposed to apply grey wolf optimization and PSO to predict rheumatoid arthritis data, and the accuracy rate improved greatly. In [12], a grey wolf optimizer composed of a new hybrid algorithm was introduced to evaluate the parameters of SVM; it was used for power system load forecasting and was compared with the prediction performance of PSO-SVM and GA-SVM, which proved the performance of the provided method. In [13], a GWO-SVM method was proposed to establish a CO detection correction model in the process of coal combustion loss in coal mines to correct the CO concentration measurement. The results show that the GWO-SVM model had higher accuracy and stability than other models. In the latest development of GWO, in [14], GWO introduced a search strategy called Representation-Based Hunting (RH) to form the R-GWO algorithm, which is a representation-based grey wolf optimizer for solving engineering problems. In [15], Gaussian walk and Lévy flight were used to improve the exploration and development capabilities of GGWO and predict the COVID-19 pandemic in the United States. Here, we use GWO optimization for Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM) hyperparameter tuning of three machine learning algorithms and add a set of random numbers to the cycle into the settings and strategies to transform the data form, make it more suitable for the model, reach the search faster, and compare and analyze its accuracy with PSO algorithm and GA optimization in refractory data prediction.

In recent years, the IoT has emerged in various fields and become the main core of Industry 4.0. In [16], an effective technique called SIoMT (Swarm Intelligence optimization technique for the IoMT) is proposed in this paper for periodically discovering, clustering, analyzing, and managing useful data about potential patients. Rarely, only ANNs, linear regression, PCA, and other algorithms are not used in integrated algorithms, nor are they used in optimization algorithms. This article will use three integrated machine learning algorithms, Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), and will apply the grey wolf optimizer to the machine learning prediction model of refractory data to form a machine learning hybrid model. This represents new exploration in the field of smart cities and smart fire protection as well as a new test of whether machine learning algorithms can be used in the prediction of fire-resistance data.

The main research contents of this article include the following:

(1) After obtaining the fire-resistance limit data set through the ABAQUS software, we perform data preprocessing on it and select three machine learning integration algorithms, Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), to make a prediction on the data set through algorithm evaluation

(2) The grey wolf optimization algorithm is improved to form a machine learning hybrid model and optimize the training of the data set

(3) The results and performance of PSO, GA, and grid tuning are compared, and their effects and accuracy are analyzed

The main contributions and novelties of this article are as follows.

(1) The refractory data at ambient temperature based on an FDS real fire simulation is used for machine learning modeling, including Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM)

(2) The grey wolf optimizer is used for algorithm prediction and to improve the grey wolf optimization algorithm; a set of random numbers is used to cycle into the judgment of the optimal solution, improve the data input form and iterative method, and make it suitable for the hyperparameters of the selected algorithm

(3) The grey wolf optimization algorithm and machine learning model are combined, and the optimal parameters obtained by the grey wolf optimization algorithm are substituted into the machine learning model to obtain three machine learning hybrid

models: GWO-ET, GWO-AdaBoost, and GWO-GBM

(4) Comparing the performance of PSO, GA, and traditional optimization on different refractory data sets, it is proved that the grey wolf optimizer optimizes the prediction effect, greatly reduces its errors, and improves the accuracy

2. Research Methods Using Machine Learning Algorithms to Predict Refractory Data

In this study, we first select three ensemble algorithms for evaluation, Extreme random Tree (ET), AdaBoost, and Gradient Boosting Machine (GBM), and use the grey wolf optimizer to optimize the prediction model.

2.1. Integrated Algorithm

(1) ET algorithm

The ET is composed of many decision trees, including root nodes, internal nodes, and leaf nodes, representing a mapping relationship between object attributes and object values. The leaf node corresponds to the decision result, and the route from the root node to one of the leaf nodes is the entire decision process of the class corresponding to the current leaf node.

The decision tree obtains the entropy by using the algorithm ID3, C4.5.

(a) Introduction to ID3 algorithm

The ID3 algorithm calculates the information gain each time it splits and then selects the largest information gain for standard splitting. D is the sample set before the current decision split, K is the sample category, and the information entropy is defined as follows:

$$\mathrm{Ent}(D) = -\sum_{k=1}^{K} p_k \log_2 p_k. \tag{1}$$

By calculating the information gain, the sample is split according to the attribute set to output a subset of each attribute sample, and then, the information entropy of a single subset is calculated according to the category of each sample subset:

$$\mathrm{Ent}(D_V) = -\sum_{k=1}^{K} p_k \log_2 p_k. \tag{2}$$

Afterward, the weight of each sample subset is assigned, and the calculation is performed according to the ratio of the number of samples at each branch node to the total number of samples. The following is the calculation of information gain:

$$\mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{D_V}{D} \mathrm{Ent}(D_V). \tag{3}$$

(b) Introduction to C4.5 algorithm

The C4.5 algorithm divides the optimal attribute using the gain rate. The formula for the gain rate is as follows, where the larger a is, the larger V and IV are.

$$\begin{aligned} \mathrm{Gain}_{\mathrm{ratio}(D,a)} &= \frac{\mathrm{Gain}(D, a)}{\mathrm{IV}(a)}, \\ \mathrm{IV}(a) &= -\sum_{v=1}^{V} \frac{|D_V|}{|D|} \log_2 \frac{|D_V|}{|D|}. \end{aligned} \tag{4}$$

(c) Gini index

The purity of the data set can be judged not only by information entropy but also by the Gini index. The following is the formula for the Gini index:

$$\mathrm{Gini}(D) = \sum_{k=1}^{K} \sum_{k^+ \neq k^-} p_k^+ p_k^- = 1 - \sum_{k=1}^{K} p_k^2. \tag{5}$$

$\mathrm{Gini}(D)$ reflects that when two samples are randomly selected, the probability of a different category is different. At the same time, the smaller the $\mathrm{Gini}(D)$, the higher the purity of the sample. Therefore, the formula for the Gini index of attribute a is as follows. Use the Gini index to calculate the splitting of the node, which is the process of calculating $a_v = \mathrm{argmin}\mathrm{Gini\_index}(D, a)$.

$$\mathrm{Gini}_{\mathrm{index}(D,a)} = \sum_{v=1}^{V} \frac{|D_V|}{|D|} \mathrm{Gini}(D_V). \tag{6}$$

(2) AdaBoost algorithm

Input training data, and $T = \{\{x_1, y_1\}, \{x_2, y_2\}, \cdots, \{x_N, y_N\}\}$. Among them, $x_i \in X, y_i \in Y = -1, 1$, and the number of iterations is M. The weight distribution of the initial training sample is $D_1 = (\omega_{1,1}, \omega_{1,2}, \cdots, \omega_{1,i}), \omega_{1,i} = 1/N, i = 1, 2, \cdots, N$. For $m = 1, 2, \cdots, M$,

(a) Use the training data set with the weight distribution D_m to learn and obtain the weak classifier $C_m(x)$

(b) Calculate the classification error rate of $C_m(x)$ on the training data set

$$e_m = \sum_{i-1}^{N} \omega_{m,i} I(G_m(x_i) \neq y_i). \tag{7}$$

(c) Calculate the weight of $G_m(x)$ in the strong classifier

$$\alpha_m = \frac{1}{2} \log \frac{1 - e_m}{e_m}. \tag{8}$$

(d) Update the weight distribution of the training data set, where z_m is the normalization factor, so that the sum of the probability distribution of the sample is 1

$$\begin{aligned} \omega_{m+1,i} &= \frac{\omega_{m,i}}{z_m} \exp\left(-\alpha_m y_i G_m(x_i)\right), i = 1, 2, \cdots, 10, \\ z_m &= \sum_{i=1}^{N} \omega_{m,i} \exp\left(-\alpha_m y_i G_m(x_i)\right). \end{aligned} \tag{9}$$

(e) Obtain the final classifier

$$F(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_m G_m(x)\right). \tag{10}$$

(3) GBM algorithm

GBM is based on the CART regression tree and gradient boosting tree.

Initialize the weak learner. For $m = 1, 2, \cdots, M$,

(a) For each sample $i = 1, 2, \cdots, N$, calculate the negative gradient, that is, the residual

$$\gamma_{im} = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}. \tag{11}$$

(b) Use the residual obtained in the previous step as the new true value of the sample, and use the data (x_i, γ_{im}), $i = 1, 2, \cdots, N$ as the training data of the next tree to obtain a new regression tree, $f_m(x)$. Its corresponding leaf node area is R_{jm}, $j = 1, 2, \cdots, J$, where J is the number of leaf nodes of the regression tree t

(c) Calculate the best fit value for the leaf area $j = 1, 2, \cdots, J$

$$\gamma_{jm} = \text{argmin} \sum_{x_i \epsilon R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma). \tag{12}$$

(d) Calculate the best fit value for the leaf area $j = 1, 2, \cdots, J$

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J} \gamma_{jm} I\left(x \epsilon R_{jm}\right). \tag{13}$$

Obtain the final learner

$$f(x) = f_M(x) = f_0(x) + \sum_{m=1}^{M} \sum_{j=1}^{J} \gamma_{jm} I\left(x \epsilon R_{jm}\right). \tag{14}$$

2.2. Grey Wolf Optimizer. The grey wolf optimizer (GWO) is a population intelligence optimization algorithm proposed by Griffith University scholar Mirjalili and others in 2014. The algorithm was inspired by the hunting activities of grey wolves and then developed into an optimized search method. It has the characteristics of strong convergence performance, few parameters, and easy implementation. In recent years, it has received extensive attention from scholars and has been successfully applied to workshop scheduling, parameter optimization, image classification, and other fields. Usually, GWO is used for SVM optimization, but in this article, the characteristics of the algorithm are used to optimize search so as to optimize the hyperparameters of the integrated algorithm.

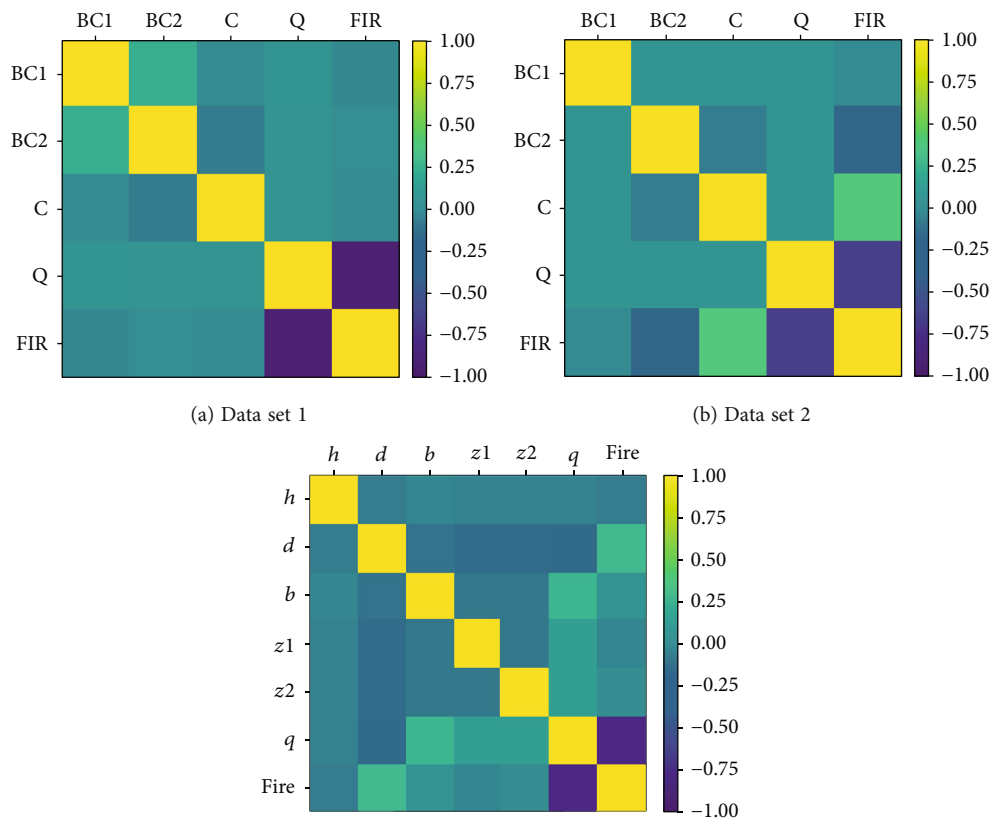
Algorithm principle

There are three wolves, α, β, and γ, in a wolf pack. α is the wolf king, and β and γ are ranked second and third, respectively. Both β and γ listen to α, and γ listens to β. The three wolves guide the other wolves in search of prey. The process of wolves looking for prey represents the process of the GWO algorithm finding the optimal solution. The GWO optimization process also includes social hierarchical stratification, tracking, encircling and attacking prey, and searching for prey. However, its core behavior is only hunting, that is, finding the optimal solution. During each iteration, the best three wolves, α, β, and γ, in the population will be retained, and the positions of other search agents will be updated according to their position information.

