Computer Science and Information Systems 16(3):689–704 https://doi.org/10.2298/CSIS180830022Z

SARSA Based Access Control with Approximation by TileCoding

Fei Zhu[1,2], Pai Peng[1], Quan Liu[1], Yuchen Fu[3], and Shan Zhong[4]

[1] School of Computer Science and Technology, Soochow University
Shizi Street No.1 158 Box, Suzhou, Jiangsu 215006, China
zhufei@suda.edu.cn, 20185227062@stu.suda.edu.cn,quanliu@suda.edu.cn
[2] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University
Shizi Street No.1 158 Box, Suzhou, Jiangsu 215006, China
[3] School of Computer Science and Engineering, Changshu Institute of Technology
Changshu, Jiangsu, 215500, China
yuchenfu@cslg.edu.cn
[4] School of Computer Science, University of Wisconsin
Eau Claire, Wisconsin, USA
zhongs@uwec.edu

Abstract. Traditional sensor nodes ignore the packet loss rate during information transmission and the access control security problem caused by server utilization when uploading data. To solve the problem, we propose a SARSA based access control method with approximation by TileCoding (SACT), which takes the sensor packet loss rate and the server error rate into account. The network state is estimated by the packet loss rate and variable bit error rate to get a server access control strategy to improve security performance. The eventual strategy complies with the minimum information loss and the maximum server utilization. Results of experiments show that the algorithm is capable of achieving good results in the total amount of information received by the server system. The SACT improves the server utilization rate and the overall security performance of the network.

Keywords: access control, TileCoding, information security, SARSA.

1. Introduction

Access control is a way of granting or restricting the subject's access to the object explicitly featuring a very important information security technology. Access control technology is applied in many areas of information systems such as intrusion detection system [5], information encryption [12], identity authentication [7], security audit [3], security and risk analysis [14]. It combines technology with theory to guarantee safe and reliable transmission. It is also applied to access information system [32], which largely retains the integrality of information effectively and reduces the information leak and omission. Access control system takes some defensive measures to manage the access of system resources so as to ensure the full use of system resources.

Authorization policy [4] is the core issue of access control. With the development of distributed computing and information technology, heterogeneous networks are interconnected and are increasingly communicating. In order to ensure secure networks, we

should consider the tense, environment and other factors when establishing the access control model. However, information error and loss will inevitably occur in the entire information transmission process from the overall and long-term perspective. Consequently, the critical issue is to maximize the safety and effectiveness of transmission of information. Quantitative security of information transmission needs to be focused.

In the wireless sensor network, the sensor nodes will have unique information loss [27] when they upload data to the server, which is called lose bag and will cause the server to receive the unevenly distributed data flow. The number of servers available is rarely taken into account in the traditional access control strategy, which often leads to the inability to receive as much amount of information as possible. On the other hand, the error rate of the port of the wireless sensor network is high enough to the point of resulting the instability of communication between the server and the sensor equipment. At the same time a dynamic change process ought to be reckoned.

Considering different packet loss rates [22] and bit error rates [30] concerning transmission security problems, we propose an access control strategy based on TileCoding reinforcement learning algorithm, referred as SACT, that considers the sensor with packet loss and a limited number of servers, and takes different sensor nodes that upload data at different times as the starting point. Considering the number of available servers and the error rate, it coordinates the rejection and reception of the sensor at certain timestamps, which makes the whole system receive the most information. Access control strategy and the algorithms of SARSA are introduced in the next section.

2. Related work

2.1. Access control

The core of access control is the authorization policy [15], which controls the subject's access to the object. Access control models can be divided into various types including traditional access control models, role-based access control models (RBAC), task-based and role-based access control models (T-RBAC), and task-based and workflow-based access control models (TBAC).

Traditional Access Control Model In general, the traditional access control model can be divided into 2 categories: mandatory access control (MAC) [26] and discretionary access control (DAC) [10]. Mandatory access control takes a coercive measure. It uses up reading/down writing to ensure data integrity and up writing/down reading to ensure data confidentiality. Although the implementation work of the MAC is heavy and the management is inconvenient and not flexible enough, MAC can realize the one-way flow of information to prevent Trojan horses effectively.

The MAC is to search for all users who have access to a particular resource. It can effectively implement authorization management. It is difficult for MAC to deal with situations where an organization modifies its members and where functions are changed.

Role-Based Access Control In role-based access control(RBAC) [29], the concept of roles is introduced, that is, the responsibilities and functions of users within the organization. Each role is under a corresponding function. In this model, permissions are assigned

to roles, and users are assigned to previously assigned roles to obtain permission for roles. Different users are assigned different roles according to their corresponding functions. The system can be simplified by pre-defining the role-permission relationships.

