Sparse Distributed Memory Approach for Reinforcement Learning Driven Efficient Routing in Mobile Wireless Network System

Varshini Vidyadhar[1], Dr. Nagaraj R[2], Dr. G Sudha[3]
Research Scholar, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bangalore, India[1]
Professor, Department of Information Science and Engineering, Bangalore Institute of Technology, Bangalore, India[2]
Associate Professor, Department of Electrical and Electronics, Bangalore Institute of Technology, Bangalore, India[3]

***Abstract*—In recent years, researchers have explored the applicability of Q-learning, a model-free reinforcement learning technology towards designing QoS-aware, resource-efficiency, and reliable routing techniquesin a dynamically changing network environment. However, Q-learning is based on tabular representation to characterize learned policies that frequently encounter a dimension disaster problem when introduced to the uncertain and dynamically changing network environment. In addition, the time required for agent learning in the training phase is too long, which makes it difficult for the agent to generalize the observation state efficiently. To this end, this paper attempts to overcome the overhead memory problems encountered in Q-learning-based routing techniques. In this paper, the study presents a novel memory-efficient intelligent routing mechanism based on adaptive Kanerva coding, which minimizes the storage cost required for storing large action and a state value. Unlike existing schemes, the proposed method optimizes memory requirements. Also, it enables better generalization by storing the learnable parameters of the function approximator present in the agent in a Kanerva-coding data structure. The Kanerva-coding is a sparse memory with distributed reading and writing mechanism which enables optimal compression and state abstractions for learning with fewer parameterized components making it highly memory efficient. The design and implementation of the proposed technique are done on the Anaconda tool. Simulation results demonstrate that the proposed technique can adaptively adjust the routing policy according to the varying network environment to meet the transmission requirements of different services with low memory requirements.**

***Keywords*—Mobile wireless network; reinforcement learning; Q-learning; Kanerva coding; routing; memory optimization**

I. INTRODUCTION

A. Background

A mobile wireless network can be regarded as a transient system that is inherently dynamic, decentralized, and formed via randomly deployed several wireless and mobile communicating sensor nodes to perform the distribution of the sensory information to the end node [1]. The ad-hoc feature in this transient system ensures fast and cost-effective network deployment. The sensory nodes operate as a router by receiving and forwarding the traffic of their nearby sensor nodes [2]. The salient features of mobile wireless networks are multi-hop communication, dynamic topology, bandwidth, and resources constraints. Interruption due to uncertain and dynamic topology changes affects the efficiency of the node resources. It also compromises the transmission of data packets from the source to the end node. In this regard, efficient routing in wireless networks has been extensively studied in the literature [3-5]. Therefore, various routing mechanisms have been introduced, mainly divided into reactive, proactive, and location-based routing protocols. The routing scheme of proactive type is a table-driven approach where information regarding the entire network topology is maintained at each sensor node. However, updating the table introduces a huge overhead problem due to the large control traffic in the dynamic network. In the reactive routing mechanism, the route discovery executes on on-demand [6]. However, it requires collecting adjacent information, which is a costly procedure, and, in many instances, it may not be able to determine the end-to-end path. In location-based routing, the selection of the next-hop nodes is carried based on the predefined parameters but not suitable to dynamic networks as it has restricted adaptability. Although the routing protocol of these types is advantageous in many specific situations, it has several limitations when introduced to the dynamic networking scenario [7-8].

Recently, machine learning (ML) has been widely employed to solve network problems. Incorporating the potential of machine learning technology in routing mechanisms helps to optimize network resources. In general, there are three particular types of ML techniques viz. supervised, unsupervised, and reinforcement learning. In supervised learning (SL), both input and output variables are required to train the models [9]. In un-supervised learning (UL), the model learns explicit features and generalizes the data category with only input variables. Reinforcement learning (RL) is the agent and environment interaction mechanism that enables a system to automatically explore, learn, through a trial-and-error process. However, RL is more suitable and dominant in literature when focusing on routing problems because it does not require any dataset like other ML models such as SL and UL [10].

B. Reinforcement Learning

The Reinforcement Learning (RL) technique is a specific type of ML method that comprises agent function and its interaction with the environment. RL has illustrated great

potential in various decision-making processes, autonomous systems, telecommunication systems, robotics, and recommender systems. Fig. 1 represents a typical process of agent and environment interaction.

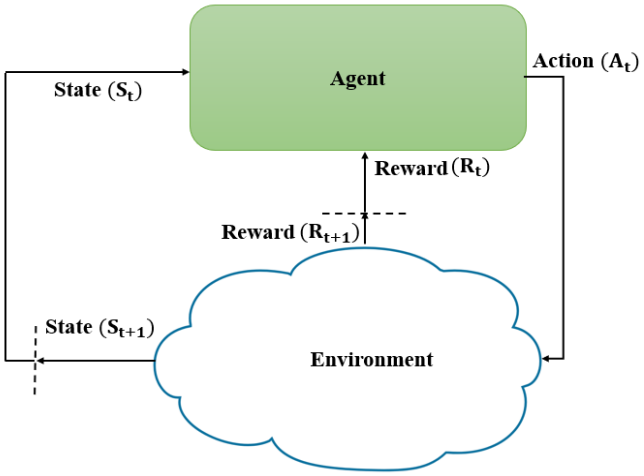

Fig. 1. Typical Function of Agent and Environment.