(1) Surround the prey

The grey wolf will gradually approach and surround its prey when hunting. Its mathematical model is as follows [17]:

$$\begin{aligned} D &= C_0 X_p(t) - X(t), \\ X(t+1) &= X_p(t) - A_0 D, \\ A &= 2ar_1 - a, \\ C &= 2r_2. \end{aligned} \tag{15}$$



(a) Data set 1

(b) Data set 2

(c) Data set 3

Figure 1: Correlation matrix diagram description.

Here, t is the current iteration number, A and C are the coordination coefficient vectors, X_p is the position vector of Lei Wu, $X(t)$ is the current position vector of the grey wolf, and r_1 and r_2 are random vectors between [0,1].

(2) Hunting behavior

Grey wolves have the ability to identify the position of potential prey (optimal solution), that is, to exhibit hunting behavior. The mathematical model is as follows:

$$D_\alpha = C_1 X_\alpha - X, D_\beta = C_2 X_\beta - X, D_\delta = C_3 X_\delta - X, \quad (16)$$

$$X_1 = X_\alpha - A_1 D_\alpha, X_2 = X_\beta - A_2 D_\beta, X_3 = X_\delta - A_3 D_\delta, \quad (17)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3}. \quad (18)$$

Here, X_α, X_β, and X_δ represent the position vector of α, β, and δ, respectively, in the current population; D_α, D_β, and D_δ represent the distance between the current candidate wolf pack and the best three wolves, respectively; when $|A| > 1$, grey wolves will scatter in various areas to look for prey. When $|A| < 1$, grey wolves will focus on one or some areas to look for prey [17].

```
1: Read CSV data
2: Build tag
3: Perform data preprocessing
4: Separate data sets
5: Normalize data
6: Call GWO algorithm and objective function
```

Algorithm 1: Model 1 steps: GWO-ET model.

(3) Attack the prey

The attacking prey will also decrease according to the decrease of the value of a, the random vector on $A \epsilon [-a, a]$, where a decreases linearly in the iterative process. When $A \epsilon [-1, 1]$ is in the interval, the next time position of the search agent can be anywhere between the current grey wolf and the prey.

(4) Looking for prey

Grey wolves rely on α, β, and δ to find their prey. They start to search for prey location information in a scattered manner and then gather to attack the prey and complete this optimization through iteration [17].

(5) Improvement of GWO algorithm

```
Input: objective function, Parameter adjustment range of super parameter, Xtrain, Ytrain, Xtest, Ytest, PopSize, Niter
Output: The global optimum (gbest)
Procedure GWO
Initialize input parameters, alpha, beta, delta, All individual samples
# Iterative optimization, # main loop of the algorithm
For l=1: Niter
  For i=1:PopSize
    Call objective function, Generates a set of random numbers in the hyperparametric range
      Update the optimal location of Alpha, Beta, and Delta
      Update X1, X2, X3 by equation (18)
      I=i+1
    End for
    l=l+1
End for
Return the global optimum (gbest)
End Procedure
```

Algorithm 2: Optimized GWO.

```
Input: Over parameter adjustment range, Xtrain, Ytrain, Xtest, Ytest
Output: MSE
Procedure objective function
For i in A set of random numbers # Cyclic substitution
i data conversion # int, float or other more complex forms
Substitute in ExtraTreesRegressor
Train
Calculate MSE by formula (19)
Return MSE
End Procedure
```

Algorithm 3: Objective function setting.

Table 1: Data set 1/2 parameter selection range.

Parameter	Tuning range
n_estimators	(1, 150)
max_depth	(20, 30)
max_features	["auto," "sqrt"]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
Bootstrap	[true, false]

Table 2: Data set 3 parameter selection range.

Parameter	Tuning range
n_estimators	(1, 50)
max_depth	(1, 21)
Subsample	(0.1,0.9)
Learning_rate	(0.1, 1.1)
min_samples_split	(2, 11)
min_samples_leaf	(1, 11)

Table 3: MSE changes before and after tuning.

Model	MSE changes before	RandomizedSearchCV MSE	GWO optimized MSE
Data set 1	0.0022	0.0021	0.0015
Data set 2	0.093	0.0447	0.0115
Data set 3	0.2300	0.2013	0.0420

Using the GWO algorithm to scatter the grey wolf to find the prey and then focus on the area to find the prey, the optimal solution is that it gradually approaches the optimal solution target, continuously narrows the search range, and finally finds the optimal solution through iteration. We narrow the scope to find the prey by improving the GWO algorithm, apply it to the form of the machine learning algorithm to adjust the hyperparameter, and add a set of random numbers in each round of iterations and looping to determine the ET, AdaBoost, and GBM super parameter. At the same time, because the GWO is set according to dimensions, its data form is converted into a form suitable for algorithm hyperparameters. Then, the optimal solution is further

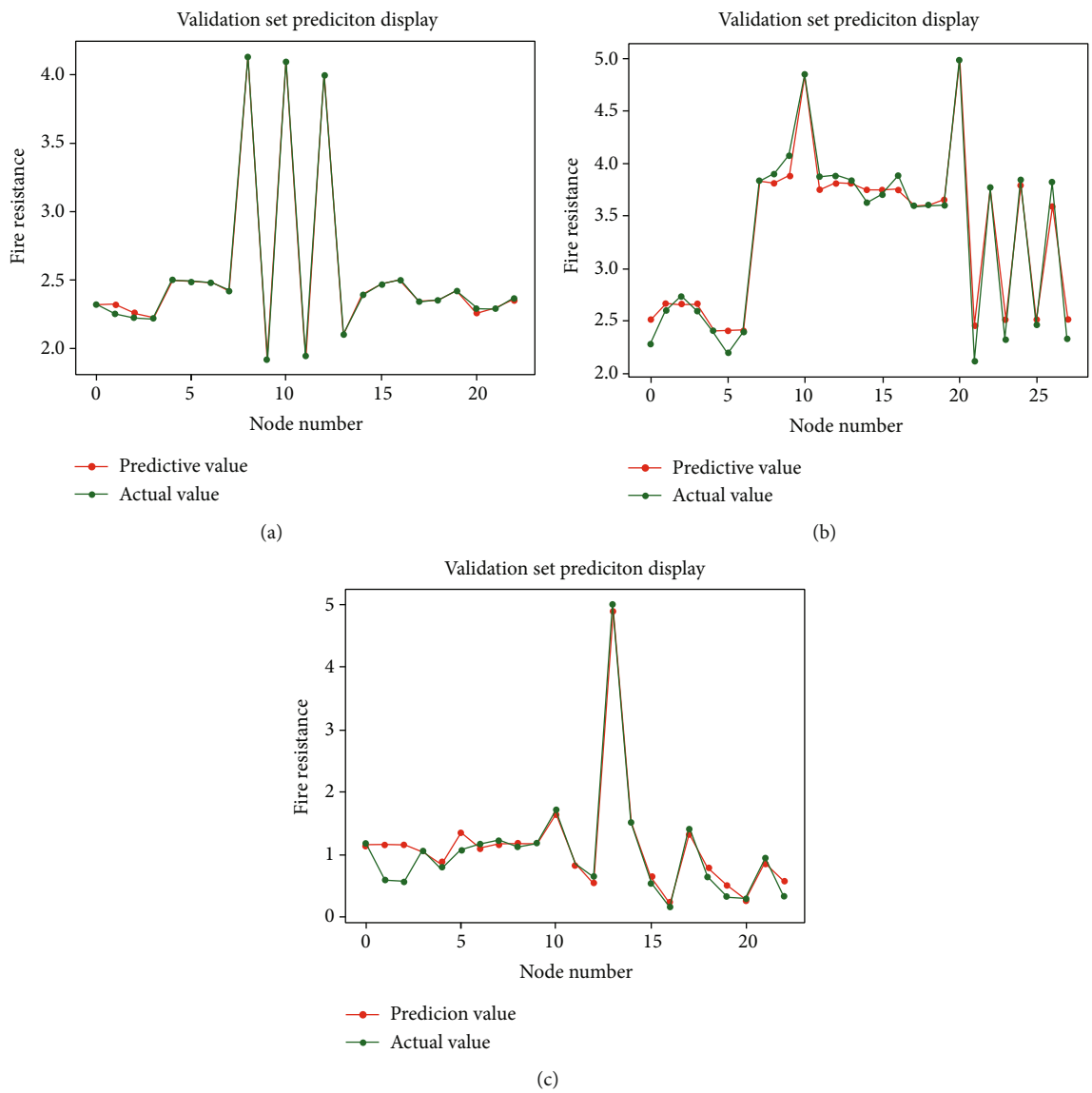


Figure 2: Data set fitting effect.

searched, and the optimal parameter is found with the smallest MSE.

Unlike the previous GWO algorithm,

1) When looking for a local optimal solution, we use a set of random numbers to cycle into and judge the optimal solution and perform related data format conversions to apply the hyperparameters of the algorithm

2) We combine the improved GWO algorithm and the machine learning model to obtain the GWO optimized machine learning prediction model

2.3. Evaluation Index. We use mean squared error (MSE) as the evaluation index. The formula of MSE is as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{m}\omega_i(y_i - y_i\wedge)^2, \tag{19}$$

where n is the number of samples, y_i is the real predicted data, and $\widehat{y_i}$ is the fitted predicted data.

3. Refractory Data Prediction Model Based on Machine Learning

3.1. Data Collection. In this article, the numerical simulation software ABAQUS is used to establish the model to obtain the data. For the three beams, finite element models with different cross-sections (BC1 and BC2), different load ratios (Q), and different protective layer thicknesses (C) are established. For steel beams, different waist heights (h) are established. The finite element model of waist thickness includes the (b), leg width (b), upper flange thickness (z1), and lower flange thickness (z2). The steel is Q345 steel, the strength grade of the concrete is C35, and the absolute zero is set to what. The fire condition is that the beam is exposed to fire on three sides, and the back of the upper flange of the steel beam and below the upper flange are all exposed to fire.