In a workflow environment, the user performing the operation is changing, so are the user's permissions when data flows in a workflow. This is linked to the context of data processing, which traditional access control technologies DAC and MAC cannot implement. The RBAC reference model also includes functional specifications, which are subdivided into management functions [24], support functions [1], review functions and so on.

Task-Based Authorization Control The Task-Based Authorization model(TBAC) [35] is an active security model, which uses dynamic authorization to realize the security model and implement the task-oriented security mechanism . In the model, users combine tasks with access rights. The status of the task determines the permissions that the system grants to the user, based on the task currently executed by the topic. However, TBAC does not make a distinction between tasks and roles and does not realize active access control thereby exposing its own shortcomings. Generally speaking, TBAC is used in combination with RBAC.

The access control of the object is not static, but varies with the context in which the task is performed. The active and dynamic nature of TBAC makes it widely used in workflow, distributed processing, information processing of multi-point access control and decision making of the the transaction management system.

Task-role-based Access Control Task-role-based Access Control(T-RBAC) [19] is an Access Control Model Based on the enterprise environment. Unlike RBAC, the T-RBAC model treats tasks and roles as equals. In this model, the task owner has specific task requirements and authority constraints. The corresponding task has the corresponding permission, which makes the permission change with the task execution. This can truly realize the demand allocation and dynamic allocation of permission. When the task completes, the role's permission is revoked; when the task is not started, the role has no permission; when the task is executing, the role is assigned permission.

Roles in T-REAC are associated with permission, and tasks serve as the bridge for roles and permission to exchange information. It not only achieves easy and convenient operational maintenance of the roles and task management, but also achieves a more secure system.

2.2. Introduce to SARSA

Reinforcement learning(RL) [34] is a classic and effective method in machine learning. As a branch of the machine learning discipline, deep learning constructs a neural network to simulate the human brain to achieve observational learning. It mimics the working principle of human brain to understand external input data, such as images, texts and sounds. So it is a great tool for implementing artificial intelligence. Reinforcement learning is a class of algorithms that are extremely suitable for achieving artificial intelligence. This discipline is based on the study of animal learning and adaptive control theory. In artificial intelligence problems, Agents are generally used to represent an object with behavioral

capabilities, such as robots, unmanned vehicles, animals, and so on. Agents generally have some specific properties: sociality, autonomy, responsiveness, initiative, and so on. The main issue is the interaction between the agent and the external environment. The probability of a certain behavioral choice increases when the agent chooses it and is rewarded by the external environment and decreases when the agent is punished. However, reinforcement learning does not have a clear decidable signal like the common machine learning. It can only evaluate if the action is encouraged according to the symbol and size of the enhanced signal, so the whole learning process is time consuming.

Reinforcement learning includes many famous algorithms such as Q-learning [36], SARSA [2], function approximation [8] and so on. Reinforcement learning has been widely used in practice in many areas [17,11]. It also has achieved numerous contributions in biomedicine [23] and game playing [18]. In reinforcement learning, the agent interacts with Markov Decision Process(MDP) [33] so as to model the reinforcement learning problem. Reinforcement learning can be described as Fig. 1.

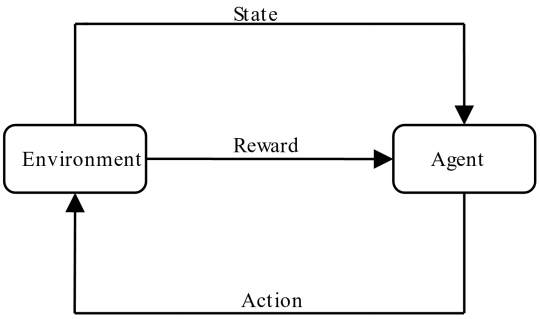

Fig. 1. The process of reinforcement learning

The Markov Decision Process (MDP) is used for modeling. An MDP model is often represented as a tuple $< S, A, P, R >$.

S denotes the state set of agent,$s_t \in S$ represents the state of agent at time t.

A denotes the action set for a state,$a_t \in A$ represents the action that can be taken at time t.

P denotes the transition probability, which is usually expressed as $P(s_{t+1}|a_t, s_t)$, indicating the probability that the agent takes action a' and reach the next state s_{t+1} at the time of t and state s_t.

R denotes the reward function, which is expressed as $r_{t+1} = R(s_t, s_{t+1}, a_t)$, indicating the immediate reward r_t given by the environment after the action a_t was taken from state s_t to state s_{t+1}.