In typical RL, the principal of the agent interaction with the environment adopts a mathematical framework of the discrete-time stochastic control process, which is represented as a tuple such that: $\{S_t, A_t, F, \rho, \gamma\}$, where S_t refers to a state, A_t denotes action, F refers to feedback, i.e., reward (R_t) or penalty provided by the environment, ρ refers to the state transition such that: $(S_t \times A_t \times S_t) \epsilon [0,1]$ and γ discount factor concerned with rewards. The modeling of RL is concerned with episodes,i.e., set of timesteps during which the agent performs the A_t and interacts with an environment by learning a policy (π_θ) determined based on the current state. Then the agent gets an immediate R_t from the environment based on its A_t taken and transfer to the next state S_{t+1}. In this regard, the A_t can be represented as follows: $A_t = \pi_\theta(S_t)$. The ultimate purpose of the agent is to determine a most suitable π° to maximize the discounted and sum of rewards received so far from any given state such that: $\pi^\circ = \max_{\pi_\theta} V^{\pi_\theta}(S_t)$, where $V^{\pi_\theta}(S_t)$ refers to the value function of π_θ with an input argument S_t numerically expressed as $V^{\pi_\theta}(S_t) = E_{\pi_\theta}\left[\sum_{j=0}^{\infty} \gamma^j R_{t+k} | S_t \epsilon S\right]$ indicating the discounted progressive R_t achieved from S_t based on π_θ. However, the value of π° is determined using Q-function such that: $\pi^\circ = \max_{\pi_\theta} Q(S_t, A_t)$ after performing A inany given S numerically expressed as follows:

$$Q(S_t, A_t) = E_{\pi_\theta}\left[\sum_{j=0}^{\infty} \gamma^j R_{t+k} | S_t \epsilon S, A_t \epsilon A\right] \quad (1)$$

Where, A denotes action space and S denotes state space.

C. Motivation

Many researchers have explored the effectiveness of the RL in network problems. The literature has shown that that the RL-driven schemes perform well in a specific context. However, it suffers from huge overhead and performance issues in the context of dynamic networking scenarios where topology changes uncertainly and dynamically due to the mobility of sensor nodes. Although, the Q-learning and its customized variant have been widely employed for designing routing schemes to improve the data packet transmission performance and resource efficiency. However, the issue with Q-learning is that it is not able to determine the optimal solution for the path selection in an appropriate time in the complex and dynamic large-scale networking scenario. Basically, in the large network, the state and action spaces are large, and since the Q-learning utilizes table-lookup mechanisms, it is usually subjected to the issue of dimensional disaster. In the real-time scenario, the network topology changes dynamically, and accordingly, the size of the network also changes. Therefore, an infinite process in the actual sense means that the mobile sensor nodes often leave and join the network dynamically. In this context, when there are more sensor nodes in the network, there will be a large state and action space, and the Q-table occupies a lot of memory. In this regard, the dimension of the state space increases exponentially with state variables, resulting in a proportional upsurge in the dimension of Q-table required to store the value of taking action in a state based on the current policy. Also, the agent requires a large time to explore the environment to learn the policy. Therefore, in a dynamic networking scenario like MANET, computing all possible states is challenging and impractical. However, few researchers suggested integrating deep learning with strong adaptability and generalization ability to solve many practical problems. However, some challenges still remain, which motivates us to introduce an effective solution regarding memory optimization without much affecting the routing performance and network resources.

Therefore, this paper introduces a unique modelling of the reinforcement learning driven routing technique that utilizes Knerva coding mechanims in the agent modelling, which enables abstraction in the action policy learning towards exploration of optimal routuing in the dynamic network. Another significant aspect of the proposed work is the usage of customized environment designed using Open AI Gym function Employing sparse distributed scheme in the routing design clould efficiently optimizes the memory requirement and offers a better routing policy according to the varying network environment to meet the transmission requirements of different services.

The remaining section of this paper is organized as follows: Section II presents the related work and highlights the problem statement for the proposed work. Section III discusses the proposed system followed by its design and methodology. Section IV presents the environment modelling for the agent interaction to explore optimal route; Section V presents agent modelling for routing using Kanerva coding; Section VI presents the experimental evaluation and performance analysis of the proposed algorithm. Finally, Section VII concludes overall contribution of this paper.

II. Related Work

In the literature, the application of the RL techniques has been widely employed to address the limitations of traditional approaches to the networking domain. However, the existing routing protocols based on RL can be classified into three different categories, viz. i) context-specific criteria, ii) design-