The heat radiation coefficient of the fire surface is 0.5, the radiation coefficient distribution is consistent, and the surface heat exchange film has a heat dissipation coefficient of 25. We import the real FDS simulation and set the corresponding analysis step and time for the fire environment temperature.

Then, we import the result database file obtained from the temperature field model into the predefined field of the mechanical model, set different loads according to the established model, apply uniformly distributed loads on the upper surface of the beam through pressure, set the boundary conditions and constraints, and perform finite element calculations on each model. We obtain various beam deformations and beam midspan deflection. From the mechanical model of each beam, we obtain the change of beam midspan deflection with time during the fire. According to "Standard for Fire Test of Building Components" (GB/T9978-2008) [18], when the maximum deflection of the beam reaches the deflection, and the deformation exceeds $L/30$ (mm) (L is the calculated span of the beam and slab), the beam can be judged to have reached the fire-resistance limit. The data set required by the fire-resistance limit prediction model is thus obtained.

3.2. Data Processing. Through the previous finite element modeling, the different parameters of the beam and the corresponding fire-resistance limit are obtained, which constitutes the data set required for this research. The training data set is divided by 80%, the evaluation data set is divided by 20%, and the data set is divided by what. We perform unified processing of the format and unit. Figure 1 shows a description of the correlation matrix diagrams of the three data sets.

Standardization of data is normalization. Data standardization is based on the columns of the feature matrix. The attributes of the data are converted to obey the normal distribution, and the mean and variance are normalized; that is, the mean is 0, and the standard deviation is 1. Its transformation function is as follows:

$$x^* = \frac{x - \mu}{\sigma}. \tag{20}$$

Among them, μ is the mean of all sample data, and σ is the standard deviation of all sample data.

3.3. Model Steps and Optimization. After preprocessing the data set, we begin to build a grey wolf optimized machine learning hybrid model. Data set 1 uses ET, data set 2 uses AdaBoost, and data set 3 uses GBM for training. To make the machine learning model better for learning and hyperparameter tuning, based on the principal formula of Section 2.2, we perform the setting update of the objective function and the data form conversion and compose it into every 10 ones. The random numbers of the group are cyclically substituted into the machine learning model for learning, and then, the grey wolf optimization population and iterative optimization are performed. We form the grey wolf before encircling and hunting, and it loops through the random array first and then encloses the optimal value of the objective function to increase the speed and probability of encircling the prey. The specific steps of this model are shown in Algorithm 1, model 1. Taking the GWO-ET model as an example, the pseudo code of the grey wolf optimization algorithm is shown in Algorithm 2. The basic grey wolf optimization algorithm of this code refers to [19–22]. The setting of the objective function is shown in Algorithm 3. PopSize is the population size, and Niter is the number of iterations.

Table 4: GWO algorithm, PSO algorithm, and GA optimization comparison.

MSE	Data set 1	Data set 2	Data set 3
GWO optimized MSE	0.0015	0.0115	0.0420
PSO optimized MSE	0.0015	0.0149	0.0525
GA optimized MSE	0.0017	0.0525	0.0643

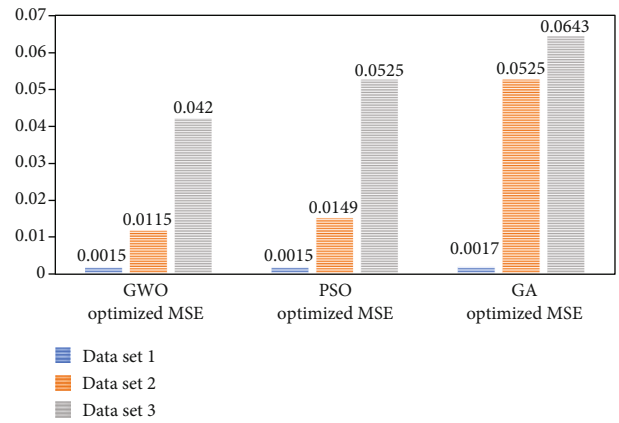


Figure 3: GWO algorithm, PSO algorithm, and GA optimization MSE comparison diagram.

4. Model Verification and Comparison

4.1. Comparison of Grid Tuning. In this section, we use RandomizedSearchCV grid tuning and grey wolf optimization parameters to compare the models. Data set 1 uses ET, data set 2 uses AdaBoost, and data set 3 uses GBM. For training, the range of grid adjustment parameters is shown in Tables 1 and 2. We set the objective function of the GWO algorithm; the population size is 20, the number of iterations is 100, and the algorithms are compared. Before tuning, the comparison results of grid tuning and GWO optimization MSE are as shown in Table 3.

Figure 2 shows the fitting effect after GWO optimization of three data sets. The red line represents the predicted value, and the green line represents the true value. As shown, the models optimized by the grey wolf optimizer greatly reduce the MSE, and the fitting effect is also better.

4.2. Comparison with PSO Algorithm and GA. In this section, we compare the GWO machine learning hybrid model with the PSO and the GA machine learning hybrid model. The population size is set to 20, and the number of iterations

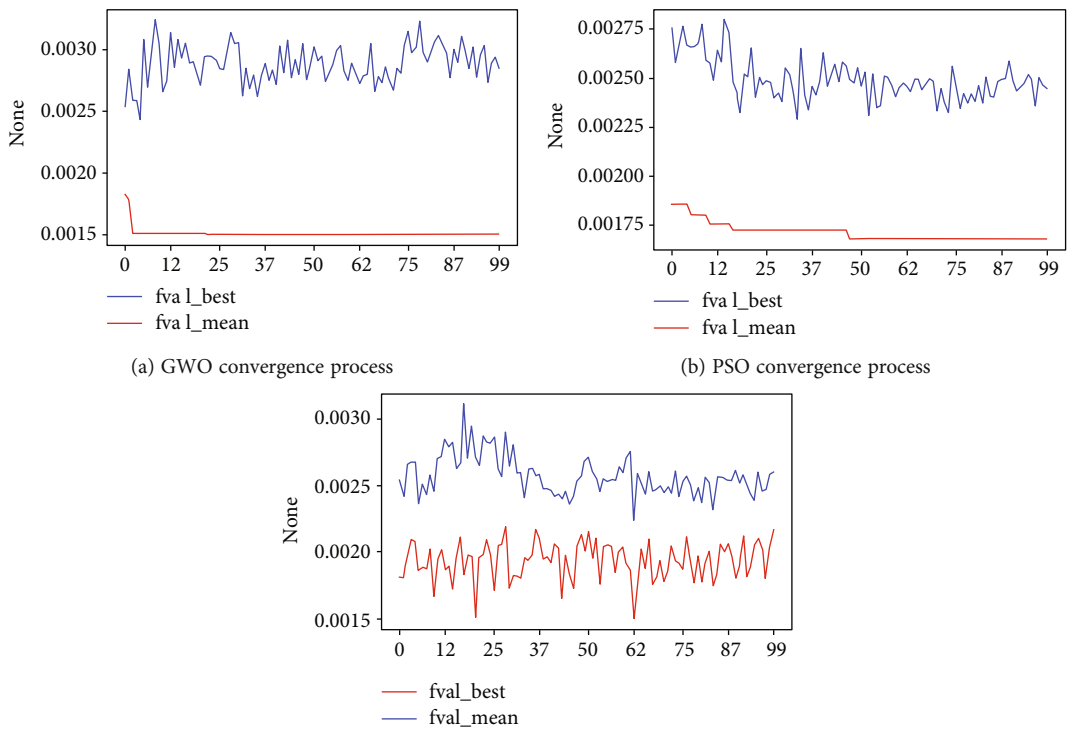


Figure 4: Convergence process comparison.

is 100, as shown in Table 4 and Figure 3. In the comparison of the optimization algorithm MSE of GWO, PSO, and GA, the GWO optimization MSE is the smallest, followed by the PSO model.

4.3. Convergence Process Comparison. After GWO quickly finds the optimal solution by narrowing the scope, iterative optimization is faster and more stable, and the convergence process is better. PSO simulates the birds in a flock of birds through a flock of particles, initializes it as a flock of random particles, and then iteratively optimizes. GA is suitable for multivariable, multiobjective complex optimization, but it is also prone to premature convergence. Figure 4 shows the comparison of the convergence process of GWO, PSO, and GA. The red line fval_best represents the optimized target value, and the blue line fval_mean represents the mean value. GWO narrows down the range after quickly finding a smaller value to finally enclose the best value. The convergence process of PSO and GWO is similar, while the convergence process of GA is relatively unstable and is not suitable for refractory data sets, more complex nonlinear function relations, or multiobjective optimization.

4.4. Calculation Performance Comparison. Although the convergence process of GWO is relatively stable, it consumes more calculation time owing to the number of iterations. However, although it does not require many iterations to find the optimal parameters, it is CPU-intensive. PSO and GA save more time. Under the same population size and iteration number of the GWO algorithm, PSO algorithm, and GA, the calculation time of PSO and GA is half that of GWO. Therefore, the GWO optimization algorithm saves calculation time and has faster convergence speed and more accurate optimization compared with grid tuning. Compared with PSO, the GA has a longer calculation time, but the optimization is more accurate, and the convergence is more stable.

5. Conclusion

This article mainly uses machine learning algorithms to predict three fire-resistant data sets. The fire-resistant data obtained under the real fire environment temperature based on FDS simulation is preprocessed, and ET, AdaBoost, and Gradient Boosting Machine (GBM) are combined with the grey wolf optimizer to form a hybrid prediction model and improve the grey wolf optimization algorithm. It uses a set of random numbers to cycle into and judge the optimal solution and performs related data format conversions to apply the hyperparameters of the hybrid model algorithm. The improved optimization algorithm hybrid model proposed in this article can greatly improve the accuracy of predicting the refractory data of the IoT. After comparing with the grid parameter adjustment, optimization algorithm PSO, and GA hybrid model, the GWO machine learning hybrid model is used in the prediction of refractory data and has a higher accuracy rate. The shortcomings of this article are that the research model is not perfect, more experimental data are needed, the calculation time of GWO calculation is longer than other optimization algorithms, and it needs to be optimized. In the future, artificial intelligence technology based on IoT smart cities will become more widespread, and the

application of machine learning algorithms for smart fire protection, big data prediction, and other smart cities will increase. These applications will require us to conduct more in-depth research.

Data Availability

The data underlying the results presented in the study are available within the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

The research reported in the paper is part of the Project 52178461 supported by the National Natural Science Foundation of China and the Project LJYT201908 supported by the Foundation of Department of Education of Liaoning Province. The financial support is highly appreciated.

References

[1] S. F. El-Fitiany and M. A. Youssef, "Assessing the flexural and axial behaviour of reinforced concrete members at elevated temperatures using sectional analysis," *Fire Safety Journal*, vol. 44, pp. 691–703, 2009.

[2] M. B. Dwaikat and V. K. R. Kodur, "Response of restrained concrete beams under design fire exposure," *Journal of Structural Engineering*, vol. 135, pp. 1408–1417, 2009.

[3] M. B. Dwaikat and V. K. R. Kodur, "A numerical approach for modeling the fire induced restraint effects in reinforced concrete beams," *Fire Safety Journal*, vol. 43, pp. 291–307, 2008.

[4] V. K. R. Kodur and M. Dwaikat, "Flexural response of reinforced concrete beams exposed to fire," *Structural Concrete*, vol. 9, pp. 45–54, 2008.

