# Improvement in Learning Efficiency of SARSA for Wireless Communication Network

**Anil Kumar Yadav**
IES college of Technology, Bhopal
**Purushottam Sharma** ( puru.mit2002@gmail.com )
Amity University https://orcid.org/0000-0002-8037-7152

**Research Article**

**Keywords:** Reward, SARSA, Q Learning, Eligibility traces

**Posted Date:** June 13th, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-1452363/v1

# Abstract

State action reward state action (SARSA) is one of the important learning methods of reinforcement learning. It updates policy using optimal action taken over the environment and uses temporal difference ( ). Where the used for eligibility trace, it used to contain a temporary record of every node in state action pairs (s, a) under each episode. In this paper, we proposed a novel algorithm and developed the update target value applying the average of all possible successive action values along with eligibility trace to store every state action value within look up table. That improves reinforcement learning model convergence characteristics and learning efficiency. In the result section, represent that our proposed algorithm has better learning efficiency than Q learning. It can be used for solving complex wireless communication networks and other solutions for the development of other intelligent artificial communication networks.


# 1. Introduction

In nowadays artificial intelligence technology play an important role in the development of the different field; machine learning is an emerging technology towards solving practical communication problems in real environments. Increased learning is widely used in environment training and for accepting sensory input data sets under policy optimization issues. Sarsa techniques interacts sensory input and environment through reward and penalty. Reinforcing learning can learn on itself without prior knowledge of the environment compared to other machine learning methods through trials and errors. Sarsa is a TD control on-policy. Both greedy policies are Sarsa's objectives policies and behavior [1]. Estimation of the value function for taking decisions under greedy policy. It is environmentally sound and perceptive and has online learning features. Increased learning therefore offers a viable and effective solution to the sequential problem. In complex environments, decision-making. Q learning acting over an unknown environment without prior knowledge of the system environmental information is widely used in model-free reinforcement learning. Without prior environmental facts, interact with environmental effects and carry out the test and best possible greedy policy for movement. Increasing discount along with regard to intelligent machines [2-5]. The framework for enhancement learning in the field of communication has also recently been widely studied resource allocation and network programming.

### 1.1 Models for the Environment

The environment maintains the current internal state and describes the internal transitions of the agent when an action is performed. It also determines if an episode has concluded that state transitions can be properly described. It is the state of the agent using the environmental model fig.1.1. It provides functions to get the current state into a state object, to execute an action and to determine if the episode is over. It includes the following methods, calculates the internal state transitions or carries out an action and measures the new state for this functionality. Indicate that after the current step since the episode has ended the model reset [6-10]. In the next state, you must write the agent to find current and internal state variables.

Function transition describes the transition function used for the internal state transitions as the interface. The function transition function is shown by $s_0 = f(s, a)$. There are also functions to get an initial state for a new episode and to determine whether a model should be reset in a particular state. The following interface methods are available to the user for this functionality. Reset State sets initial episode states. This may be random states or certain states. Some initial methods for the sampling state are already described, such as random or zero initialization [11-13]. If the episode in the state is unsuccessful, return to the beginning state. This transition function keeps the current agent condition and uses the transition function for state transitions.

The model of action is to be executed in more than one-step, and therefore the number of steps needed should be kept. These steps should not be fixed, but may depend on the current condition. The action must store the action values used for a continuous space of action. Continuing actions with discrete actions such as robotic football, navigation, and shooting should also be intermixed. Another action contains primitive actions, which have been decided on the following action model; discreet actions from an action set do not contain any information.

An action object is always only created once and a search criterion in action set fig.1.1 is the action object pointer. Every action type with modifiable data has a particular action data object while the action values are stored in continuous measures. In the action's action data, the current activity is stored. Yet another problem comes with this approach. For instance, what happens if an algorithm wants to change the action data by setting other constant actions? All other listeners will receive these falsified actions, because all listeners receive the same action when the action information is changed. Therefore, you really need to change the action data at least after you use it on all listeners [14-17]. For all methods that receive action, it therefore introduces additional action data parameters. A listener must not modify action data, but must use his/her individual action data. The data on the action changed only by the agent himself and is always the action taken in the current step. All actions offer a general interface for the action data object to return if no action data are used and the action data of the appropriate type created. Furthermore, all action data provides functions to set and copy actions by another action data object. For actions that are not fixed, because the duration of the variable has to be stored in action information. The action details include the following:

• Number of steps already executed by the action.

• Whether the action is completed as it is. Multi-stage actions ($s_t$, $s_{t+1}$) are represented. This approach offers two ways to use multi-step actions [18-20].

• Duration of the environment model may be specified. This is useful in robotics, for example, where the exact duration of a specific action cannot be known before performance. The duration can be measured after execution and then saved to the multi-stage data. Controller of agent defined in fig. 1.1 this requires an interface and the used learning algorithms for controllers. It introduces a single agent controller object that the user can set. The controller cannot alter only the agent's action data, so how can changing action

information from the controller be returned. One action data set is a part of an action set, whereby a new corresponding action data is stored for each action in the action set. If the agent wants to retrieve an action from a controller, he always gives the controller additional action data [21]. The controller now selects an action specific, changes the action data assigned to the action and returns the pointer of the action chosen. The agent has to obtain this from his data set. After the action recovered from the agent, the agent changes the content of the current action data to the controller's actions data. State representation used as one of the most important steps in the learning of problems since it describes a strong environmental model. In order that the various formulations avoid misunderstandings:

• State: all that is the object of the State.