The policy of reinforcement learning π: $S \rightarrow A$ is a mapping from states to actions. In general, reinforcement learning uses state-action function values to evaluate and to improve strategies. The state-action value function is the expected reward when imple-

menting the strategy, which can be generally expressed as:

$$Q(s_t, a_t) = E[R_t|s_t, a_t, \pi] \tag{1}$$

In traditional reinforcement learning, Bellman equation[28] plays an important role, which can be expressed as:

$$Q(s, a) = E[r + \gamma max(Q(s', a')|s, a] \tag{2}$$

where r is the immediate reward,γ is the discount factor, s is the state and a is the action.

SARSA algorithm estimates the action-value function (Q function)[31] rather than the state value function. In other words, we estimate the action value function of all available actions a on any state s under the policy. SARSA algorithm makes full use of markov property, that is, the future state is only related to the current state. SARSA updates the status value at each step using formula as follows:

$$Q(s, a) = Q(s, a) + \alpha(R + \gamma Q(s', a') - Q(s, a)) \tag{3}$$

The complete process of SARSA algorithm is shown in Algorithm 1.

Algorithm 1 SARSA

Input: $Q(s, a)$ arbitrarily
Output: $Q(s, a)$ updated
1: Initialize S
2: **repeat**
3: Choose A from S using policy from Q (e.g., Boltzmann's method)
4: $Q(s, a) \leftarrow Q(s, a) + \alpha[Q(s', a') - Q(s, a)]$
5: $s \leftarrow s', a \leftarrow a'$
6: **until** terminal
7: **return** $Q(s, a)$

The state space and action space are usually large in practical applications. The iterative algorithm used to seek the optimal policy is more computationally intensive and less feasible. Therefore, it is necessary to generalize the large-scale state space. The representing method of function is used to approximate the Q value,as follows:

$$\delta = r_{t+1} + \gamma \max_u Q(s_{t+1}, a_{t+1}; \theta_{t+1}) - Q(s_t, a_t; \theta_t) \tag{4}$$

$$\theta_{t+1} = \theta_t + \alpha \delta e_t \tag{5}$$

$$e_t = \gamma \lambda e_{t-1} + \nabla Q(s_t, a_t; \theta_t) \tag{6}$$

where δ is TD error, e is eligibility trace,and θ is function parameter. Linear functions or nonlinear functions are used to approximate the function. In this paper, function approximation is performed with TileCoding.

In recent years, there are a lot of researches on applying reinforcement learning to the field of access control, routing planning and others. In wireless sensor network structure, the majority of them goes toward two directions. The first is how to reduce the waste of resources, that is, to maximize the use of resources. The second is to improve the efficiency of node and to minimize the cost. Fathi et al.presents q-learning for multiple access control in wireless sensor networks to save energy of sensor node [9]. Chu Yi et al. applies Q-learning to frame based ALOHA as an intelligent slot selection strategy capable of migrating from random access to perfect scheduling [6]. Yun Lin et al. proposes a hybrid spectrum access algorithm that is based on a reinforcement learning model for the power allocation problem of both the control channel and the transmission channel [20]. Many scholars also use the planning method [21] to obtain the optimal solution.

However, these algorithms still suffer from some general issues such as low mobility, security, collision avoidance [25]. Built on the reinforcement learning method, this paper proposes an access control method to maximize the received information data and select the policy with the minimized loss of information.

3. SARSA based Access Control

In the access control of wireless sensor network [16], data transmitted to the server by different sensors may be affected by different levels of error rates, resulting in the loss of important information. The energy consumption of the sensor is also a problem that needs to be considered. In order to address the deficiency of the current sensor network, an optimal access control strategy is got to obtain the most complete information by the server with the minimum energy consumption of the sensor. The algorithm can guarantee a satisfying secure transmission.

In this paper, SARSA based access control with approximation by TileCoding is proposed. The algorithm builds a model based on the idea of reinforcement learning, considering error when uploading, and the package loss of different sensors. We will show the optimal access control policy to achieve the overall minimum information loss and the highest security rate of transmission. The SACT algorithm uses TileCoding to improve the function approximation parameters in the enhanced learning, exploits differential semi-gradient SARSA to constantly update the Q value [13], and utilizes the average reward to update the reward function. After several iterations, the information loss in the transmission of multiple sensors is reduced.

3.1. Model and Algorithm Description

In wireless sensor network, each sensor node may suffer from a series of problems such as information loss, energy consumption and data error when transmitting information to the server. Taking these data security factors into consideration, an algorithm for the access control problem of the sensor node can be obtained. In the algorithm referred in this paper, each server can at the same time receive only one sensor to upload data. We assume the server control system as an agent, which can take steps to accept or reject a sensor. Our goal is to seek the action strategy of the control system for the new sensor node for different states consisting the sensor node and servers.