specific criteria, and iii) performance-specific criteria. In the context-specific criteria, the RL addresses networking problems such as related issues, such as routing, channel selection QoS, and resource optimization. The work carried out by Saleem et al. [11] implemented RL to address the problem associated with channel selection and routing in the cognitive radio network (CRN). In this study, the authors have designed a model-based intelligent system to optimize routing and QoS in the context of the cluster-based and packet delivery ratio (PDR), respectively. In Debowski et al. [12], Q-learning-based path selection mechanisms are developed to optimize the node resources. The work of Jung et al. [13] presented a data packet-driven efficient routing scheme in the un-manned robotic network, a kind of mobile ad-hoc network (MANET). In this study, a modified Q-learning is adopted to formula tea location-based routing technique considering the mobility factor of the sensor nodes. The work of Zeng et al. [14] presented a hybrid scheme formulated based on Q-learning and fuzzy logic system to minimize and achieve an efficient balancing scheme in the MCA collision in the flying Ad-hoc network. Here, fuzzy logic is employed tochoose leader nodes considering the mobility pattern, and Q-learning is used to stimulate member node-rewarding to learn and evaluate multi-hop routes. The research work by Varshini et al. [15-16] presented a significant contribution in the networking, where the authors in [15] suggested a customized environment, namely NetAI-Gym, to evaluate RL agent for routing. In [16], the authors have presented a routing protocol based on Q-learning to select optimal routes. Also, the performance of presented routing schemes is evaluated with a rule-based agent algorithm. Hence, there are many RL-based approaches, but the existing studies lack modeling of a suitable environment to evaluate agent performance. However, there are few significant research works towards agent design and modeling. In the design-specific criteria, the researchers attempt to customized and enhance the design of agent mechanisms such as model-free approaches and model-based to achieve efficiency and accuracy in the model performance. The work carried out by Shen et al. [17] modeled a load balancing protocol based on the model-free approaches to minimize the congesting in peer-to-peer networking systems. The concept of the RL is mechanized to observe the environment state, such as processing capacity, queries, and resources associated with each peer. Further, the algorithm determines the suitable peers to relay queries based on the state observation. The study of Hendriks et al. [18], designed a Q-routing mechanism to perform optimal path selection and overhead reduction in the Ad-hoc wireless network. This study utilizes the AODV protocol for the route discovery process, and Q-learning is used to optimize the path discovery concerning QoS requirements. Johnston et al. [19] have introduced an intelligent routing scheme for battel networks to meet the real-time requirements. In this scheme, an approach of Q-learning is utilized to generalize and learn the next-hops to perform successful transmission of unicast- packets to the end nodes. The study considers duplication of the packets during unstable paths, and the packets are forwarded securely through multi-hop routes. The study uses cost-metric for the case of duplication, where if the cost factor is closer to zero, then more possibly that path is broken; if closer to 1, the path is stable. The researchers presented techniques emphasizing state overhead, action overhead, control packet overhead, and performance optimization in the performance-specific criteria. In the study of Wang et al. [20], the RL is utilized in the software-defined networking (SDN) enabled Internet of Things to improve routing performance. The SDN controller has a global view of the nodes and adapts routing based on mobility and traffic conditions. Further, an optimal route is determined based on the Q-learning approach. In Lin et al. [21], an adaptive routing scheme is suggested based on Q-learning concerning QoS optimization, including delay, loss, and bandwidth factor. The study of Tang et al. [22] adopted RL to develop opportunistic routing to support video streaming in the application of multi-hop wireless networks. The researchers also adopted the deep RL concept to achieve efficiency in the routing protocol [23]. The deep RL technique is used in Lan et al. [24] to perform efficient routing in the SDN. Table I summarizes the above-discussed literature to provide a quick insight for the readers.

TABLE I. SUMMARY OF ABOVE-DISCUSSED LITERATURE

Citation	Network Type	RoutingContext	Design	QoS metrics
[11]	CRN	Cluster-based	Model Based	PDR
[12]	WSN	Data-driven	Model Free	Delay Energy
[13]	MANET	Data-driven	Model Free	Delay, Overhead
[14]	FANET	Data-driven	Model Free	Delay, Throughput
[15]	MANET	-	Model Free	-
[16]	MANET	Data-driven	Model Free	PDR, Delay
[17]	P2P	Cluster-based	Model Free	Search time
[18]	WANET	Data-driven	Model Free	Delay, PDR
[19]	Battel networks	Data-driven	Model Free	Throughput
[20]	SDN	Standard Protocol driven	Model Free	PDR
[21]	SDN	Route Request driven	Model Free	Bandwidth, Delay and loss
[22]	Multi-hop network	Data-driven	Model Free	Delay, Throughput
[23]	Wireless network	Survey	Survey	-
[24]	SDN	Data-driven	Model Free	PDR, Delay and loss

Following are the significant open issues explored based on the above-discussed literature.

- It has been found that the majority of the study lacks modeling of a suitable environment that supports the function of Open AI Gym to assess RL agent algorithm.
- Open-AI Gym is a toolkit for benchmarking the agent algorithm. However, it is not considered in the existing approaches in the context of network problem.
- Due to the ever-increasing requirements for accuracy and efficiency in decision-making process for network routing, various approaches have been suggested over the years that can only approximate the complexity of the routing problem.
- The applicability of the existing methods is limited to the specific context and is not much effective in dealing with dynamic scenario, where the network topology and size changes dynamically.
- Very few research studies concerning Q-routing are found to emphasize the overhead memory problem.

The problem statement for the proposed study can be stated as "it is a very challenging task to integrate reinforcement learning function in the memory-efficient routing mechanism in an uncertain and dynamically-changing network environment."

III. Proposed System

The proposed study suggests a memory-efficient RL-driven routing mechanism. The proposed routing is based on the RL agent which is developed using function approximator that uses the Kanerva (K) coding scheme to store learnable parameters (weight and bias) to represent the learned policy for the action being performed by the agent towards exploration of better route establishment. The proposed algorithm searches for a near-optimal prototype set that provides a significant level of abstraction in memory consumption. The proposed algorithm is introduced in a dynamic network environment to perform path establishment for reliable data transmission.

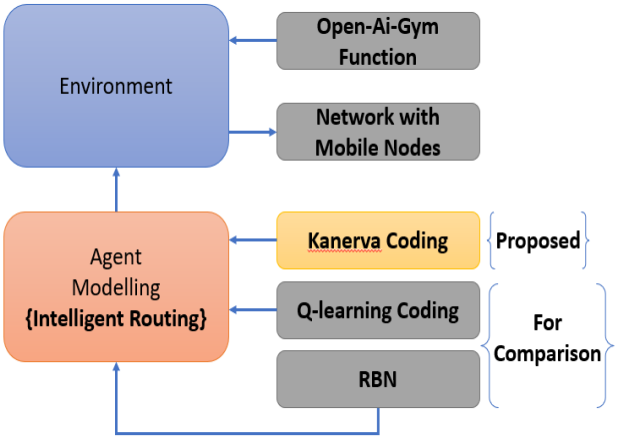

Fig. 2. Schematic Architecture of the Proposed System.

The schematic architecture of the proposed system is shown in Fig. 2, where environment modelling is carried out considering Ad-network with mobile nodes using Open-AI gym function. On the other hand, agent modelling is carried out to perform routing operation based on Kanerva coding technique and also the proposed study implements two other algorithm such as Q-learning and radial basis function (RBF) for the comparative analysis.