• State variable: the variable of a single state so that, for example, the variable of a continuous state could be the agent location x.

• State object obtained from the environment, the internal state of the agent.

For example, the ongoing model state may suffer discrimination. There are an arbitrary number of continuous and discrete state variables for general enhancement learning tasks. These state variables are collected in one state by State. The state properties are the number of discrete and continuous status variables maintained by a state object. It also stores the discrete state sizes and the valid range of continuous state variables for discrete State variables. Additional information on whether the variable is regular or not, e.g., for angles, can be given for continuous state variables. The status properties are created by the environment where the user needs to either specify the exact properties or by the State modifier for modified states [22]. All state objects, which describe the same state, hold a pointer to the specific status. The state of the environment should usually contain all information about the internal situation of the agent. The state does not need to discriminate against the constant state variables, because this type of information is stored elsewhere. Training tests to trace the policy learned or use the stored pathways for education. It stores both the states and actions and the value of the award. Learning data in robotics, for example, is very difficult to collect and use for the whole training trials to recreate with other algorithm parameters. Only off-position learning can be used for the stored episodes. Off-political learning often leads to poor performance, but can be used before real learning begins. A list of states is necessary to store a whole episode in your memory, as a state has to be dynamically allocated each time. Episode is referred to as a learning process, one episode can be saved at a time in memory. They're designed as listeners already. Once the new episode starts, the episode object rejects all stored data. It is therefore possible to specify which states to store. There are states and measures to be stored in an episode.

The learning process defined as an agent can store several episodes. It interfaces to obtain the agent's data. An episode list is kept by the agent. The single states can be found from the episodes. The episodes should be remembered as numbers. When the entire study is stored at a table. The learning Data in state-action pairing or Q-tables must be stored in each RL algorithm. There are features for such learned information, such like storing or loading the learned data. It determines that a policy is either good or bad.

It can estimate a certain number of states' future discounted award or average award in certain episodes through learning experience. The number of episodes for evaluation may be used by the Policy Evaluator. The initial status of episodes is sampled by the environment as usual. If the initial states are sampled at random, large initial state spaces will require a large number of episodes specially to achieve reliable results. The same set of initial states can be used for each assessment every time.

### 1.2 Wireless Communication System

In wireless sensor network dynamic environment mechanism is more challenging than another environment to solve a particular. Innocent routing protocols upgrade and setups various sensor base station which are used for transmitting and receiving data in both mode weather static and dynamic. Here, especially described dynamic base station, which operated and controlling packets from various node along with shortest path and also sure that it regulates within each node. so that every node smoothly transmitted data through this path continuously. Another episode may be static that based on various application sensing nodes. Object detection and tracking nodes through sensing devices in short of time responses, for clear explanation, is applicable in the early-stage monitoring of forests to protect and avoid burning and other accidental causes in that places. Static monitoring phenomenon based on the reactive manner along with constructive mode [23-25].

Data transmission has more applicable in the field of wireless sensor network. It is working on continuous event process, structure-based analysis and hybrid in nature. Each sensor transmits data packets in continuously to the nearest base station. That are very challenging task to design schematic diagram of network architecture. There are more than hundred sensors are connected together and operated over a given target-oriented task. The number of nodes governed by routing protocols that deals which protocol are sending to which nodes that is more important for used protocol. Data Fusion – decomposition of the transmitting data for sending to the various nodes. Method of integrating data from various sources based on a set of criteria. Signal amplification techniques are used to do this. Some routing protocols use this strategy to improve energy efficiency and data transfer. In this method, coping data are decomposed based on unique key, which received from various nodes. Wireless sensors fly around the mobile sync sensing section from the receiving nodes, the sensor node using the mobile sync node architecture. A mobile sync node transforms it into a sensor node. It may also make use of a data collector that is mounted within the sensing region. The following approaches used to collect data in wireless sensor networks with handheld sink nodes. Discovery – Information base independent of mobility, data transfer –collaborative data discovery and transfer, proxy-based, flat routing and motion control – trajectory – static, dynamic, speed, and hybrid

Sensor Node- System that collects data from every connected sensor node over the environment, processes it, and communicates with other nodes. Received all sensing data after synchronization it sent to respective nodes. It's used for a variety of purposes, including data collection and information gathering. It functions similarly to a base station or access point.

**Topologies of the Network:** Working principle of the bus topology is described through broadcasting many nodes connected to each other over network. It may handle traffic congestion and establish communication through the one-to-one communication. When a network bus has more than a few hundred nodes, performance issues are likely to arise.

Topology of Trees- In this case, the tree network can be thought of as a cross between the two stars [26-27]. A wireless sensor network route may have a single hop or several hops, each of which is a suitable sensor node for the sensor to receive and exchange environmental data. They synchronize and send the sensor to their parents after receiving the data message from their children. In this, tree was defined through load balancing and also established communication between each node, has a flaw. Topology of the Stars- A centralized coordination center connects star networks (sync). Nodes are unable to interact with one another directly[28]. The entire communication must go through a single point of contact. The highest remaining battery capacity, the shortest various stages, and the lowest movement consignment are all factors in determining the finest route from initial to target. We use same routing parameters in two separate topographic regions with the A-Star search algorithm and fuzzy approach to compare our point of view on the efficacy of the network's better utilization of the consumption of the energy [29].