[5] J. Y. Shan Li, R. Liew, and M.-X. Xiong, "Prediction of fire resistance of concrete encased steel composite columns using artificial neural network," *Engineering Structures*, vol. 245, 2021.

[6] B. Karami, A. Shishegaran, H. Taghavizade, and T. Rabczuk, "Presenting innovative ensemble model for prediction of the load carrying capacity of composite castellated steel beam under fire," *Structures*, vol. 33, pp. 4031–4052, 2021.

[7] L. Pei, T. Yu, L. Xu, W. Li, Y. Han, and X. Li, "Prediction of Decay of Pavement Quality or Performance Index Based on Light Gradient Boost Machine," *Advances in Intelligent Automation and Soft Computing*, vol. 80, pp. 1173–1179, 2022.

[8] A. S. Bakouregui, H. M. Mohamed, A. Yahia, and B. Benmokrane, "Explainable extreme gradient boosting tree-based prediction of load-carrying capacity of FRP-RC columns," *Engineering Structures*, vol. 245, 2021.

[9] A. Altan, S. Karasu, and E. Zio, "A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer," *Applied Soft Computing*, vol. 100, p. 106996, 2021.

[10] M. Abdel-Nasser, "Accurate photovoltaic power forecasting models using deep LSTM-RNN," *Neural Computing and Applications*, vol. 31, 2019.

[11] S. Sundaramurthy and P. Jayavel, "A hybrid grey wolf optimization and particle swarm optimization with C4.5 approach for prediction of rheumatoid arthritis," *Applied Soft Computing*, vol. 94, 2020.

[12] M. Barman, D. Choudhury, and N. Behari, "A similarity based hybrid GWO-SVM method of power system load forecasting for regional special event days in anomalous load situations in Assam, India," *Sustainable Cities and Society*, vol. 61, 2020.

[13] J. Deng, W. L. Chen, C. Liang et al., "Correction model for CO detection in the coal combustion loss process in mines based on GWO-SVM," *Journal of Loss Prevention in the Process Industries*, vol. 71, 2021.

[14] M. Banaie-Dezfouli, M. H. Nadimi-Shahraki, and Z. Beheshti, "R-GWO: representative-based grey wolf optimizer for solving engineering problems," *Applied Soft Computing*, vol. 106, 2021.

[15] S. Khalilpourazari, H. H. Doulabi, A. Ö. Çiftçioğlu, and G.-W. Weber, "Gradient-based grey wolf optimizer with Gaussian walk: application in modelling and prediction of the COVID-19 pandemic," *Expert Systems with Applications*, vol. 177, 2021.

[16] E. El-shafeiy, "A clustering based Swarm Intelligence optimization technique for the Internet of Medical Things," *Expert Systems with Applications*, vol. 173, 2021.

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[18] GB/T9978-2008, *Fire resistance test methods of building components*, National Standard of the People's Republic of China, 2008.

[19] Genlovy-Hoo, "utils_hoo[Source code]," 2021, https://github.com/Genlovy-Hoo/utils_hoo/tree/master/utils_hoo.

[20] H. Faris, I. Aljarah, S. Mirjalili, P. Castillo, and J. Merelo, "EvoloPy: an open-source nature-inspired optimization framework in Python," in *Proceedings of the 8th International Joint Conference on Computational Intelligenc*, vol. 1, p. 171, 2016.

[21] R. Qaddoura, H. Faris, I. Aljarah, and P. A. Castillo, "EvoCluster: an open-source nature-inspired optimization clustering framework in Python," in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pp. 20–36, Springer, Cham, 2020.

[22] R. A. Khurma, I. Aljarah, A. Sharieh, and S. Mirjalili, "Evolopy-FS: An Open-Source Nature-Inspired Optimization Framework in Python for Feature Selection," in *Evolutionary Machine Learning Techniques*, pp. 131–173, Springer, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6196810, 12 pages
https://doi.org/10.1155/2022/6196810

Research Article

Foreground Information-Aware Image Superresolution Reconstruction for Image Processing IoT Systems in Smart City

Yanfen Cheng,[1] **Chenhao Li**,[1] **Xun Shao**,[2] **and Fan He**[1]

[1]*School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430063, China*
[2]*School of Regional Innovation and Social Design Engineering, Kitami Institute of Technology, Kitami, Japan*

Correspondence should be addressed to Xun Shao; x-shao@ieee.org and Fan He; hefan@whut.edu.cn

Received 15 November 2021; Revised 15 December 2021; Accepted 21 December 2021; Published 18 January 2022

Academic Editor: Han Liu

In recent years, with the rise of Internet of Things (IoT), a majority of smart technologies, such as autonomous vehicles, smart healthcare, and urban surveillance, require a huge number of images of high quality and resolution. Currently, image superresolution reconstruction technologies are widely used for obtaining high quality images. Unfortunately, the existing methods generally focus on the whole image without highlighting foreground information and lack visual focus. Also, they have low utilization of shallow features and numerous training parameters. In this paper, we propose a feature extraction module that focuses on foreground information: the parallel attention module (PAM). PAM computes channel and spatial attention in parallel, inputs the obtained attention values into a cascaded gated network, and dynamically adjusts the weights of both using nonuniform joint loss to focus on image foreground information and detail features to improve the reconstructed image's foreground sharpness. To further improve the performance, we propose to connect multiple PAM modules in series with skip connections and call it PAMNet. PAMNet can better leverage the shallow residual features, and the reconstructed images are closer to ground truth. Thereby, the applications in the urban image processing IoT systems can obtain high-resolution images more quickly and precisely. The comprehensive experimental results show that PAMNet performs better than the state-of-the-art technologies.

1. Introduction

With the rapid development of artificial intelligence (AI) [1–5] and 5G [6, 7], many emerging technologies, such as Internet of Things [8–17], blockchain [18–21] autonomous vehicles [22–24], smart healthcare [25–31], and urban surveillance, that meet people's aspirations for a better life, are developing very fast. In these smart technologies, image processing IoT applications such as autonomous vehicles, smart healthcare, and urban surveillance are playing important roles in the upcoming smart society. Figure 1 shows the application of urban IoT systems.

However, due to the heterogenous properties of the smart camera devices and complicated network environment, the smart applications deployed in the remote cloud can often only obtain low-resolution images, which largely limit the usage of the smart applications. For example, (1) high-speed cars need to recognize the contents of road signs as early as possible, but due to the long shooting distance and small road signs, it is necessary to convert the captured low-resolution images into high-resolution images with the help of image superresolution methods. (2) In suburban community hospitals, we need to superresolve the transmitted low-resolution images to improve the accuracy of doctors' remote diagnosis due to the poor quality of the captured equipment. (3) Police often tracks the trajectory of suspects through urban surveillance systems, and after image superresolution reconstruction, get a clearer picture of the suspect's appearance and characteristics to speed up the process of crime solving. In summary, image superresolution reconstruction has broad applications in urban IoT systems.

Single image superresolution (SISR) is the task of generating high-resolution images using a single low-resolution image [32]. SISR algorithms are divided into three main categories: interpolation-based methods [33], reconstruction-based methods [34, 35], and learning-based methods [36,

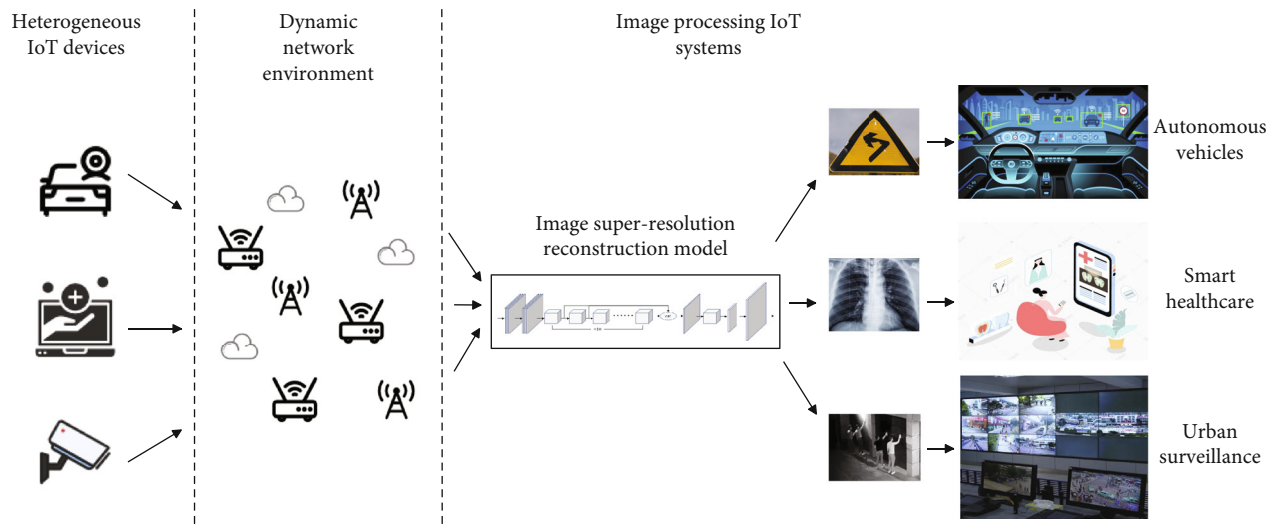


Figure 1: Image superresolution reconstruction for image processing IoT systems in Smart City.

37]. Learning-based methods are one of the most widely used methods at present. In particular, with the development of deep learning and generative adversarial networks, image superresolution has made great progress.

Dong et al. [38] proposed an SRCNN method that has realized end-to-end super resolution image reconstruction and better performance compared with other previous methods. However, the simple network structure limits its ability to extract features, and the MSE loss used by SRCNN stresses improving the image objective index, ignoring the subjective effect of the image. The detailed features of the blurred reconstructed images are VDSRs—depth models based on residual learning, which were proposed by Kim et al. [39]—that improve the model performance by introducing a residual structure, but there are problems such as large number of training parameters and unclear background of reconstructed images. EDSR proposed by Lim et al. [40] removes the BN layer and superimposes more layers to improve the reconstructed image quality by reducing the memory consumption of the BN layer. However, since L1 loss is used for training, the objective index of the reconstructed image is low.

Thanks to the generative adversarial networks proposed by Goodfellow et al. [41], the image superresolution task has opened a new chapter dominated by generative adversarial structures. SRGAN proposed by Ledig et al. [42] uses generative adversarial networks for image superresolution while using perceptual loss and adversarial loss to improve the realism of the reconstructed image, which makes the reconstructed image and the ground truth closer in semantics and style. However, the reconstructed image loses some high-frequency information due to the mere use of MSE loss to train the generator. ESRGAN proposed by Wang et al. [43] removes the BN layer based on SRGAN and introduces dense connections to avoid artifacts. VGG features before activation are used to improve perceptual loss and to make the edges and details of the reconstructed images clearer. The idea of relativistic GAN [44] is applied for reference to judge the probability that real images are more realistic than generated images in the discriminator, greatly enhancing the subjective effect of reconstructed images. Nevertheless, ESRGAN has many parameters and a long training time. RFB-ESRGAN proposed by Shang et al. [45] introduces a multi-scale receptive field module to extract edge features of images and alternately uses nearest-neighbor interpolation [46] and pixel-shuffle [47] in the upsampling module to promote the information interaction between network space and depth. However, asymmetric convolution in the multi-scale module can reduce the parameters and affect the accuracy of feature extraction, which is not conducive to restoring the original image's detailed features.