The optimization of wireless sensor network can be described as a single agent's reinforcement learning problem of seeking out optimal strategies for separate states. In other words, this reinforcement learning problem can be seen as the agent's value function being continuously updated. It is presumed that a wireless sensor network with multiple sensor nodes is sent to a fixed number of servers in a random queue sequence. The system has several features as follows: every server has a certain error rate due to energy loss, which relates to its performance decline over time. Only one sensor node uploads data in a time slice. The server system can decide whether to accept the data uploaded by the sensor node according to the current status. And different sensor nodes have different package loss rates.

Reinforcement learning algorithms require modeling and analysis of state, action, and reward functions. In this wireless sensor network system, n_i servers are initially used to receive data, and there are enough sensor nodes in the queue to transmit data. The sensor nodes are divided into n_j types, and their package loss rate is $e_1, e_2 \cdots e_j$, respectively. In a time slice, only one sensor node can upload data to a server. A busy server has a certain possibility p to complete data transmission and continue to receive the node data of the next sensor. The error rate of the server increases by $w\%$ for every T period. The agent here refers to the server control system, and the state refers to the current number of idle servers, the error rate priority of the server receiving data and the type of sensor node at the current queue head. Action a refers to the option to reject or receive the uploaded data of the current queue head sensor according to the current state. According to the reinforcement learning method, the Q value of state-action team function as well as the state function are evaluated. Some exploration method, such as ϵ-greedy and boltzmann exploration, is used to select the action according to the state function.

3.2. SACT Algorithm

The SACT algorithm is proposed in this paper to obtain the most complete possible information against the increasing server error rate. When considering the reward function, it is needed to integrate the bit error rate of the current server and the package loss rate of sensor nodes. Therefore, the reward function is defined as:

$$r(s_t, a_t) = \alpha(1 - e_t)Inf - \beta w_t \tag{7}$$

where α,β denotes accordingly the weights of the packet loss rate and the bit error rate and Inf denotes the maximum information the sensor node can upload among them.

The agent selects action a_{t+1} with boltzmann probability under the state s_{t+1} with weight w, and updates the average reward and weight w. The average reward and weight are updated as follows:

$$\delta \leftarrow r - \overline{R} + Q(s_{t+1}, a_{t+1}, w) - Q(s_t, a_t, w) \tag{8}$$

$$\overline{R} \leftarrow \overline{R} + m\delta \tag{9}$$

$$w \leftarrow w + n\delta\nabla\overline{Q}(s_t, a_t, w) \tag{10}$$

where r means the immediate reward,m and n are updated steps.

Exploiting the concept of average reward, we define a new type of reward as the sum of the difference values between the rewards and the average rewards, which is called differential reward. The algorithm of differential semi-gradient SARSA is adopted to continuously sample and to learn. The value w and the Q value will converge finally. We apply TileCoding, a kind of CourseCoding to code in the learning stage. A unit of computation is a tiling, and every sensory field characteristic value is associated with a tile. The sensory field of the eigenvalue is divided into the computing unit of the input space. The value function of TileCoding is composed of the weight of each tile, and the expression is:

$$V(s) = \sum_{i=1}^{n} b_i(s) w_i \tag{11}$$

where n denotes the total number of tiles, $b_i(s)$ denotes the ith tile of the corresponding state s and w_i is the corresponding weight.

It is generally not necessary to sum up several tiles for a given state. We update the estimated value of state s according to the following updates expression under the MDP model:

$$\nabla V(s) = \max_a [R(s,a) + \gamma V(T(s,a))] - V(s) \tag{12}$$

Weight update formula is as follows:

$$w_i \leftarrow w_i + \frac{\alpha}{m} b_i(s) \nabla V(s) \tag{13}$$

where m is the number of tilings.

The algorithm of TileCoding is as follows:

Algorithm 2 Updating the parameter by TileCoding

Input: the number of tiles n, the number of tiling m, the initial weights
Output: the weights W updated

for $i = 1$ to n **do**
  Initialize the tiling by using n/m tiles
end for
for $j = 1$ to n/m **do**
  Initialize the tile with weights
end for
repeat
  $s \leftarrow$Random State from S
  $\nabla V(s) = \max_a [R(s,a) + \gamma V(T(s,a))] - V(s)$
  for $k = 1$ to m **do**
    $W \leftarrow W + \alpha/m \nabla V(s)$
  end for
until terminal
return W