For communication- purposes, each node in a ring network has exactly two neighbors. In a loop, all messages move in the same direction, clockwise or counterclockwise.

When a node fails, the loop is broken, and the whole network is brought down to a lower level. The ring network, on the other hand, effectively manages traffic and dual-path link congestion.

Topology of Meshes- Messages in the mesh topology will travel in multiple directions from their origin to their destination. (Remember that, even though two paths exist in a loop, the message can only move in one direction.) A complete network is one in which every node is connected to every other node. This consists of some devices that are connected to others through nodes.

# 2. Proposed Model And Algorithm

SARSA (status-action-reward–state-action) learning consists of learning and storing state and actions in the context of a greedy policies ($\epsilon$). Both SARSA and Q Learning are usually suitable in the Marko decision-making process to deal with most of the tabular cases. This enables creation based on greedy policy every iterative data saved in to look up table under restricted storage space and computing power. Sarsa and query-based learning are systematic, complete online teaching methods using temporal differentiation (TD). Beginning of the state action pairs, the agent takes a state action pair and performs every time the episode. Estimate the action value will be updated by state action-pairs along with greedy policy- $Q\pi(s, a) = + \gamma(x_1^{-i}) \sum \pi a' Q_t(s_{t+1}, a_{t+1})$…..(2), to reduce the calculation burden. (1) (2). State action pair Qt(st + 1, at + 1) of the above-mentioned Eqs. (1) and (2), and each update needs to only achieve one-step action pair. Where state current (st), action (at), state st + 1 successively, and action on + 1 follow the immediate Reward Rt + 1[34–35]. To reduce the computational burden, along with greedy policy- $Q\pi(s, a)$

= $\gamma(^{x}{}_{1}{}^{-i}) \sum \pi a'$ $Q_t$ ($s_{t+1}$, $a_{t+1}$).....(2), the above-mentioned Eqs. (1) and (2) state action pair $Q_t(s_{t+1}, a_{t+1})$, action pair. Where current state ($s_t$), action ($a_t$), successive state $s_{t+1\ and}$ successive action $a_{t+1\ and}$ the immediate reward $R_{t+1}$.

In the above figure-2, γ is the discount rate and their range (0 < γ < 1), which reduces the effect of future expected rewards. N is counted step, i is total step and R is total reward .

This algorithm can search for an optimistic path without repetition in an unknown environment with the help of WSN. The purpose of an algorithm is to traverse all the states firstly with the help of look-up-table which contain approximate value of the number of further paths possible along a particular path. The algorithm accepts some pseudo code as the input for training process of agent. Complexity of time and space is deliberate through the algorithm. In which sarsa agent moved in all possible directions and arrived through independent paths from the start state to the goal state.

This paper assumes episodic tasks with a termination state, and the proposed algorithm is given below-

The complete proposed algorithms in this work are as follows-

### 2.1 Proposed algorithm and implementation

1. Parameter used in algorithm: eligibility trace discount rate λ ∈ (0, 1), learning rate α > 0
2. Initialize look-up-table Q (s, a) and eligibility trace E (s, a) for all s ∈ S, a ∈ A
3. define loop for every epoch

Begin

For (every state $s_i$: S)

Initialize E (s, a), for all s ∈ S, a ∈ A

fetch the start state $s_i$

find possible actions and execute

check for goal status

if (yes: goal = start state)

Update discount rate $R1 = R\gamma^{N-i}$ and $i = i + 1$

E (s, a) ← E (s, a) + 1 (accumulating trace)

else

{

Goal ≠ stat states

Store (s,a) value in look-up-table

}

repeat until goal reached

End.

# 3. Results And Discussions

In this part, we discussed about on-policy and off-policy temporal difference learning. Further we use Markov decision process with 16 states. The state space can be visualized using a 4x4 grid along with SARSA learning. As shown in Fig. 3 represents the state of the grid world environment. The actions performed by the agent over given environment via every cell along with four actions possible in each state: north, south, east, and west. Due to the boundary wall problem agent is not able to jump from one boundary to next state. After movement, the agent receives a reward of -1 on each transition. According to the received signal agent will perform action immediately to backtrack to the current position in grid world. It received reward in + 1. In this, randomness of decision-making agent is not allowed between cell.

The value function for the random policy is shown in Fig. 3. For each state the random policy randomly chooses one of the four possible actions. The numbers in the states represent the expected values of the states.

The optimal value function is shown in Fig. 3. Again, starting in the lower left corner, calculating the sum of the reinforcements when performing the optimal policy (the policy that will maximize the sum of the reinforcements).

### 3.1 Performance Comparison

SARSA and Q learning Performance are used by the following table and graph to evaluate accuracy, learning efficiency with respect to various episode.