Due to the good performance of attention mechanisms in computer vision tasks represented by image classification [48], object detection [49], and semantic segmentation [50], Zhang et al. [51] first introduced channel attention into the image superresolution reconstruction task and proposed RCAN, which highlights the foreground information of reconstructed images to some extent. The SAN proposed by Dai et al. [52] uses a second-order attention network to capture distant spatial features, leveraging the underlying image features, and the reconstructed image color is closer to the original image. Liu et al. [53] proposed RFANet based on EDSR's proposed RFA module to exploit shallow residual features to achieve a good balance between model performance and parameter number and proposed an ESA spatial domain attention module to extract spatial domain features using stride length convolution and pooling instead of dilation convolution for dimensionality reduction to avoid the lack of image detail information caused by dilated convolution and achieve better results.

Although the above methods have achieved good results in image superresolution tasks, there remain problems such as the image foreground not being highlighted, lack of visual focus, etc. In this paper, we innovatively propose the parallel attention module (PAM) and use it as the basis to introduce skip connection and group convolution to build PAMNet, aiming to design a high-performance, high-quality image superresolution model that attends more to image

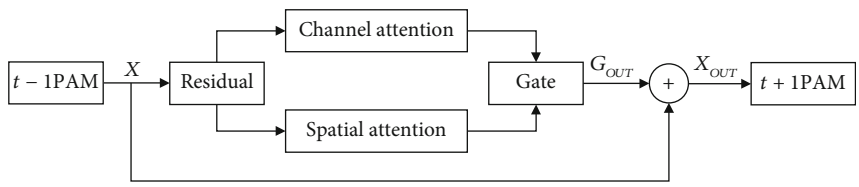


Figure 2: The structure of PAM in time t.

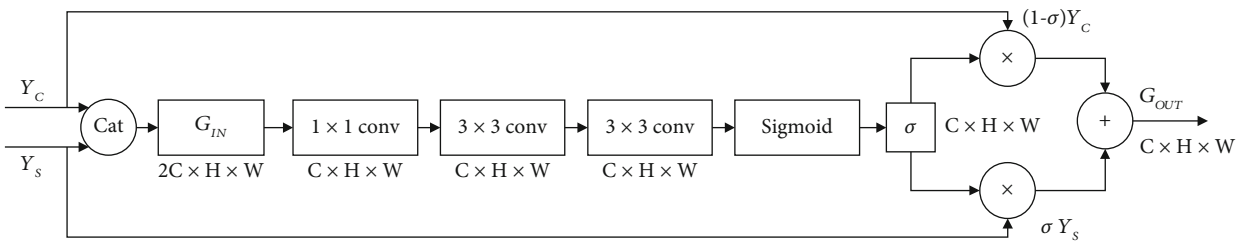


Figure 3: Gate network module.

foreground information and detailed features and has a smaller number of training parameters. The main contributions of this paper are as follows:

(1) Proposing a generic module named PAM, which computes channel attention and spatial attention in parallel on the residual block's residual branch, and then dynamically adjusts the weights of both using gated networks and nonuniform joint loss, so that the PAM module focuses on the attention domain with higher weights and thus can extract foreground information deeply

(2) Based on the PAM modules, we proposed PAMNet. By concatenating multiple PAM modules in PAMNet and introducing skip connections, the residual features from all the preceding PAM modules are fed directly to the PAM module at the end of the network for aggregation to leverage the shallow residual features, and the reconstructed images are closer to ground truth. In addition, by using group convolution, PAMNet is more lightweight than other methods

The reminder of the paper is organized as follows. Section 2 describes the PAM module, PAMNet, and the loss function of this paper in detail. Section 3 verifies the effectiveness and generality of this paper's method through ablation experiments and comparison experiments. Finally, Section 4 presents the conclusion of this study.

2. Method

2.1. PAM. The PAM module proposed in this paper can directly replace the residual block in the ResNet [54] backbone network, compute channel and spatial attention in parallel, splice the results in the channel dimension, and feed them into the gated network to extract the weight coefficients. In the backpropagation process, the channel attention and spatial attention weights are dynamically adjusted by nonuniform joint loss, focusing on extracting image foreground information in the attention domain with higher weights. The specific structure of PAM is shown in Figure 2.

In computing channel attention, a structure similar to SENet [55] is used, and the fully connected layer in SENet is replaced by 1×1 convolution, which can preserve the image's spatial features. The specific computation of channel attention is given by Eq. (1):

$$YC = X + CA(XR), \tag{1}$$

where $X \in R^{C\times H\times W}$ represents the input of the residual block, $XR \in R^{C\times H\times W}$ represents the output after computing the residuals, CA represents computing channel attention, and YC represents the final output of the channel attention. Meanwhile, in this paper, C represents the channel dimension of the feature map, H represents the height of the feature map, and W represents the width of the feature map, so that the three dimensions of a feature map can be represented as (C, H, W).

Referring to the HDC idea [56], PAM computes spatial attention using a three-layer cascaded dilation convolution with dilation rates of 1, 2, and 3. First, we use a 1×1 convolution to downscale the feature map with input dimensions (C, H, W) into a feature map with $(C/K, H, W)$ dimensions, where K is the downscaling factor, and in this paper, we take $K = 4$. Second, the feature map after downscaling is convoluted with three different expansion rates to expand the perceptual field with the minimum number of parameters in a finite number of steps to ensure the continuity of the perceptual domain and avoid the information loss caused by pooling. Finally, we use a 1×1 convolution to fuse the information of different channels of the feature map and go through Sigmoid activation to get the feature map weights in the $(1, H, W)$ dimension and assign the weights in the (H, W) dimension to multiply to the input feature map to focus on the image foreground information. The

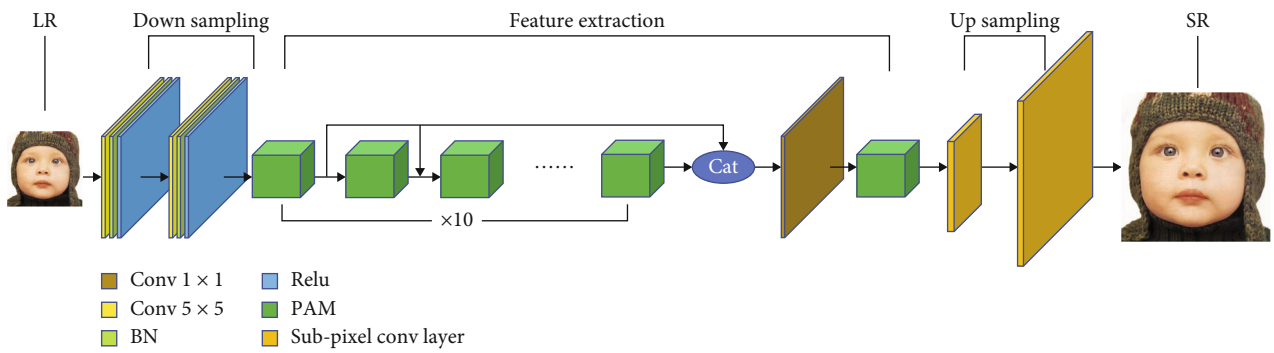


Figure 4: The structure of PAMNet.

specific computation of spatial attention is given by Eq. (2):

$$YS = X + SA(XR), \tag{2}$$

where $X \in R^{C\times H\times W}$ represents the input of the residual block, $XR \in R^{C\times H\times W}$ represents the output after computing the residuals, SA represents computing spatial attention, and YS represents the final output of spatial attention.

After obtaining the channel attention YC and spatial attention YS using the above method, the two are spliced in the channel dimension to obtain the input of the gated network$GIN \in R^{2C\times H\times W}$. Then, we use a 1×1 convolution to fuse the information and reduce the GIN dimension of (C, H, W). Then, two 3×3 convolutions for feature extraction and Sigmoid activation are used to obtain an activation output $\sigma \in R^{C\times H\times W}$ with values in the range (0, 1). Finally, the final output $GOUT \in R^{C\times H\times W}$ is obtained by multiplying σ by YC and YS as a linear combination of coefficients. Meanwhile, this weight is continuously updated during the backpropagation process, and the weights of channel attention and spatial attention are dynamically assigned in learning progress, focusing on extracting image foreground information in the attention domain with higher weights. The computation is given by Eq. (3):

$$\mathrm{GOUT} = (1-\sigma)YC + \sigma YS. \tag{3}$$

The specific structure of the gated network module is shown in Figure 3.

2.2. PAMNet. PAMNet is built with the PAM module as the core unit and postupsampling as the base structure, using skip connection, group convolution, and feature fusion. The network comprises a down-sampling layer, a feature extraction layer, and an upsampling layer. In this case, the downsampling layer uses a serial 5×5 convolution to initially extract image color, contour, and texture features. The upsampling layer uses pixel shuffle to enlarge the image. PAMNet benefits from the PAM module and skips connection, which focuses more on image foreground information reconstruction and can leverage shallow residual features, highlighting the visual focus of reconstructed images. The specific structure of PAMNet is shown in Figure 4.

The downsampling layer initially extracts the image's underlying features by two times 5×5 convolution and increases the number of feature map channels. The feature extraction layer and the upsampling layer are the core of PAMNet. The feature extraction layer uses PAM as the basic unit and serially multiple PAM modules to extract detailed features. The basic structure of the traditional residual block is two 3×3 same convolutions; serializing multiple blocks induces many parameters and complex computations, which seriously slows down the model's training. Therefore, in this paper, we use group convolution in the PAMNet feature extraction layer to reduce the number of parameters and add 1×1 convolution to fuse the group information. Taking the input feature map $XIN \in R^{C\times H\times W}$, output feature map $XOUT \in R^{C\times H\times W}$, and convolution kernel $F \in R^{C\times 3\times 3}$ as an example, the number of parameters of a residual block is given by Eq. (4):

$$PN = 3\times 3\times C\times C\times 2. \tag{4}$$

While using group convolution with group number g and 1×1 convolution, the number of parameters is reduced to Eq. (5):

$$\begin{aligned} PG &= \left(3\times 3\times C\times C\times \frac{1}{g} + 1\times 1\times C\times C\right)\times 2 \\ &= 2\times C\times C\times \left(3\times 3\times \frac{1}{g} + \right)1. \end{aligned} \tag{5}$$

According to the reference [57], take $C = 64$ and $g = 16$, we can get $PN = 73728$ and $PG = 12800$; we can see that the number of parameters using grouped convolution is only 17.36% of the normal convolution, which simplifies the number of parameters of the model while significantly increasing the training speed. Since the shallow residual features must pass through multiple computations before reaching the last PAM module, the deeper layers of the network fail to leverage the shallow information and lose some of the image's shallow features, which is inconvenient for reconstructing the image's color and texture information and severely limits the model's image reconstruction capability. Existing SR methods, such as RFANet, only use skip connection inside the RFA module, which fails to preserve

Table 1: Network training parameters.

Parameters	Values
Scale	4
Batch size	16
Optimizer	Adam
Learning rate	0.0002
PAM channels	64
Group of convolution	16

the shallow features of the image completely. In this study, we introduce a skip connection within PAMNet, and by skip connection, we input the residual features of all preceding PAM modules to the last PAM module in the feature extraction layer, reducing dimensionality and aggregating shallow features by 1×1 convolution. Compared with the simple stacking of multiple residual blocks, PAMNet retains the underlying image information so that it can participate in the subsequent computation to further extract the high-level semantic information while sending it directly to the end PAM module without any interference, which retains the underlying features and focuses on extracting the high-level image information.