Active-tile returns the number of tiles that are activated. By utilizing multiple layers of tilings to generalize in multiple directions, the tradeoff between accuracy and speed is avoided. The tiling of the lattice is used to divide the state space into several tilings in this

paper, which is obtained through the random uniqueness of the discrete grid. Coordinates are obtained in each tiling to obtain the corresponding feature vectors for the number of servers and the status of sensor nodes in the team head. The approximation function of its state space is as follows:

$$V_t(s) = \theta_t^T \phi_s = \sum_{i=1}^{n} \theta_t(i)\phi_s(i) \tag{14}$$

Updating Parameter is expressed as follows:

$$\theta_{t+1} = \theta_t + \alpha[V^\pi(s_t) - V_t(s)]\nabla_{\theta_t} V_t(s_t) \tag{15}$$

TileCoding allocates several large tiles at the beginning of learning. These tiles will be then divided into several sub-tiles all at once. Tiles are split into two parts in this article to simplify calculation. The learning speed and performance indexes were optimized on the basis of compound tiling. Set a global counter u to record the update times. When u reaches the threshold, select a tile to segment. Since the quota policy is based on control or prediction, sub-tiles need to be recorded or screened. So,k status values contain $2k$ tiles. When each tile is generated, the weight of the sub-tile is initialized to 0. As the status is continuously updated, the activated sub tile k is updated accordingly. The following diagram illustrates this process. Fig. 2 shows the sequential segmentation of tiles.

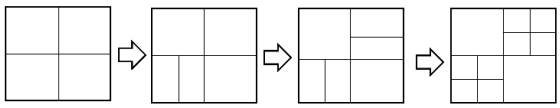

Fig. 2. Sequential segmentation of tiles

In the learning process, we utilized hash coding, which greatly reduced the amount of storage space. We also use pseudo-random algorithm to reduce the number of tiles. Hash correlation is used to form a limited tiling with non-adjacent tiles in the state space randomly. Hash coding can effectively solve the dimension disaster problem. Hashing mapping is showed in Fig. 3.

Main procedures of this algorithm are as follows: we initialize all the servers to be idle and obtain the Q value through the TileCoding function for all the sensor nodes at certain state. The weight parameters are initialized to 0 in tile. All the initial rewards are initialized to 0. The package loss rate and current server bit error rate are used to obtain information from each new sensor nodes. The initial state is chosen by the random policy. The action is chosen by Boltzmann distribution probability according to the free servers and the type of sensor at queue head. The Q value is updated by the SARSA Algorithm. Finally, through multiple sensor nodes in the queue and the bit error rate of the server which increases over time, the evaluation value gradually converges with iteration. The specific algorithm is combined with TileCoding as follows.

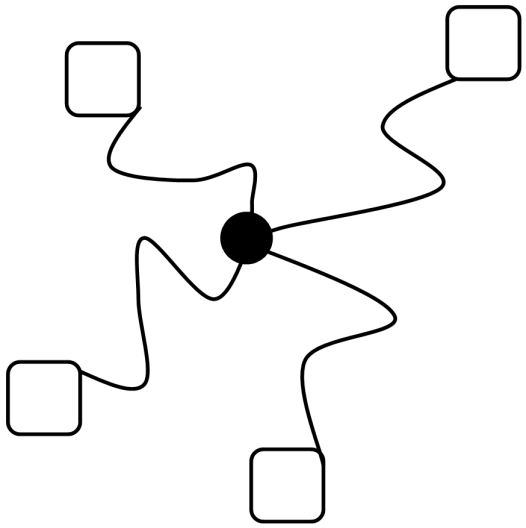

Fig. 3. The hash map of one tile

where m is the number of tilings, e_t is the bit error rate, λ/T is the packet loss rate based on the growth of fixed time period and w is an array of the weights.

The traditional access control problem only considers the condition of the visitor. It ignores the data security processing of some servers. The algorithm in this paper takes into account the package loss rate of data uploaded by sensor nodes and the bit error rate of the server compared to the traditional algorithm, which can be better applied to the actual situation.

4. Experiments and Analysis

We simulate the situation of sensor nodes and servers. The simulation environment is that there are 5 different types of sensor nodes in the queue, and their loss rates of uploaded data are 0.01%, 0.02%, 0.04%, 0.08% and 0.16%, respectively. There are 12 servers that receive data. And bit error rate increases by 3% every fifty thousand time steps. For each busy server, there is an 8% probability that the busy state will change to idle state getting ready to receive data. To facilitate comparison, we set their immediate rewards to 16, 8, 4, 2, and 1 for the sensor nodes with different packet loss rates that are randomly generated in the queue head. Immediate rewards will be multiplied by the corresponding percentages as time goes on. The purpose of this task is to explore a policy to decide whether to receive data in each time step according to the nature of sensor nodes, number of servers and the current state, so as to obtain a secure data transmission policy.