IV. Environment Modelling

The proposed study performs environment modeling that imitates the scenario of the mobile wireless networking system. The design and development of the environment are inspired by the work carried out by [15], in which a customized environment is developed, namely Net-AI-Gym. The network is composed of mobile sensor nodes with Ad-hoc features. The network as the environment is modeled as G(V,E), where V indicates vertices, i.e., sensor nodes, and E is the link for connecting the sensor nodes in the network. In this regard, the environment is represented as a collection of n vectors as in set: $\eta = \{\overrightarrow{N_1}, \overrightarrow{N_2}, \overrightarrow{N_3}, \cdots \overrightarrow{N_n}\}$, where, $\forall \overrightarrow{N_k} \in \eta$ denotes a sensor node with $\{X_k, R_k\} \in \overrightarrow{N_k}$, where X_k is the Node id and R_k denotes set of link and $k \in [1, n]$ and $n \in \mathbb{N}$, wheren ≥ 2. The study considers a mobile sensor node $\overrightarrow{N_k}$can be linked with many of the other sensor nodes within its proximity such that$\eta - \{\overrightarrow{N_k}\}$, therefore, $\forall n, R_k$is represented as follows: $R_l = \{\overrightarrow{L_1}, \overrightarrow{L_2}, \overrightarrow{L_3}, \cdots \overrightarrow{L_m}\}$ s.t $\forall \overrightarrow{L_k} \in R_l$ including $X_{k,}$, and W_k, where the W_k denotes the weight of the R_k. Fig. 3 shows a flow diagram of the environment with the Open-Ai-Gym function.

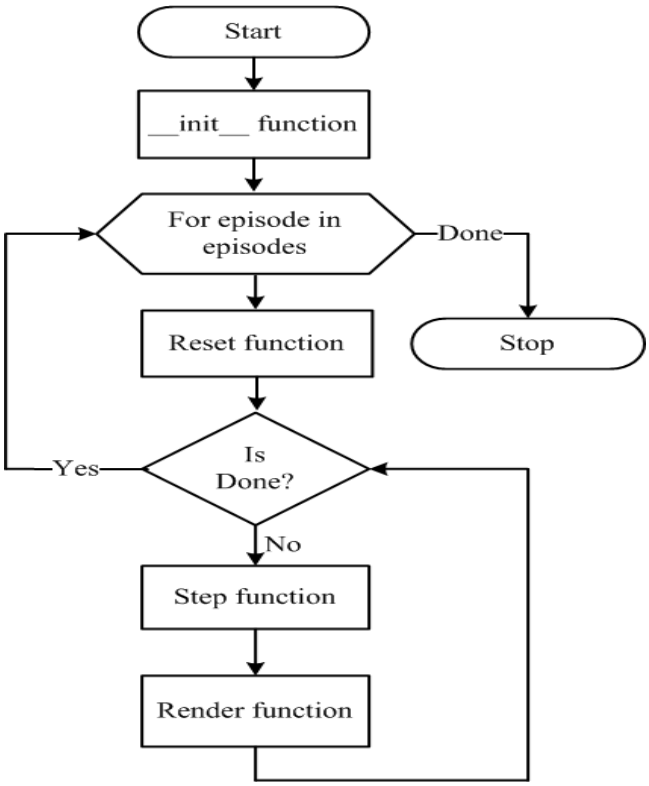

Fig. 3. Flow of Environment for Net-AI Gym [15].

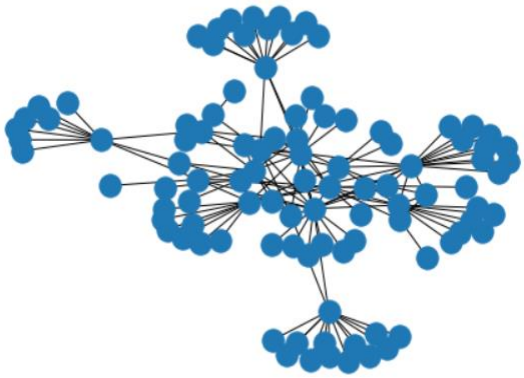

Fig. 4. An Environment with 100- Nodes Network.

Fig. 4 demonstrates the environment scenario with 100 mobile sensor nodes connected to each other. The design and development of the environment are carried out using the Open-AI gym function as mentioned in the flowchart depicted in Fig. 3 [15]. Open-AI gym enables the environment to satisfy the Markov process. The environment implemented in this study is scalable and flexible to N number of sensor nodes comprising Ad-hoc and mobility features. Thus, the proposed RL environment is the dynamic and uncertain imitating scenario of a mobile ad-hoc network. However, ensuring efficient routing is quite challenging due to the mobile and ad-hoc nature of the network. Therefore, the proposed study presents an efficient and sparse distributed memory-based agent model to perform routing operations in dynamic networks, discussed in the next section.

V. Agent Modelling

The prime objective of the study is to build an agent for solving the routing problem and optimizing the memory with the Kanerva coding. In the present study, the proposed agent mechanism uses function approximator as the Q function. However, the weights and biases of this function approximator are stored using the Kanerva coding. Due to which the storage space remains constant all the time. However, before discussing the proposed routing algorithm, it is better to understand the routing problem, its formulation, and the role of the function approximator in RL agent modeling.