Table 3.1
SARSA Learning and Q Learning Algorithm comparison table between different episodes

| Learning rate/discount rate | For Epoch E = 1000 | For Epoch E = 3000 | For Epoch E = 5000 | For Epoch E = 7000 | For Epoch E = 9000 | For Epoch E = 11000 |
|---|---|---|---|---|---|---|
| d1 | 71 | 95.80 | 83 | 97.79 | 83 | 98.70 |
| d2 | 72.40 | 71 | 71 | 85 | 95.82 | 96.82 |
| d3 | 73.50 | 68.70 | 96.76 | 95.66 | 90.75 | 84.61 |
| d4 | 89 | 78.70 | 96.52 | 85.80 | 98.91 | 95.71 |
| d5 | 97.99 | 95.82 | 79.11 | 86.86 | 98.81 | 94.56 |
| Average | 78.87% | 83.50% | 80.69% | 92.59% | 91.70% | 97.30% |

### 3.2 Comparison between SARSA and Q Learning reward with various episodes

In this section, we compare the performance between SARSA and Q learning algorithm in the grid world problem in the context of reward with respect to the different episodes and episodes with time steps. In which we are used various Parameters as γ =0.99, is discount rate, α = 0.02, learning rate under number of episodes.

We can find that sarsa learning algorithm increase to converge, while the performance of Q learning, which indicates the total reward of an episode is less than sarsa learning.  Sarsa learns secure path, in which learning agent will move towards the target on path. Penalty in term of wrong probability assigned by agent during negative feedback of the grid world problem.  Therefore, in the grid world problem, sarsa is better than Q learning.

### 3.2 Comparison between SARSA and Q Learning algorithm learning efficiency with different episodes

In this section, we discussed the several discount rates on each sarsa and Q learning along with number of episodes. We focused the agent to learn the optimal learning algorithm faster and more efficiently to attain better performance. Therefore, first of all we have taken 60 episodes for agent performance under ten runs for each learning rate. Range of given learning rate is from 0.1 to 0.9 and it increased in step by 0.1 and the eligibility trace of sarsa is set to 0.4. In figure 3.1 shows the total rewards over 60 episodes under ten runs. We observed that the performance of sarsa learning increased exponentially as compared to Q learning in the context of the learning rate. We also observed that expected sarsa along with eligibility trace given better performance than other algorithms. Eligibility trace enhanced learning efficiency and reduced the number of episodes so that proposed algorithms find short routes within the communication network along with faster discount speed. Increased learning efficiency performed by the agent through all possible actions, the proposed algorithm can also perform well learning rate. SARSA

learning uses max possible action value of succeeding actions for route finding in the grid world environment.

# 4. Conclusion

We have studied the different intelligent target detection problems in communication networks scenario along with their learning techniques. In most of the algorithms used, Q-learning RL algorithm to find a routing strategy. Different from others, we proposed a novel algorithm based on sarsa learning used to the updates target value, applying the average of all possible successive state action value Q(s, a) along with eligibility trace to store every state action value within lookup table. This algorithm is able to reduce the number of episodes during policy decisions for taking action over a given environment, which increases learning efficiency.

Based on performance comparison, we have found that our proposed algorithm has better learning efficiency than Q learning in the grid world environment. And this also increased then learning efficiency and reduced the number of episodes during decision-making in the grid world problem. It also provides better solution to study and design of distributed wireless communication networks. In the future work, we will extend to design along with optimization of route selection in distributed IOT systems, bandwidth allocation and power control optimization in wireless communication systems. The further expansion of RL, as deep reinforcement learning and multi-agent methods.

# Declarations

### Data availability

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

### Compliance with Ethical Standards

### Disclosure of potential conflicts of interest

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

### Research involving Human Participants and/or Animals

This is an observational study. This research includes No involvement of Human and Animals, so no ethical approval is required.

### Funding:

The authors did not receive support from any organization for the submitted work.

### Informed Consent

The studies are conducted on already available data for which consent not required.

# References


1. Rigas, E. S., Ramchurn, S. D., & Bassiliades, N. (Aug 2015). Managing electric vehicles in the smart grid using artificial intelligence: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 1619–1635
2. Wang, H., Lei, Z., Zhang, X., Peng, J., & Jiang, H., "Multi objective reinforcement learning-based intelligent approach for optimization of activation rules in automatic generation control,"IEEE Access, vol. 7, pp. 17 480–17 492, 2019.
3. Huang, H., Song, Y., Yang, J., Gui, G., & Adachi, F. (March 2019). Deep-learning-based millimeter-wave massive for hybrid precoding. *IEEE Transactions on Vehicular Technology*, 68(3), 3027–3032
4. Zhao, Y., Chen, Q., Cao, W., Yang, J., Xiong, J., & Gui, G. (2019). Deep learning for risk detection and trajectory tracking at construction sites. *IEEE Access*, 7(912), 905–930
5. Stulp, F., Buchli, J., Ellmer, A., Mistry, M., Theodorou, E. A., & Schaal, S. (Dec 2012). Model-free reinforcement learning of impedance control in stochastic environments. *IEEE Transactions on Autonomous Mental Development*, 4(4), 330–341
6. Kusy, M., & Zajdel, R. (2015). "Application of reinforcement learning algorithms for the adaptive computation of the smoothing parameter for probabilistic neural network," IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 9, pp. 2163–2175, Sep.
7. Wang, Y., Liu, M., Yang, J., & Gui, G. (April 2019). Data-driven deep learning for automatic modulation recognition in cognitive radios. *IEEE Transactions on Vehicular Technology*, 68(4), 4074–4077
8. Liu, M., Song, T., & Gui, G. (April 2019). Deep cognitive perspective: Resource allocation for noma-based heterogeneous IOT with imperfect sic. *IEEE Internet of Things Journal*, 6(2), 2885–2894
9. Sim, G. H., Klos, S., Asadi, A., Klein, A., & Hollick, M. (Dec 2018). An online context-aware machine learning algorithm for 5g mmwave vehicular communications. *IEEE/ACM Transactions on Networking*, 26(6), 2487–2500
10. Kong, P., & Tham, C. (Oct 2010). Distributed reinforcement learning frameworks for cooperative retransmission in wireless networks. *IEEE Transactions on Vehicular Technology*, 59(8), 4157–4162
11. Shams, F., Bacci, G., & Luise, M. (March 2015). Energy-efficient power control for multiple-relay cooperative networks using q-learning. *IEEE Transactions on Wireless Communications*, 14(3), 1567–1580
12. Venkatraman, P., Hamdaoui, B., & Guizani, M. (July 2010). Opportunistic bandwidth sharing through reinforcement learning. *IEEE Transactions on Vehicular Technology*, 59(6), 3148–3153
13. Tathe, P. K., & Sharma, M. (2018). "Dynamic actor-critic: Reinforcement learning based radio resource scheduling for lte-advanced," in 2018 Fourth International Conference on Computing Communication