The upsampling layer acts as the final layer of the network and is responsible for scaling the image to a specified magnification. Commonly used upsampling methods include linear interpolation, deconvolution [58], transposed convolution [59], subpixel convolution [60], and metaupscale [61]. Interpolation methods are the fastest, but reconstructed images are blurred and have low definition. Deconvolution and transposed convolution reconstruct images with a field of perception up to the same magnification as the image, which is not conducive to obtaining global features, and the reconstructed images are prone to checkerboard artifacts. Subpixel convolution has a larger field of perception and more contextual information, and the reconstructed image is clear in detail. The metaupscale does not need to determine the scale factor in advance, the image can be continuously enlarged by any factor, and the reconstructed image is high definition, which is often used for video superresolution reconstruction. Due to the faster computation speed of subpixel convolution and the high quality of reconstructed images, pixel shuffle is used for upsampling in this paper.

2.3. Loss Function. Similar to existing methods [42, 43, 51–53], this paper trains the network model based on the generative adversarial structure and optimizes the model parameters by the joint discriminator loss and generator loss, where discriminator loss LD is defined as Eq. (6):

$$LD = -\mathbb{E}xr[\log(D(xr, xf))] - \mathbb{E}xf[\log(1 - D(xf, xr)], \quad (6)$$

where xr is the real image, xf is the reconstructed image, and $D(xr, xf)$ computes the difference between the real image and the reconstructed image and uses the Sigmoid restriction $D(xr, xf) \in (0, 1)$.

Unlike the above methods, the generator loss in this paper comprises nonuniform joint loss, adversarial loss, and content loss. By using nonuniform joint loss, constraint the network learn image color and texture features while extracting more discriminative features and detailed information, focusing more on the reconstruction of image foreground information.

The nonuniform joint loss LU is based on L1 loss, and the reconstructed image and the original image are fed into the pretrained VGG-19 network to compute L1 loss LVGG1 before the first pooling layer and L1 loss LVGG2 before the last pooling layer by adjusting the weights of LVGG1 and LVGG2 to constrain the generator to extract the underlying features while learning more detailed information and discriminative features. The specific computation is given by Eq. (7):

$$LU = \alpha L\text{VGG1} + \beta \text{LVGG2}, \quad (7)$$

where α is the weight of LVGG1, and β is the weight of LVGG2; in this paper, we take $\alpha = 0.2$ and $\beta = 1$.

The adversarial loss LG is computed as in [11], and the specific computation is defined by Eq. (8):

$$LG = -\mathbb{E}xr[\log(1 - D(xr, xf))] - \mathbb{E}xf[\log(D(xf, xr)]. \quad (8)$$

Content loss LC computes the pixel difference between the real image and the reconstructed image using both L1 loss and L2 loss. Methods such as RFANet only use the L1 loss to compute the content loss, which induces the loss of some high-frequency information in the reconstructed images, and L1 loss is prone to sparse solutions and cannot be derived at the zero point, increasing the instability of GAN training. SRGAN only uses L2 loss to compute the content loss, which is influenced by outlier points. Although the reconstructed image has a higher peak signal-to-noise ratio (PSNR (dB)) but is prone to artifacts, the visual effect is poor, which opposes the original intention of image superresolution. PAMNet computes content loss using both L1 loss and L2 loss to enhance the method's robustness while reducing sparse solutions. The specific computation is given by Eq. (9):

$$LC = \mu L1(xr, xf) + \theta L2(xr, xf), \quad (9)$$

where xr represents the ground truth, xf represents the reconstructed image, and μ and θ represent the weight of L1 loss and L2 loss, respectively. In this study, we take $\mu = 0.75$ and $\theta = 0.25$.

In summary, the generator loss is defined by Eq. (10):

$$L = \gamma LG + \lambda LU + \eta LC, \quad (10)$$

where γ, λ, and η represent the weights of adversarial loss, nonuniform joint loss, and content loss. In this paper, we take $\gamma = 0.005$, $\lambda = 1$, and $\eta = 0.1$.

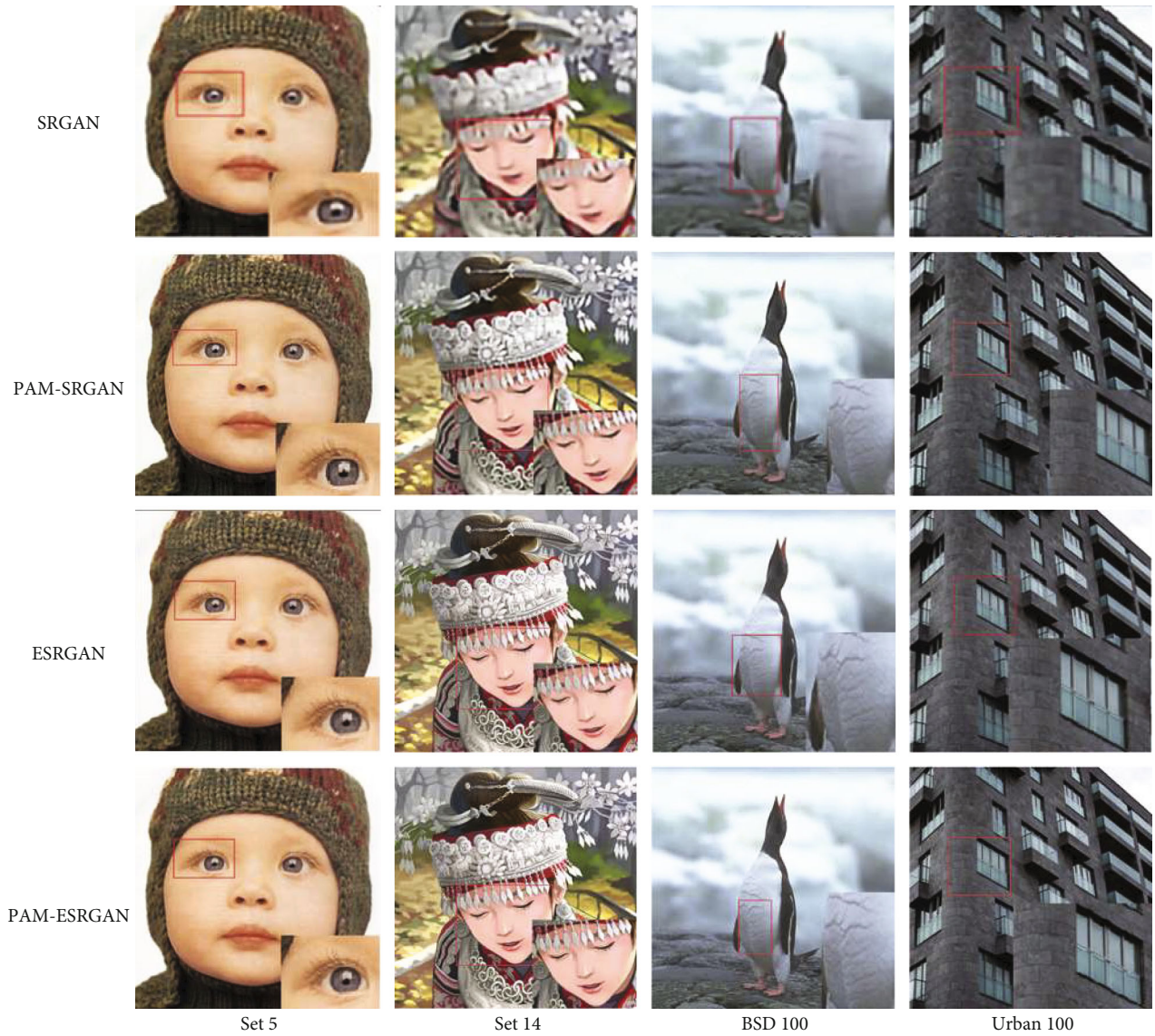


Figure 5: SRGAN, PAM-SRGAN, ESRGAN, and PAM-ESRGAN.

3. Experiment

3.1. Settings. Referring to the existing methods [40, 43, 51–53], to verify the effectiveness of this paper, we select 3450 images from DIV2K [62] and Flickr2K [63] as the training dataset and randomly select 60,000 subimages as the training images after cropping and mirror reversal operations on the original images. Meanwhile, we select Set5 [64], Set14 [65], BSD100 [66], and Urban100 [67] as the test datasets. The main parameters of the network are shown in Table 1.

This paper is based on PyTorch for experiments with the following hardware parameters: Intel i7 9700, NVIDIA 2080ti, and 32gRAM.

3.2. Results. In this paper, we focused on SISR reconstruction on a four-time deflation factor and used Set5, Set14, BSD100, and Urban100 as the test sets to compare with existing image superresolution methods from both subjective and objective aspects. We also embedded the PAM module into the backbone networks of SRGAN and ESRGAN to verify the effectiveness and generality of the module. Meanwhile, PSNR and SSIM were used as objective indices to quantify the quality of the reconstructed images.

3.2.1. Effectiveness and Generality of PAM. This section verifies the effectiveness and generality of the PAM module by replacing the basic residual block of SRGAN and the RRDB structure in ESRGAN using the PAM module and keeping the other structures and loss functions in the original network unchanged. The replaced models are called PAM-SRGAN and PAM-ESRGAN, and we selected the images in Set5, Set14, BSD100, and Urban100 for analysis. The results are shown in Figure 5.

The performance of SRGAN and ESRGAN with embedded PAM modules on different datasets is shown in Table 2.

Table 2: SRGAN, PAM-SRGAN, ESRGAN, and PAM-ESRGAN.

Method	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM
SRGAN	29.40/0.847	26.02/0.739	25.16/0.669	24.29/0.661
PAM-SRGAN	31.24/0.883	27.93/0.749	25.91/0.701	24.83/0.692
ESRGAN	32.60/0.901	28.88/0.791	27.76/0.745	26.73/0.815
PAM-ESRGAN	32.74/0.902	28.91/0.787	27.80/0.745	26.94/0.813

Table 3: Numbers of PAM.

N	Set5 PSNR	Set14 PSNR	BSD100 PSNR	Urban100 PSNR
3	22.80	20.15	19.97	19.76
5	28.14	24.86	23.93	23.54
7	29.98	26.93	25.26	25.31
9	32.13	28.46	27.48	26.47
11	32.73	28.93	27.81	26.93

As seen in Table 2, PAM-SRGAN improves PSNR by 1.84 dB, 1.91 dB, 0.75 dB, and 0.54 dB over SRGAN on the four test sets; PAM-ESRGAN improves PSNR by 0.14 dB, 0.03 dB, 0.04 dB, and 0.21 dB over ESRGAN on the four test sets. The results show that the PAM module improves the performance of SRGAN and ESRGAN networks with good generality.

3.2.2. Performance of PAMNet. To give PAMNet the best performance, we performed the following experiments on the number of PAM modules in the feature extraction layer. Let the total number of PAM modules in PAMNet be N and $N \in [3, 5, 7, 9, 11]$. Keeping the other structures in PAMNet unchanged, the test results on different datasets are shown in Table 3.

As seen in Table 3, the performance of PAMNet outperforms SOTA method RFB-ESRGAN (32.66 dB, 28.88 dB, 27.79 dB, and 26.92 dB) and RFANet (32.72 dB, 28.91 dB, 27.77 dB, and 26.89 dB) when $N = 11$.