A. Routing Problem

To determine the finest path from the source node to the destination node, context-adaptive and efficient routing mechanisms increase the probability of reaching the destination's data packets. Since the environment considered in the proposed study has a completely random and dynamic networking scenario, selecting the optimal number of intermediate sensor nodes for transmitting data packets is challenging. Thus, the routing process in a dynamic networking environment can be formulated as a Markov decision problem. Let us considered the sensor node n_i characterized by MDP tuple $\{S_i, A_i, C_i, T_i\}$.The S_i element of this tuple refers to set of states S in n_i. Let us considered N_i as a set of sensor nodes within the proximity or range (R_i) of n_i. In this regard, the state S in n_i comprises $\vec{d}$ and e, where the vector d is the distance value of all sensor nodes such that: $d_{i,j} \forall\, n_j \in N_i$ ande represents the energy value of all nodes in the range of n_i such that:$e_{i,j} \forall\, n_j \in N_i$. The $\vec{d}$ is obtained by computing the Euclidean distance between n_i and $n_j \in N_i$ as follows:

$$\vec{d} = \sqrt{\left(x_i^t - x_j^t\right)^2 + \left(y_i^t - y_j^t\right)^2} \tag{2}$$

Where, x and y is the positioning coordinates of the sensor nodes n_i and n_j. Moreover, the transmission or proximity range R_i of n_i can be determined into different intervals (I) of length ι expressed as follows:

$$I = \frac{R_i}{l} \tag{3}$$

According to the above numerical equation (3), the distance between n_i and n_j is a positive integer $\{1 \cdots I\}$ computed based on l and real distance value $d_{i,j}^t$ resides at time t. Here, the time 't' is considered because of the random nature of the network where the sensor nodes leave or join the network dynamically. Also, it is to be noted that $\forall\, n_i, l$ represents a unit of d and state interval I is subjected to theR_e^i. Furthermore, the remaining energy of $e_{i,j}, j \in N_i$ at $t+1$ can be computed as follows:

$$e_{i,j}^{t+1} = \left(e_{i,j}^t - \chi_j^t\right) \tag{4}$$

Where, χ_j^t indicates the amount of energy utilized by n_j at timet. The illustration of R_i is shown in Fig. 5. Considering all the above notions, the entire state S can be expressed as follows:

$$S = \left(d_{n_{i,j}}^t, e_{n_{i,j}}^t\right) \tag{5}$$

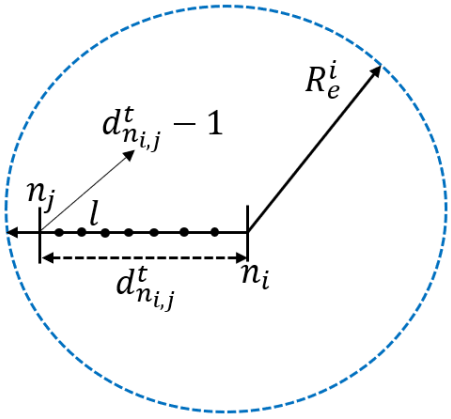

Fig. 5. Illustration of Transmission Range for n_i.

The second element A_i of the tuple represents a set of actions that n_i executes. The tuple element C_i refers to the return function of n_i such that: $C_i \leftarrow S_i \times A_i$ a Cartesian product of state-action space. The symbol T_i is the state transition probabilities for performing an action such that: $T_i \leftarrow S_i \times A_i \times S_i \in [0,1]$. This actually represents the transition from stateS (eq. 5) at time t such that:$S^t$to next state at time $t+1$ such that S^{t+1}. However, computing the precise value of T_i is usually impractical due to the absence of prior information about the network model, its random parameters, and its dynamic nature. In the proposed study, the development of agent is carried out based model-based

approach, and each mobile sensor node approximates its T_i using maximum likelihood approach and the possible $T_i \in \{transfer, drop, reset, and\ delivered\}$ and the path establishment process is completely in the control of the agent [15].

B. Agent Modeling based on Function Approximation

As discussed in previous section, the RL algorithm encounters communication and memory overhead problems when action space is very large in dynamic state spaces. In order to address this problem, the researchers have suggested the implementation of the function approximator, which is a basically an approach of neural network that the RL agents utilize to improvise its learning performance when introduced to dynamic and complex state spaces. Fig. 6 exhibits modeling of the agent using function approximator for dynamic network environment.

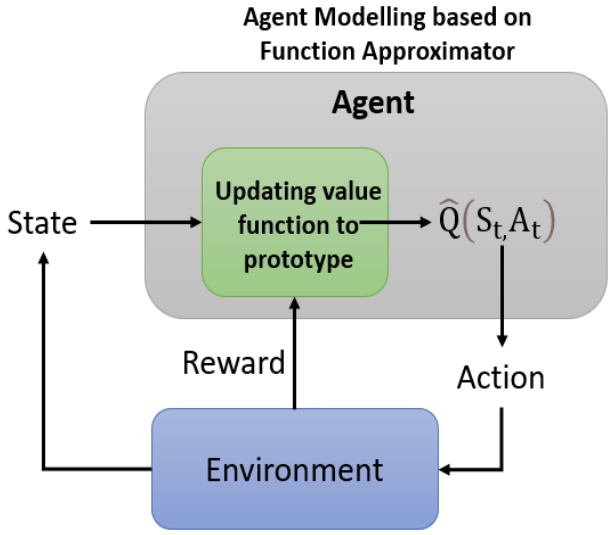

Fig. 6. Agent Modelling based Function Approximator.