Control and Automation (ICCUBEA), pp. 1–4,

14. Yu, F. R., Song, M., & Han, Z. (Jan 2018). User scheduling and resource allocation in hetnets with hybrid energy supply: An actor-critic reinforcement learning approach. *IEEE Transactions on Wireless Communications*, 17(1), 680–692
15. Kim, D., Aceves, J. J. G. L., Obraczka, Cano, J. C., & Manzoni (2002). "Power-aware routing based on the energy drain rate for mobile ad-hoc networks", 11th International Conference on Computer Communications and Networks,
16. NS. (2004). "*The UCB/LBNL/VINT Network Simulator (*. NS)"
17. Kulik, J., & Balakrishnan, H. (1999). "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", In Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking (MOBICOM), Seattle, WA, USA, pp. 174–185,
18. Heinzelman, W. R. (2000). *Application-Specific Protocol Architectures for Wireless Networks*. Massachusetts Institute of Technology
19. OssamaYounis, & Fahmy, S. (2002). "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-efficient Approach", September
20. OssamaYounis and Sonia Fahmy" Heed. (2004). A hybrid, Energy-efficient, Distributed Clustering Approach for Ad-hoc Networks". *IEEE Transactions on Mobile Computing*, 3(4), 366–369
21. Mangeshkar, A., & Agrawal, D. P. (2001). "TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks", in the Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA,
22. Lindsey, S., & Raghavendra, C. S. (2002). "PEGASIS: Power-efficient Gathering in Sensor Information System", Proceedings IEEE Aerospace Conference, vol. 3, Big Sky, MT, pp. 1125–1130,
23. Taheri, H., Neamatollahi, P., Younis, O. M., Naghibzadeh, S., & Yaghmaee, M. H. (2012). "An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic Ad Hoc Netw", pp.1469–1481,
24. Devika, R., & Santhi, B. (2013). T. Sivasubramanian "Survey on routing protocol in wireless sensor network", pp. 350–356,
25. Lin, H., & Chen, P. (2014). L. "WangFan-shaped clustering for large-scale sensor networks",Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. pp.361–364,
26. Chand, S., Kumar, R., Kumar, B., Singh, S., & Malik, A. (2016). "Novel energy-efficient clustering protocol for prolonging lifetime of WSNs", pp.151–157,
27. Arjunan, S., & Pothula, S. (2016). "A survey on unequal clustering protocols in Wireless Sensor Networks",
28. Yadav, A. K., Sharma, P., & Yadav, R. K. (2021). "A novel algorithm for wireless sensor network routing protocols based on reinforcement learning,"International Journal of System Assurance Engineering and Management, pp.1–7,

29. Hao, & Jiang (2019). Renji Gui “An Improved Sarsa Reinforcement Learning Algorithm for Wireless Communication Systems,” artificial intelligence for physical layer wirless communication, pp. 5418–5427,