As the number of PAM modules (N) increases, the PSNR of PAMNet reconstructed images on different datasets grows accordingly. The variation relationship is shown in Figure 6 for the Urban100 dataset, for example.

Figure 6 shows that the PSNR of the reconstructed images does not continue to improve significantly with the increase in the number of PAM modules (N), and for a good balance between model performance and complexity, PAMNet takes $N = 11$. After determining the number of PAM modules and selecting images from the Set5, Set14, BSD100, and Urban100 test sets, one image from each test set was taken for analysis, and the results are shown in Figure 7.

As can be seen in Figure 7, due to the addition of a gated network and nonuniform joint loss in PAMNet, our method can produce sharper foreground information than existing methods (Figures 7(a) and 7(b)), and the detailed texture features of the reconstructed images are closer to Ground Truth (Figures 7(c) and 7(d)). In addition, PAMNet basically preserves the color and texture features of the image by introducing skip connection, and the overall image sharpness is basically on par with SOTA methods such as RFB-ESRGAN and RFANet.

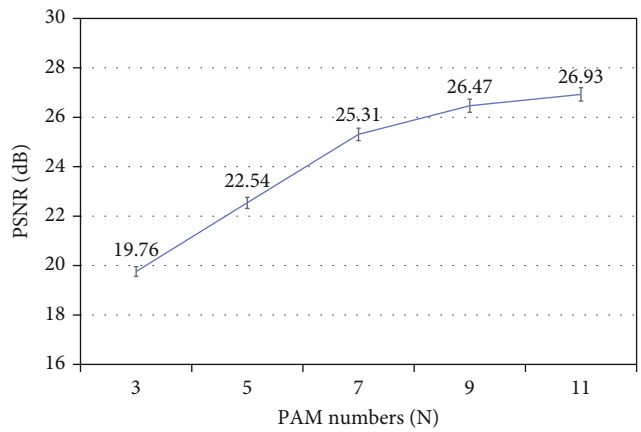


Figure 6: PAM numbers vs. PSNR on Urban100.

To verify the effectiveness of PAMNet from an objective perspective, we selected PSNR and SSIM as objective indices. The PSNR and SSIM of each image in Figure 7 are shown in Table 4.

As shown in Table 4, the PSNR and SSIM of PAMNet reconstructed images outperformed other methods, and only the RFB-ESRGAN method had slightly higher PSNR than PAMNet on Figure 6(a). To verify the generalization performance of PAMNet, the PSNR and SSIM of different methods on different test sets are shown in Table 5.

As shown in Table 5, the PSNR of PAMNet reconstructed images outperformed other methods in each test set, improving 0.01 dB, 0.02 dB, 0.04 dB, and 0.04 dB over RFB-ESRGAN in four test sets and improving 0.07 dB, 0.05 dB, 0.02 dB, and 0.01 dB over RFANet, and SSIM was Set5, and Urban100 datasets were slightly lower than RFB-ESRGAN. The experimental results show that, thanks to the PAM module and nonuniform joint loss, PAMNet can effectively extract image foreground information, improve the PSNR and SSIM of the reconstructed images, and enhance the foreground clarity while ensuring a clear background in the reconstructed images.

3.2.3. The Effect of Skip Connection. In this section, the skip connection in PAMNet was removed, and the other structures and loss functions were kept unchanged to investigate the effect of skip connection on PAMNet. The experimental results are presented in Table 6.

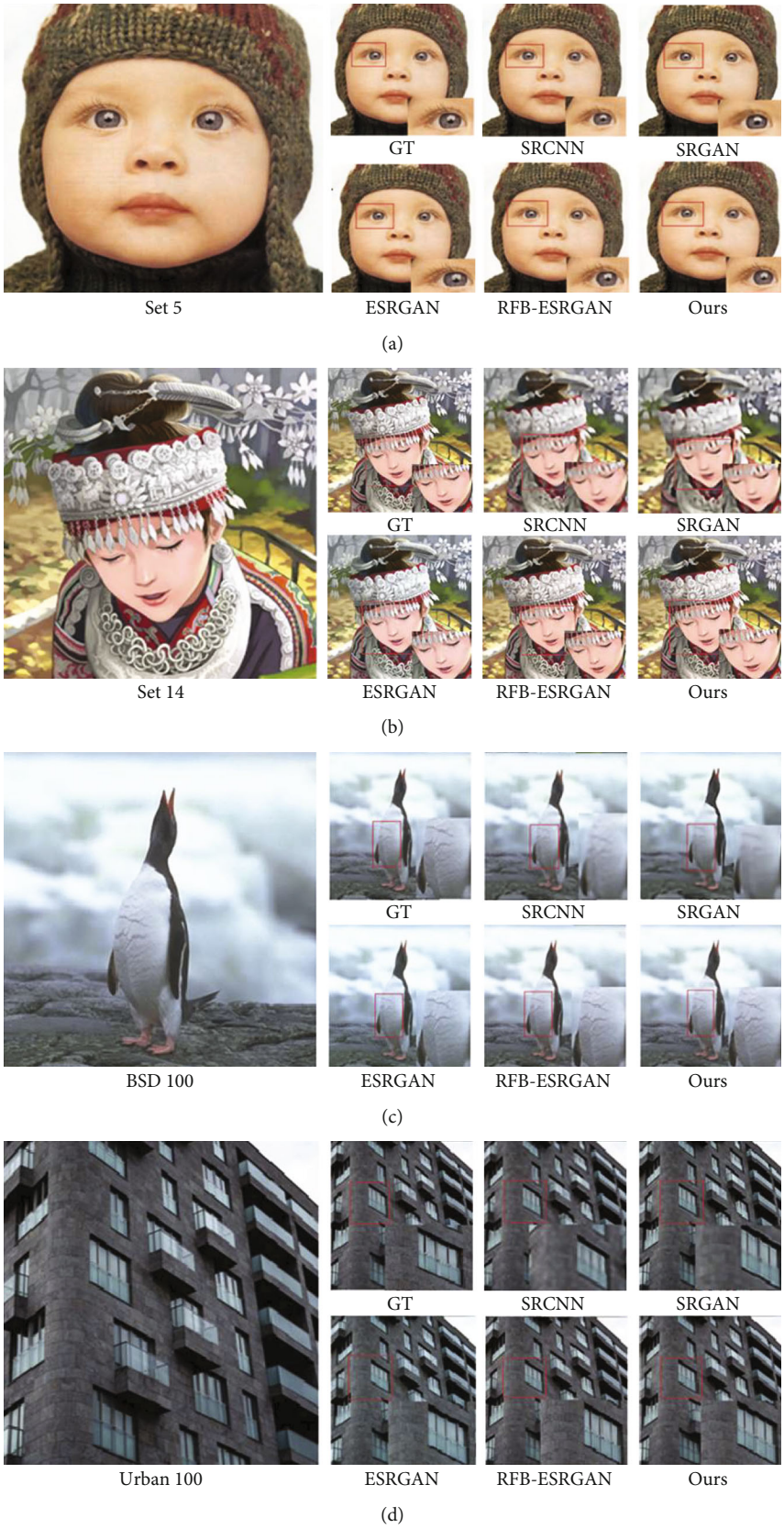


Figure 7: Reconstructed image of PAMNet: (a) from Set5, (b) from Set14, (c) from BSD100, and (d) from Urban100.

Table 4: Objective indicator comparison from Figure 7.

Method	Figure 7(a) PSNR/SSIM	Figure 7(b) PSNR/SSIM	Figure 7(c) PSNR/SSIM	Figure 7(d) PSNR/SSIM
SRCNN	29.04/0.815	29.28/0.786	26.72/0.748	26.95/0.782
SRGAN	30.11/0.862	28.06/0.775	27.02/0.743	27.29/0.779
ESRGAN	33.87/0.911	29.34/0.825	28.03/0.796	27.93/0.813
RFB-ESRGAN	33.91/0.912	31.02/0.843	28.13/0.797	28.07/0.823
PAMNet (ours)	33.89/0.916	32.65/0.851	28.41/0.804	28.24/0.830

Table 5: Average objective indicator.

Method	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM
SRCNN	30.48/0.862	27.50/0.751	26.90/0.710	24.52/0.722
SRGAN	29.40/0.847	26.02/0.739	25.16/0.669	24.29/0.661
VDSR	31.35/0.883	28.02/0.768	27.29/0.7260	25.18/0.754
EDSR	32.46/0.896	28.80/0.787	27.71/0.742	26.64/0.803
DBPN	32.47/0.898	28.82/0.786	27.72/0.740	26.38/0.794
ESRGAN	32.60/0.901	28.88/0.791	27.76/0.745	26.73/0.815
SAN	32.64/0.900	28.92/0.788	27.78/0.743	26.79/0.806
RFB-ESRGAN	32.72/0.902	28.91/0.801	27.77/0.746	26.89/0.817
RFANet	32.66/0.900	28.88/0.789	27.79/0.744	26.92/0.811
PAMNet(ours)	32.73/0.901	28.93/0.802	27.81/0.750	26.93/0.816

Table 6: Skip connection in PAMNet.

Name	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM
No skip connection	32.21/0.894	28.36/0.801	27.65/0.744	26.69/0.812
Skip connection	32.73/0.901	28.930.802	27.81/0.750	26.93/0.816

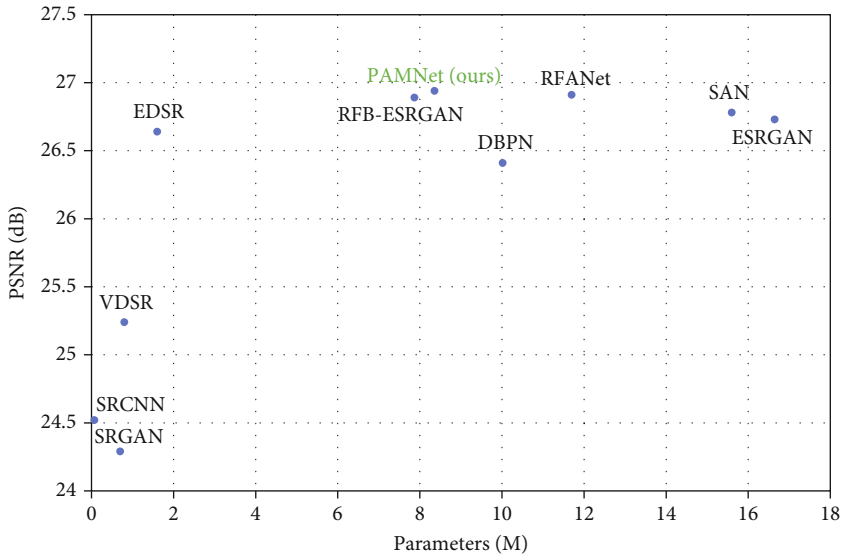


Figure 8: PSNR vs. parameters on Urban100 (×4).

As shown in Table 6, the PSNR of PAMNet reconstructed images on different test sets decreased by 0.52 dB, 0.57 dB, 0.16 dB, and 0.24 dB, after removing the skip connection in PAMNet, and the performance of PAMNet decreased significantly, which constrained the utilization of shallow features by the model. The experimental results show that the skip connection significantly impacted PAMNet, and the use of skip connection could improve the utilization of shallow features in PAMNet, thereby enhancing the comprehensive performance of the model.