In this work, the study utilizes simple artificial neural network (ANN) architecture that has parametrize function $'\theta'$ concerned with the learnable parameters, namely weight and biases, to represent approximated S_t and A_t value such that $\widehat{Q}(S_t, A_t)$. The values for all S_t and A_t pairs extracted from the large space are mapped into abstract components in $\vec{\theta}$. The approximated $\widehat{Q}(S_t, A_t)$ concerning $\vec{\theta}$ expressed as follows:

$$\widehat{Q}(S_t, A_t; \vec{\theta}) = \vec{\theta}^T \vec{\varphi}_{S_t, A_t} \quad (6)$$

$$\vec{\theta}\vec{\varphi}_{S_t, A_t} = \sum_{i=1}^{N} \theta_{A_t}(i) \varphi_{S_t, A_t}(i) \quad (7)$$

Where, $\vec{\varphi}$ denotes vector that consists of prototypes with N components that are constructed by state representation. In the proposed study, Kanerva coding is used as a state representation technique. The ideology behind using Kanerva coding [25] in the proposed agent modeling is that the Kanerva coding considers a small state as a prototype to store the value functions. Kanerva coding maintains a set of prototypes p as parameterized elements for the approximation and a value $\theta(p, A_t)$ is stored and updated for each prototype p concerning A_t. The approximation of state-action (S_t, A_t) is computed by a linear combination of $\vec{\theta}$ values of all adjacent prototypes of A_t, expressed as follows:

$$\widehat{Q}(S_t, A_t) = \sum_{p \in D} \theta(p, A_t) \mu(S_t, p)$$

where D denotes adjacent p with respect to S_t. The mechanisms of Kanerva coding in the proposed agent is designed based on the following algorithm.

C. Kanerva Coding

Kanerva coding (K-coding) deals with an architecture of sparse distributed memory [25] that utilizes prototype states to characterize the input sample states. The implementation of K-coding in the proposed agent for performing routing operation has multiple advantages viz. i) with the increase in network dimension (state space) due to increase in the number of sensor nodes in the network does not exponentially increase the prototypes required to learn the policy. Thus, facilitating efficient storage utilization and better scalability, i.e., constant memory, even the network size is increasing. The prime objective of implementing K-coding is to optimize the prototype set required to characterize a state space in an uncertain, dynamic, and large network system with minimal memory cost. The Block diagram of a proposed agent with K-coding is shown in Fig. 7.

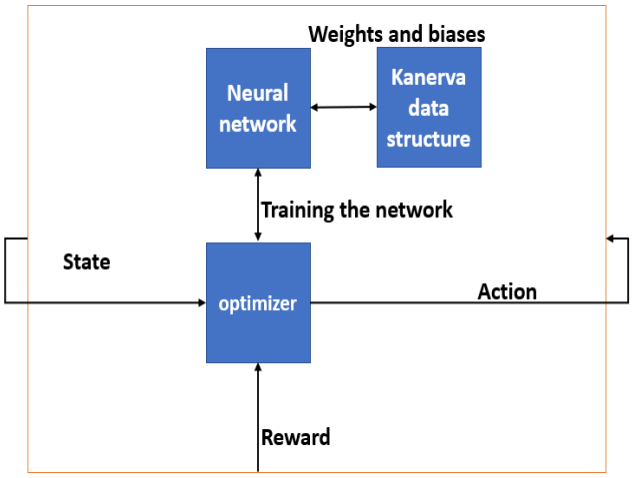

Fig. 7. Kanerva Coding in the Proposed Agent Function.

The proposed method uses K-coding as storage in order to store the values of weights and biases of the function approximator. The model works with the help of an ANN which employs regular weights and biases to find the best solution at the given state. In the case of networking, both state and action represent the current node in which the packet is present. The route is always decided with the help of ICMP packets. The agent will be present in all nodes, and the same will be updated everywhere as well. In the proposed method, Kanerva is used purely as a storage to the ANN, where it can store its weights and biases. Following is the complete algorithm for the efficient routing based on the K-coding function approximator agent mechanism.

Algorithm for routing with ANN and Kanerva Coding

Input: S, E, K, n, C_1, C_2, C_3, N
Output:
Start
1. **Init** $S_t \in$ S(state)
2. Build **ANN** Model → M
3. M ← f_1(N, 253, N)// f_1 model building function
   // where N is number of nodes
4. **Init** w(weights), b(biases)
   // where, w & b initialized as random values
5. **For** E in 4000 **do**// E is number of episodes
6. A_t = model. predict(S)// A_t is action
7. S = Env(A_t)// State changes when action is performed
8. M. train(R_t)// R_t is reward
9. **For** i:w **do**
10. **For** j: n **do** // $j = 1$ and n number of prototypes(P)
11. Compute D = $f_2(P, S)$// where D is the distance
    // f_2 distance function (Euclidian)
12. **End for**
13. **End for**
14. I = $f_3(D)$
15. **For** m = 1: 3
16. Ind_m = I(1 to c_m)
17. Store Ind_m
18. **End for**
19. **End for**

End

The algorithm takes parameters as input values K a set of S_t, n (ratio), closer prototype (C), where $C = \{c_1, c_2, c_3\}$ utilized to determine the C to the current S_t, based on the distance function. The proposed study considers the distance function as a Euclidean distance. In the first step of the algorithm randomly initializes p and takes input as a S_t. For each set of S_t i.e., K, the algorithm performs the computation of Euclidean distance between the prototype (p) and state S_t. The algorithm further stores the identity of computed distanced in vector format. In the next step of the algorithm, constructs a matrix Indm for the first 3 features and then performs the mechanism of offsetting by (m-1) x K. Basically, the K-coding mechanisms compute the length space between a state variable its actual distance. Further, the obtained data is then merged with a better-quality state similarity to achieve higher accuracy in its computations. K-coding diminishes the requirement for reallocation and resizing of prototypes, which significantly shortens the large dependencies of storing large action-space value in the Q-learning. Due to the strong learning ability and reduced computational complexity, the K-coding mechanism also improvises the entire learning experience.