In the learning process, we used the average reward as the reward function, set accept action as 1 and reject action as 0, set step size for learning state-action value as 0.01,

Algorithm 3 SARSA based access control with approximation by TileCoding (SACT)

Input: the style of sensor node in the queue head Y,step sizes $\alpha, \beta > 0$, number of busy servers N

Output: the server system reject or accept the new sensor node

for $i = 0$ to n **do**
  Initialize the weights w of the i tile and $2k$ sub tiles e.g., $w = 0$
end for
Initialize average reward $\overline{R}$ arbitrarily(e.g.,$\overline{R} = 0$)
Initialize state $S = [N, Y]$, actions A
repeat
  Take actions A, observe state S
  Choose A' as the function of $Q(S', w)$(e.g., Boltzmann's method)
  $r(S, A) = (1 - \mu)(1 - e_t)Inf - \mu\lambda/T$
  $\delta \leftarrow r - \overline{R} + Q(s_{t+1}, a_{t+1}, w) - Q(s_t, a_t, w)$
  $\overline{R} \leftarrow \overline{R} + \beta\delta$
  $w \leftarrow w + \frac{\alpha}{m}\delta\nabla Q(s_t, a_t, w)$
  $S \leftarrow S'$
  $A \leftarrow A'$
until terminal
return (S, A)

step size for learning average reward as 0.01, and updated the parameters according to TileCoding. Therefore, in the experiment in this paper, the aim is to get a policy after experiencing 1 million time steps under different numbers of idle servers and the sensor nodes at the queue head required to upload data.

Fig. 4 shows the differential value of optimal actions for sensor nodes that produce different package loss rates in 1 million time steps. It can be observed from the figure that for the sensor nodes with different packet loss rates, the value obtained through TileCoding reinforcement learning is inversely proportional to the packet loss rate, which is consistent with intuition. It can also be seen from the figure that when the number of idle servers reaches 9, there is a huge amount of transitions under different sensor nodes. When the number of busy server is 3, the data is at its most complete and all states are traversed. In addition, the performance appears not satisfactory when the value is lower than 0. When the number of idle servers is in range only from 0 to 4, the data obtained in the simulation experiment cannot cover all cases.

Fig. 5 shows the result of the strategy after 1 million time steps where the dark color means receiving new sensor node data, and light color means refusing to accept new sensor node. On the whole, as the number of idle servers decreases, higher packet loss rate corresponds to greater chance of rejection.

More specifically, when the number of free server is 0, all sensor nodes are rejected which makes the algorithm reliable. The server system can take the corresponding policy according to the figure. In order to maximize security of data transmission, the server system can only choose the packet loss rate of type 4 and type 3 when the free server number is 1. When the number of free servers is 2 or 3, type 0 and type 1 sensor nodes cannot upload data. When the number of free servers is 11 or 12, the system still refuses type 0 or type 1 because the system prefers to keep as much security as possible and get as much reward as possible. By comparing Fig. 4 with Fig. 5, we can find that the two

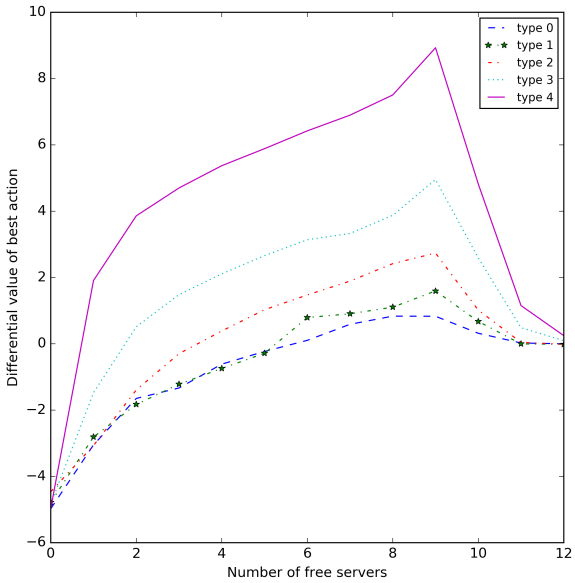

Fig. 4. The differential value of optimal actions of the five different sensor nodes that produce different package loss rates in the 1 million time steps for the different number of free servers. Type 0 to 4 of the sensor nodes denotes the rewards that are getting larger from 1 to 16.

graphs can be related. When the number of idle servers is from 0 to 3, or from 11 to 12, we find that the value is very low so most servers of these numbers are rejected accordingly. In addition, when the values are highest, the server system is in a position to receive all types of sensor nodes to upload data.