3.2.4. Model Complexity. To evaluate the complexity of the PAMNet model, it was compared with existing SR methods: SRCNN, SRGAN, VDSR, EDSR, DBPN, SAN, ESRGAN, RFB-ESRGAN, and RFANet. The results are shown in Figure 8.

As seen in Figure 8, PAMNet has smaller parameters and better performance than DBPN, RFANet, SAN, and ESRGAN. Compared with RFB-ESRGAN, PAMNet has a slightly larger number of parameters but slightly outperforms RFB-ESRGAN overall.

4. Conclusion

In this paper, we proposed a generic PAM module for image superresolution reconstruction to extract foreground information and high-frequency features of images. The module computed channel attention and spatial attention in parallel and used the gated network to extract the two-weight coefficients and cooperated with the nonuniform joint loss to dynamically modify the two weights during the backpropagation process, so that the network attended more to the extraction of foreground information and discriminative features. To fully reflect the good performance of PAM modules, PAMNet was further proposed to connect multiple PAM modules in series in PAMNet. The ablation experiments verified the effectiveness and generality of the PAM module and the necessity of skip connection in PAMNet. By contrast experiments with existing state-of-the-art image superresolution methods, the average PSNR improvement of PAMNet on different data sets is 0.4 dB, and the average SSIM improvement is 0.005. It is verified that PAMNet achieves a good balance between performance and model complexity. By using PAMNet, in many applications of urban IoT systems, such as autonomous vehicles, smart healthcare, and urban surveillance, it is possible to generate clearer and more foreground-focused high-resolution images than existing image superresolution methods, improving the reliability of urban IoT systems and satisfying people's vision of a better life. Limited by the training equipment and the research content, only the image superresolution method on the ×4 magnification factor has been studied. In the future, we will also continue to research faster and greater magnification image superresolution methods, so that various smart technologies can continue to benefit humanity and all families.

Data Availability

The open datasets used to support the findings of this study are included within the article. The link is as follows: https://data.vision.ee.ethz.ch/cvl/DIV2K/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by Telecommunications Advancement Foundation (Japan) Research Grant, RIEC Nationwide Cooperative Research Projects, Research Institute of Electrical Communication, Tohoku University, Japan, H31/B18, and ROIS NII Open Collaborative Research 2021 (21FA03).

References

[1] K. Yu, Z. Guo, Y. Shen, J. C. W. Lin, T. Sato, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, 2021.

[2] L. Tan, K. Yu, F. Ming, X. Cheng, and G. Srivastava, "Secure and resilient artificial intelligence of things: a HoneyNet approach for threat detection and situational awareness," *IEEE Consumer Electronics Magazine*, p. 1, 2021.

[3] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13784–13795, 2020.

[4] Z. Guo, K. Yu, A. Jolfaei, A. K. Bashir, A. O. Almagrabi, and N. Kumar, "fuzzy detection system for rumors through explainable adaptive learning," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 12, pp. 3650–3664, 2021.

[5] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, "Anti-Forensics for face swapping videos via adversarial training," *IEEE Transactions on Multimedia*, p. 1, 2021.

[6] F. Ding, G. Zhu, M. Alazab, X. Li, and K. Yu, "Deep-learning-empowered digital forensics for edge consumer electronics in 5G HetNets," *IEEE Consumer Electronics Magazine*, p. 1, 2020.

[7] L. Tan, K. Yu, L. Lin et al., "Speech emotion recognition enhanced traffic efficiency solution for autonomous vehicles in a 5G-enabled space-air-ground integrated intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021.

[8] J. Zhang, X. Hu, Z. Ning et al., "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633–2645, 2017.

[9] Z. Ning, X. Hu, Z. Chen et al., "A cooperative quality-aware service access system for social internet of vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2506–2517, 2017.

[10] B. Hu, G. P. Gao, L. L. He, X. D. Cong, and J. N. Zhao, "Bending and on-arm effects on a wearable antenna for 2.45 GHz body area network," *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 378–381, 2016.

[11] X. Hu, J. Cheng, M. Zhou et al., "Emotion-aware cognitive system in multi-channel cognitive radio ad hoc networks," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 180–187, 2018.

[12] K. Yu, L. Tan, S. Mumtaz et al., "Securing critical infrastructures: Deep-Learning-Based threat detection in IIoT," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 76–82, 2021.

[13] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the internet of health things," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, 2021.

[14] L. Zhen, Y. Zhang, K. Yu, N. Kumar, A. Barnawi, and Y. Xie, "Early collision detection for massive random access in satellite-based internet of things," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5184–5189, 2021.

[15] Z. Guo, K. Yu, Y. Li, G. Srivastava, and J. C. W. Lin, "Deep learning-embedded social internet of things for ambiguity-aware social recommendations," *IEEE Transactions on Network Science and Engineering*, 2021.

[16] T. Guo, K. Yu, M. Aloqaily, and S. Wan, "Constructing a prior-dependent graph for data clustering and dimension reduction in the edge of AIoT," *Future Generation Computer Systems*, vol. 128, pp. 381–394, 2022.

[17] Y. Gong, L. Zhang, R. Liu, K. Yu, and G. Srivastava, "Nonlinear MIMO for industrial Internet of Things in cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5533–5541, 2021.

[18] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 8, pp. 2072–2085, 2015.

[19] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K. K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones," *IEEE Internet of Things Journal*, 2021.

[20] C. Feng, B. Liu, K. Yu, S. K. Goudos, and S. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[21] K. Yu, L. Tan, C. Yang et al., "A blockchain-based shamir's threshold cryptography scheme for data protection in industrial internet of things settings," *IEEE Internet of Things Journal*, 2021.

[22] F. Ding, K. Yu, Z. Gu, X. Li, and Y. Shi, "Perceptual enhancement for autonomous vehicles: restoring visually degraded images for context prediction via adversarial training," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[23] L. Zhao, H. Li, N. Lin, M. Lin, C. Fan, and J. Shi, "Intelligent content caching strategy in autonomous driving Toward 6G," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, pp. 1–11, 2021.

[24] L. Zhao, W. Zhao, A. Hawbani et al., "Novel online sequential learning-based adaptive routing for edge software-defined vehicular networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 2991–3004, 2021.

[25] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote e-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.

[26] L. Tan, K. Yu, N. Shi, C. Yang, W. Wei, and H. Lu, "Towards secure and privacy-preserving data sharing for COVID-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 271–281, 2021.

[27] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "PMRSS: privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1981–1990, 2022.

[28] W. Shang, J. Chen, H. Bi, Y. Sui, Y. Chen, and H. Yu, "Impacts of COVID-19 pandemic on user behaviors and environmental benefits of bike sharing: a big-data analysis," *Applied Energy*, vol. 285, no. 116429, p. 116429, 2021.

[29] H. Peng, B. Hu, Q. Shi et al., "Removal of ocular artifacts in EEG—an improved approach combining DWT and ANC for portable applications," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 600–607, 2013.

[30] L. Yang, K. Yu, S. X. Yang, C. Chakraborty, Y. Lu, and T. Guo, "An intelligent trust cloud management method for secure clustering in 5G enabled internet of medical things," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[31] D. Wang, Y. He, K. Yu, G. Srivastava, L. Nie, and R. Zhang, "Delay sensitive secure NOMA transmission for hierarchical HAP-LAP medical-care IoT networks," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[32] J. L. Harris, "Diffraction and resolving power," *Journal of the Optical Society of America*, vol. 54, no. 7, pp. 931–936, 1964.

[33] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020.

[34] H. Stark and P. Oskoui, "High-resolution image recovery from image-plane arrays, using convex projections," *Journal of the Optical Society of America A*, vol. 6, no. 11, pp. 1715–1726, 1989.

[35] M. Irani and S. Peleg, "Super resolution from image sequences," in *Proceedings. 10th International Conference on Pattern Recognition*, pp. 115–120, Atlantic City, USA, 1990.

[36] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[37] C. Ma, C. Y. Yang, X. Yang, and M. H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017.

[38] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[39] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 1646–1654, Las Vegas, NA, USA, 2016.

[40] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017, pp. 136–144, Hawaii, HI, USA, 2017.

[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014.

[42] C. Ledig, L. Theis, F. Huszár et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, CVPR 2017, pp. 4681–4690, Hawaii, HI, USA, 2017.

[43] X. Wang, K. Yu, S. Wu et al., "Esrgan: enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision, ECCV 2018*, Munich, Germany, 2018.

[44] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," in *International Conference on Learning Representations, ICLR 2019*, New Orleans, Louisiana, USA, 2019.

[45] T. Shang, Q. Dai, S. Zhu, T. Yang, and Y. Guo, "Perceptual extreme super-resolution network with receptive field block," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR 2020*, pp. 440-441, Seattle, WA, USA, 2020.

[46] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, ""When is "nearest neighbor" meaningful?"," in *International Conference on Database Theory*, pp. 217–235, Berlin, Heidelberg, 1999.

[47] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR 2018*, pp. 2472–2481, Salt Lake City, UT, USA, 2018.

[48] T. Durand, N. Mehrasa, and G. Mori, "Learning a deep Convnet for multi-label classification with partial labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pp. 647–657, Long Beach, CA, USA, 2019.

[49] T. Wang, T. Yang, and M. Danelljan, "Learning human-object interaction detection using interaction points," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pp. 4116–4125, Seattle, WA, USA, 2020.

[50] C. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang, "Context prior for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pp. 12416–12425, Seattle, WA, USA, 2020.

[51] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision, ECCV 2018*, pp. 286–301, Munich, Germany, 2018.

[52] T. Dai, J. Cai, Y. Zhang, S. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pp. 11065–11074, Long Beach, CA, USA, 2019.

[53] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu, "Residual feature aggregation network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pp. 2359–2368, Seattle, WA, USA, 2020.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016,*, pp. 770–778, Las Vegas, NV, USA, 2016.

[55] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pp. 7132–7141, Salt Lake City, UT, USA, 2018.

[56] Y. Yuan, K. Yang, and C. Zhang, "Hard-aware deeply cascaded embedding," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017*, pp. 814–823, Venice, Italy, 2017.

[57] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi, "Deep roots: improving cnn efficiency with hierarchical filter groups," in *Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR 2017*, pp. 1231–1240, Honolulu, HI, USA, 2017.

[58] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pp. 2528–2535, San Francisco, CA, USA, 2010.

[59] H. Gao, H. Yuan, Z. Wang, and S. Ji, "Pixel transposed convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1218–1227, 2020.

[60] W. Shi, J. Caballero, F. Huszar et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 1874–1883, Las Vegas, NV, USA, 2016.

[61] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, "Meta-SR: a magnification-arbitrary network for super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pp. 1575–1584, Long Beach, CA, USA, 2019.

[62] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: dataset and study,," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops, CVPRW2017*, pp. 126–135, Honolulu, HI, USA, 2017.

[63] R. Timofte, E. Agustsson, L. Van Gool, M. H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops, CVPRW2017*, pp. 114–125, Honolulu, HI, USA, 2017.

[64] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proceedings of the 23rd British Machine Vision Conference, BMVC 2012*, Guildford, U.K., 2012.

[65] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, pp. 711–730, Springer, Berlin, Heidelberg, 2010.

[66] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001*, vol. 2, pp. 416–423, Vancouver, Canada, 2001.

[67] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR 2015*, pp. 5197–5206, Boston, MA, USA, 2015.