# Figures

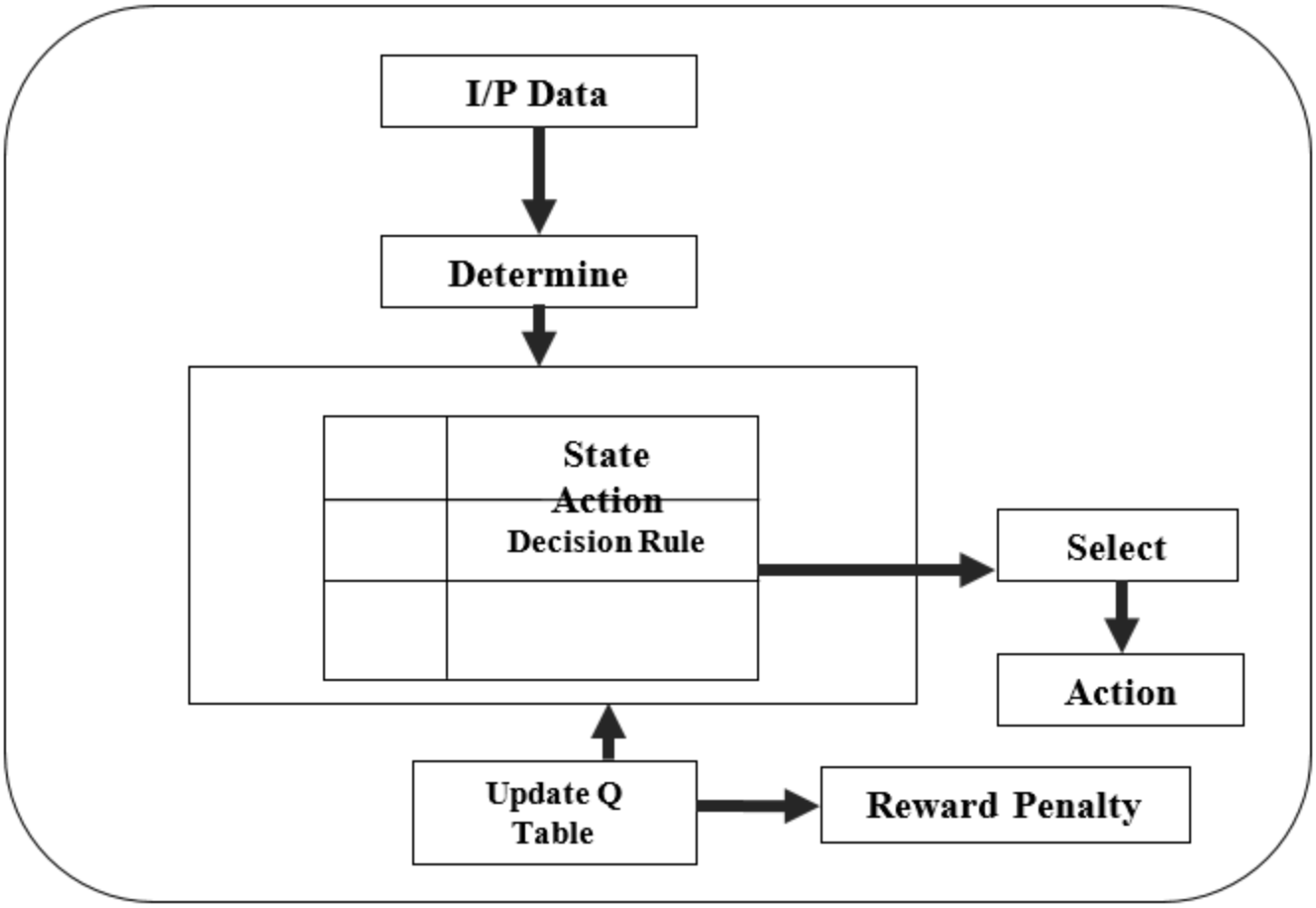
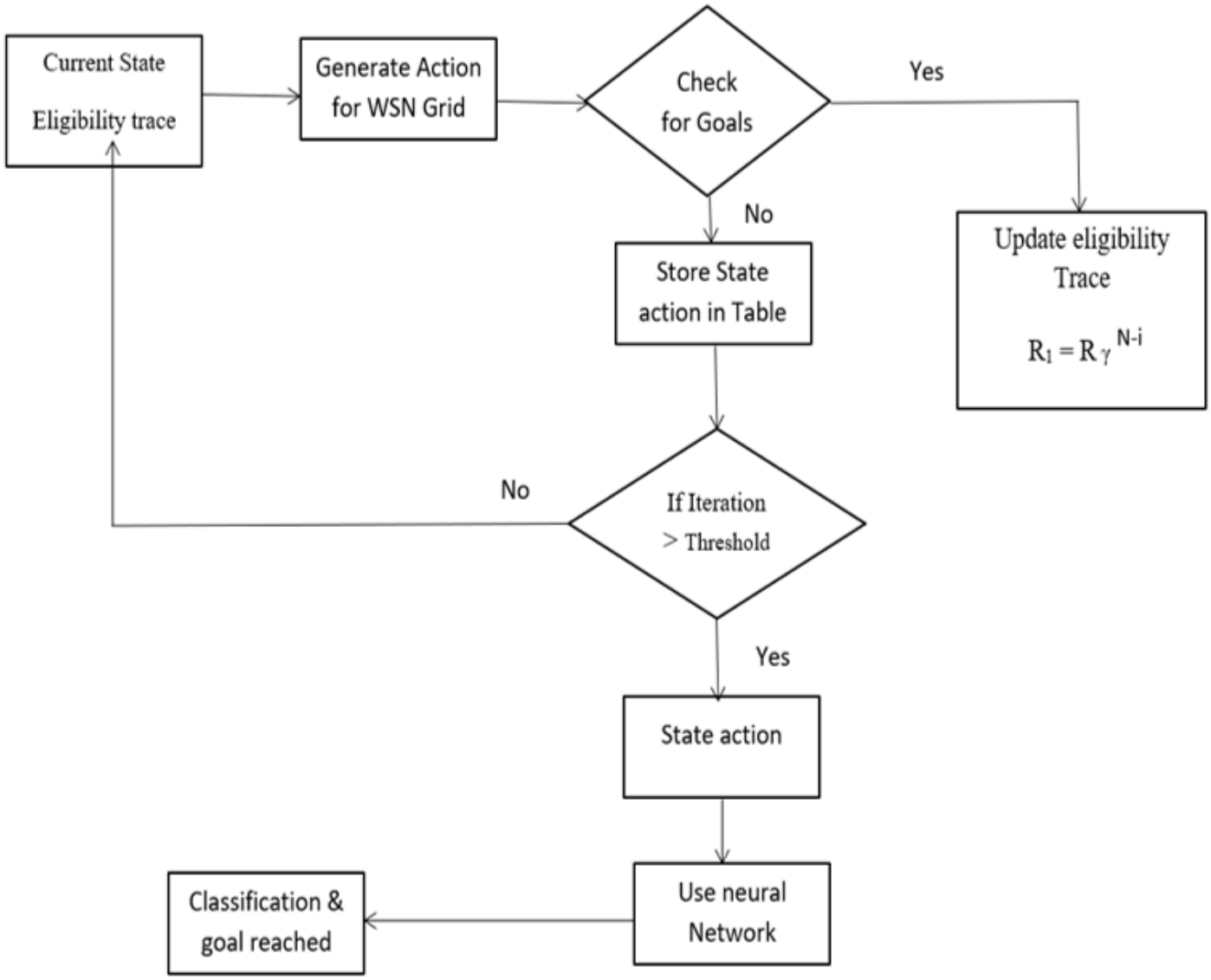