5. Conclusion

The defect of traditional access control in wireless sensor networks can lead to a large number of information omissions without adopting a long-term effective strategy. This will make the information transmission insecure. In this paper, an efficient and feasible access control policy based on TileCoding reinforcement learning algorithm is proposed, which provides us with a policy for sensor nodes in different complex situations. After taking full factors into account such as the package loss rate of the sensor node and the bit error rate of the server, the SACT algorithm adopts the method of function approximation by TileCoding and takes corresponding actions through the probability of Boltzmann distribution. We adopt the idea of average reward and simulate the credible strategy. Simulation results show that the SACT algorithm can provide a relatively safe strategy for the access control of sensor nodes.

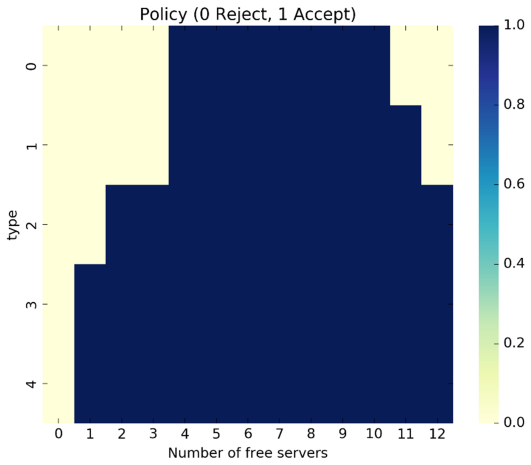

Fig. 5. The result of the strategy for the server system after 1 million time steps. Dark color means receiving new sensor node data, and light color means refusing to accept new sensor node. The actions can be chosen by the state of the number of free servers and type of sensor nodes in the figure

Acknowledgment This work was supported by National Natural Science Foundation of China (61303108, 61373094, 61702055); The Natural Science Foundation of Jiangsu Higher Education Institutions of China (17KJA520004); Suzhou Key Industries Technological Innovation-Prospective Applied Research Project (SYG201804); Program of the Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) (KJS1524).

References

1. Matthias Althoff and Goran Frehse. Combining zonotopes and support functions for efficient reachability analysis of linear systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7439–7446. IEEE, 2016.
2. J Amaré, S Cebrián, C Cuesta, E García, M Martínez, MA Oliván, Y Ortigoza, A Ortiz de Solórzano, C Pobes, J Puimedón, et al. Status of the anais dark matter project at the canfranc underground laboratory. In *Journal of Physics: Conference Series*, volume 718, page 042052. IOP Publishing, 2016.
3. Abeba Mitiku Asfaw, Mesele Damte Argaw, and Lemessa Bayissa. The impact of training and development on employee performance and effectiveness: A case study of district five administration office, bole sub-city, addis ababa, ethiopia. *Journal of Human Resource and Sustainability Studies*, 03(04):188–202, 2015.
4. Prosunjit Biswas, Ravi Sandhu, and Ram Krishnan. Label-based access control: An abac model with enumerated authorization policy. In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, pages 1–12. ACM, 2016.
5. Loic Bontemps, Van Loi Cao, James Mcdermott, and Nhienan Lekhac. Collective anomaly detection based on long short-term memory recurrent neural networks. *arXiv: Learning*, pages 141–152, 2016.

6. Yi Chu, Selahattin Kosunalp, Paul D Mitchell, David Grace, and Tim Clarke. Application of reinforcement learning to medium access control for wireless sensor networks. *Engineering Applications of Artificial Intelligence*, 46:23–32, 2015.
7. Wang Ding, Haibo Cheng, Debiao He, and Wang Ping. On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices. *IEEE Systems Journal*, PP(99):1–10, 2016.
8. Thanh-Toan Do and Ngai-Man Cheung. Embedding based on function approximation for large scale image search. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):626–638, 2018.
9. Mohammad Fathi. Reinforcement learning for multiple access control in wireless sensor networks: Review, model, and open issues. *Wireless Personal Communications*, 72(1):535–547, 2013.
10. David Ferraiolo, Ramaswamy Chandramouli, Rick Kuhn, and Vincent Hu. Extensible access control markup language (xacml) and next generation access control (ngac). In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, pages 13–24. ACM, 2016.
11. Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.
12. Shay Gueron and Vlad Krasnov. Fast prime field elliptic-curve cryptography with 256-bit primes. *Journal of Cryptographic Engineering*, 5(2):141–151, 2015.
13. Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.
14. Andreas Jacobsson, Martin Boldt, and Bengt Carlsson. A risk analysis of a smart home automation system. *Future Generation Computer Systems*, 56(C):719–733, 2016.
15. Shellie L Keast, Hyunjee Kim, Richard A Deyo, Luke Middleton, K John McConnell, Kun Zhang, Sharia M Ahmed, Nancy Nesser, and Daniel M Hartung. Effects of a prior authorization policy for extended-release/long-acting opioids on utilization and outcomes in a state medicaid program. *Addiction*, 113(9):1651–1660, 2018.
16. Imran Khan, Fatna Belqasmi, Roch Glitho, Noel Crespi, Monique Morrow, and Paul Polakos. Wireless sensor network virtualization: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):553–576, 2016.
17. Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pages 3675–3683, 2016.
18. Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4190–4203, 2017.
19. Hongjiao Li, Shan Wang, Xiuxia Tian, Weimin Wei, and Chaochao Sun. A survey of extended role-based access control in cloud computing. In *Proceedings of the 4th International Conference on Computer Engineering and Networks*, pages 821–831. Springer, 2015.
20. Yun Lin, Chao Wang, Jiaxing Wang, and Zheng Dou. A novel dynamic spectrum access framework based on reinforcement learning for cognitive radio sensor networks. *Sensors*, 16(10):1675, 2016.
21. LShuai, Lei Liu, Hong Jiang, and Wei Wei. Classical optimal planning method based on compacted encodings. *Journal of Jilin University*, 40(6):1644–1649, 2010.
22. Renquan Lu, Yong Xu, and Ridong Zhang. A new design of model predictive tracking control for networked control system under random packet loss and uncertainties. *IEEE Transactions on Industrial Electronics*, 63(11):6999–7007, 2016.
23. Polina Mamoshina, Armando Vieira, Evgeny Putin, and Alex Zhavoronkov. Applications of deep learning in biomedicine. *Molecular pharmaceutics*, 13(5):1445–1454, 2016.

24. Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Steven Latré, Marinos Charalambides, and Diego Lopez. Management and orchestration challenges in network functions virtualization. *IEEE Communications Magazine*, 54(1):98–105, 2016.
25. Amir Mukhtar, Likun Xia, and Tong Boon Tang. Vehicle detection techniques for collision avoidance systems: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2318–2338, 2015.
26. Simone Mutti, Enrico Bacis, and Stefano Paraboschi. Sesqlite: Security enhanced sqlite: Mandatory access control for android databases. In *Proceedings of the 31st Annual Computer Security Applications Conference*, pages 411–420. ACM, 2015.
27. M Naghiloo, JJ Alonso, A Romito, E Lutz, and KW Murch. Information gain and loss for a quantum maxwell's demon. *Physical review letters*, 121(3):030604, 2018.
28. Huyên Pham and Xiaoli Wei. Bellman equation and viscosity solutions for mean-field stochastic control problem. *ESAIM: Control, Optimisation and Calculus of Variations*, 24(1):437–461, 2018.
29. Qasim Mahmood Rajpoot, Christian Damsgaard Jensen, and Ram Krishnan. Attributes enhanced role-based access control model. In *International Conference on Trust and Privacy in Digital Business*, pages 3–17. Springer, 2015.
30. Hector Reyes, Sriram Subramaniam, and Naima Kaabouch. A bayesian network model of the bit error rate for cognitive radio networks. In *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*, pages 1–4. IEEE, 2015.
31. Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320, 2015.
32. S. Shahrabadi, M. Moreno, J. I. Rodrigues, J. M. Rodrigues, and J. M. Buf. Computer vision and gis for the navigation of blind persons in buildings. *Universal Access in the Information Society*, 14(1):67–80, 2015.
33. Haiying Shen and Liuhua Chen. Distributed autonomous virtual resource management in datacenters using finite-markov decision process. *IEEE/ACM Transactions on Networking*, 25(6):3836–3849, 2017.
34. Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
35. JD Ultra and Susan Pancho-Festin. A simple model of separation of duty for access control models. *Computers & Security*, 68:69–80, 2017.
36. Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Fei Zhu is a member of China Computer Federation. He is a PhD and an associate professor. His main research interests include machine learning, reinforcement learning, and bioinformatics. Email:zhufei@suda.edu.cn.

Pai Peng is a postgraduate student in the Soochow University. His main research interest is reinforcement learning. He programmed the algorithms and implemented the experiments. Email:20185227062@suda.edu.cn.

Quan Liu is a member of China Computer Federation. He is a PhD, post-doctor, professor and PhD supervisor. His main research interests include reinforcement learning, intelligence information processing and automated reasoning. Email:quanliu@suda.edu.cn.

Yuchen Fu (corresponding author) is a member of China Computer Federation. He is a PhD and professor. His research interest covers reinforcement learning, intelligence in-

formation processing, and deep Web. He is the corresponding author of this paper. Email: yuchenfu@cslg.edu.cn.

Shan Zhong got her PhD in computer science and technology. Her main interests include machine learning, artificial intelligence and application of reinforcement learning. Email:zhongs@uwec.edu.

Received: August 30, 2018; Accepted: July 6, 2019.