VI. Experimental Evaluation

The proposed work's design and implementation are carried out using Python programming language in the Anaconda development environment. The experiment analysis is carried out considering comparative analysis, where the performance of the proposed agent mechanism is compared with other algorithms such as Q-learning and RBF. Both Q-learning and RBF are implemented in the study in RL agent design and evaluated on the same environment designed using the net-Ai gym environment proposed in a previous paper [15]. The following assumptions are considered in the simulation setting and the experimental analysis:

- The weights in the network represent the difficulty of packets being transferred.
- The weight is a composition of signal interruptions, battery, and distance.
- The weights keep varying to simulate dynamic or mobile networks.
- The number of nodes considered is 6 to 100 nodes.
- The various parameters shown here are recorded for networks with a various number of nodes.
- In this study, each network is trained for 4000 episodes.
- An episode is nothing but a simulation of a single packet from source to destination.
- The episode ends when the packet either drops or reaches the destination.

For the comparative study, the proposed study considered multiple performance metrics such as memory utilization, throughput analysis, average throughput, the processing time for route establishment, and analysis of the pathlength. Fig. 8 presents performance analysis regarding memory utilization.

The graph trend of Fig. 8 exhibits that the memory in Q learning increases exponentially, linearly in the case of RBN, and stays constant in the case of Kanerva coding.

The graph trend in Fig. 9 indicates that Q learning has low throughput, whereas RBN and K code has achieved higher throughput. Since K-code uses a function approximator in order to store the values, it underperforms a little bit compared to RBN; however, this difference is insignificant compared to Q learning. Even though K-coding underperforms slightly compared to RBN, it saves a lot of memory.

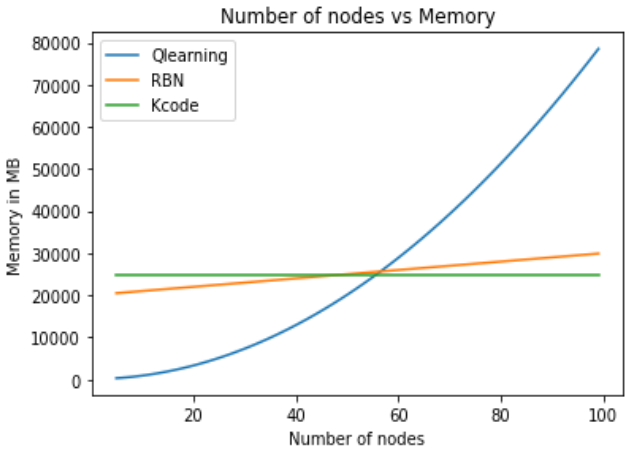

Fig. 8. Analysis of Memory.

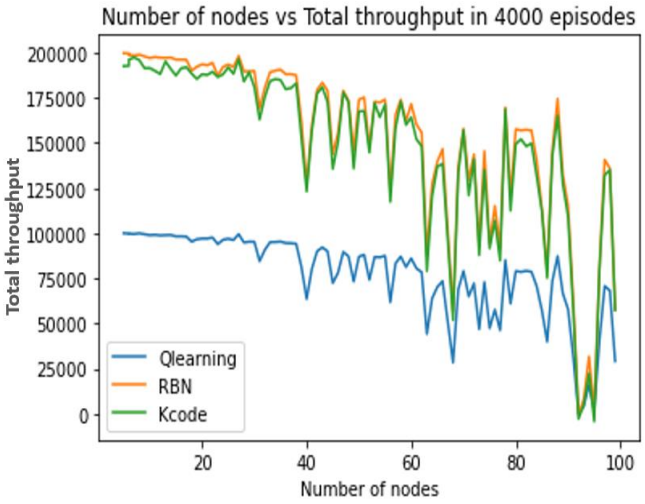

Fig. 9. Analysis of Total Throughput.

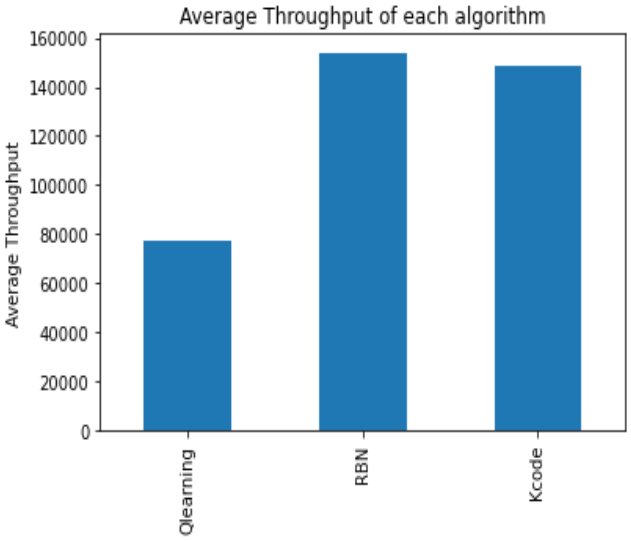

Fig. 10. Analysis of Average Throughput.

Fig. 10 shows the average throughput of each algorithm. The difference in the throughput of the RBN and K-coding is very insignificant.

Fig. 11 demonstrates the analysis of the average routing time of each algorithm. Routing time is defined as the time required by the algorithm to search for a suitable path in any network. Here Kanerva coding offers the least routing time, which is desirable. This is because the K-coding consumes lower memory and is faster to get trained.

Fig. 12 represents a change of path length along with nodes. It can be observed here that the K-coding always finds the shortest path. It is better than Q learning and RBN all the time.

A. Result Implication

- It is seen that the Q learning is not showing a good throughput. This is because the Q learning has less trainable parameters, and the table decides the reward. Even though the rewards are stored and stored aptly, the mechanism to calculate the future reward isn't as robust as the other two methods.
- The Q learning fails to perform in the case of memory management as well since the number of actions and states increases with an increase in nodes. To be specific, the memory consumption increases exponentially since the rewards are stored in the form of a table.
- The purpose of the Kanerva coding is to maintain a constant memory throughout.
- As observed from the above results, Kanerva coding underperforms in only one aspect: throughput. However, it does not pose a significant disadvantage as compared to RBN.