**Figure 1**

Figure 1.1 Agent-environment interaction with SARSA learning




## Figure 2

Figure 2: Proposed model of combining SARSA Learning and WSN

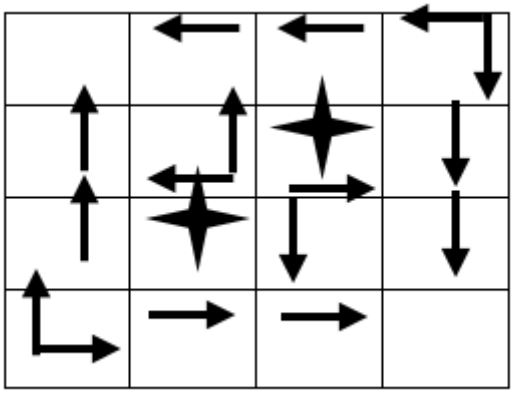


**Figure 3**

Fig.3 agent actions

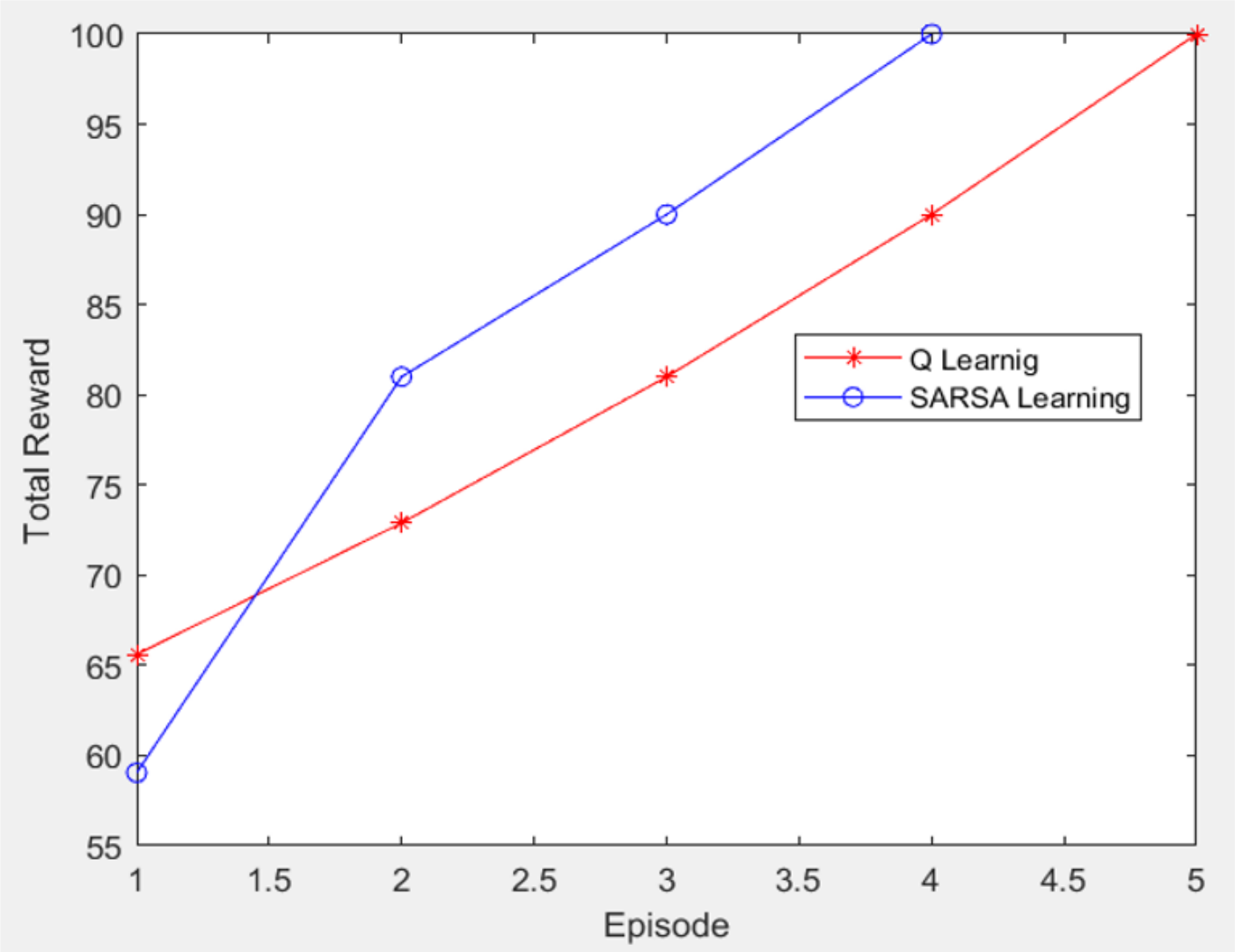



**Figure 4**

Fig 3.1. Total reward of Q Learning and SARSA Learning with respect to various episodes

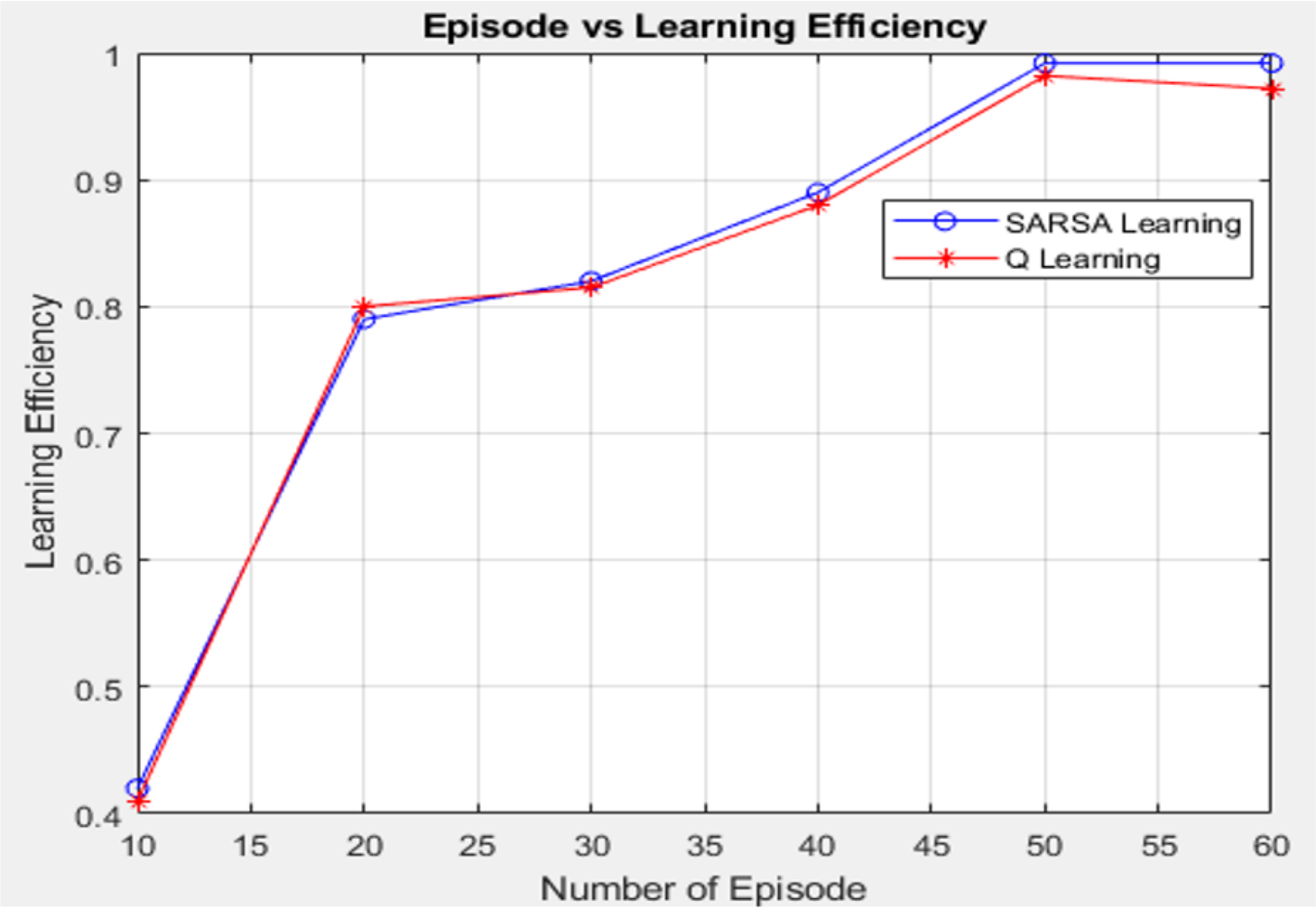



**Figure 5**

Fig.3.2 Comparison between different episodes and learning efficiency