The proposed routing is designed based on the RL agent that utilizes K-code to achieve abstraction in the state space. Therefore, the proposed agent mechanism dynamically establishes the best node pathswith a low computational burden under uncertain and dynamic network traffic conditions.

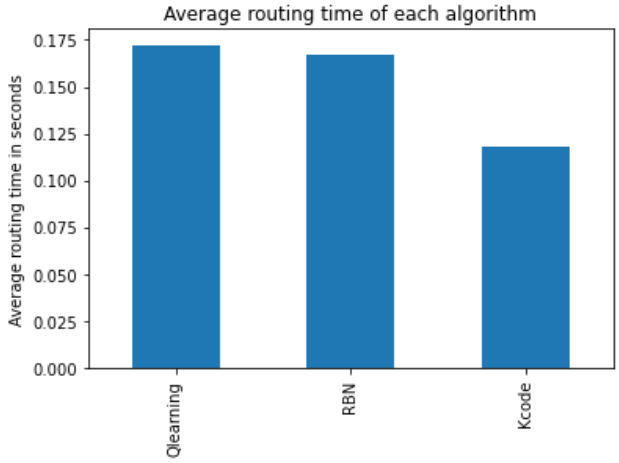

Fig. 11. Analysis of Average Routing Time.

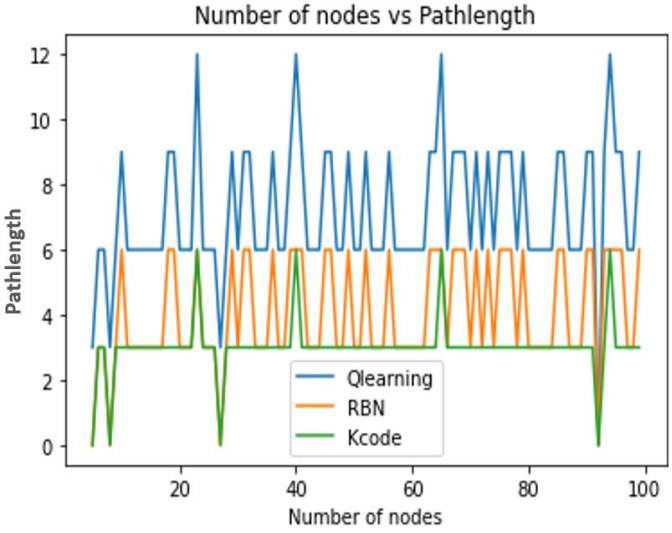

Fig. 12. Analysis of Pathlength.

VII. CONCLUSION

The proposed work is an extension of our previous research works, where in the first work, a suitable environment is designed to solve routing problems in network using the RL agent. On the other hand, the effectiveness of the proposed environment is evaluated in the second work by implementing a routing algorithm based on Q-learning and rule-based methods. In this paper, the proposed study improvises Q-routing performance to have better time and memory efficiency in the current work. The Q-learning consumes much memory and is not time efficient. RBN gives higher accuracy and time efficiency; however, the memory required for the algorithm will still increase with an increasing number of nodes. Hence, this work Kanerva coding is implemented to store the weights and biases of the function approximators used to build an agent for solving the routing problem and optimizing the memory. The benchmarking of the proposed system is carried out based on the comparative analysis concerning multiple network performance metrics. The study outcome proves the effectiveness of the proposed agent mechanism for routing operation under any given traffic condition in the network. In the future work, the proposed work can be extended in the context of multi-agent modeling of energy and security aware routing protocol in the dynamic networking environment.

REFERENCES

[1] Ramasamy, Dr. Velmani. (2017). Mobile Wireless Sensor Networks: An Overview. 10.5772/intechopen.70592.

[2] Lanzolla, A. and Spadavecchia, M., 2021. Wireless Sensor Networks for Environmental Monitoring.

[3] Khalaf OI, Abdulsahib GM. Energy efficient routing and reliable data transmission protocol in WSN. Int. J. Advance Soft Compu. Appl. 2020 Nov 1;12(3):45-53.

[4] Nakas C, Kandris D, Visvardis G. Energy efficient routing in wireless sensor networks: a comprehensive survey. Algorithms. 2020 Mar;13(3):72.

[5] Prabha K. Performance assessment and comparison of efficient ad hoc reactive and proactive network routing protocols. SN Computer Science. 2020 Jan;1(1):1-7.

[6] Thiagarajan R, Moorthi M. Efficient routing protocols for mobile ad hoc network. In2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB) 2017 Feb 27 (pp. 427-431). IEEE.

[7] Fan X, Cai W, Lin J. A survey of routing protocols for highly dynamic mobile ad hoc networks. In2017 IEEE 17th International Conference on Communication Technology (ICCT) 2017 Oct 27 (pp. 1412-1417). IEEE.

[8] Chandel A, Chouhan VS, Sharma S. A Survey on Routing Protocols for Wireless Sensor Networks. InAdvances in Information Communication Technology and Computing 2021 (pp. 143-164). Springer, Singapore.

[9] Boutaba R, Salahuddin MA, Limam N, Ayoubi S, Shahriar N, Estrada-Solano F, Caicedo OM. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications. 2018 Dec;9(1):1-99.

[10] Usama M, Qadir J, Raza A, Arif H, Yau KL, Elkhatib Y, Hussain A, Al-Fuqaha A. Unsupervised machine learning for networking: Techniques, applications and research challenges. IEEE access. 2019 May 14;7:65579-615.

[11] Y. Saleem, K. A. Yau, H. Mohamad, N. Ramli, M. H. Rehmani, "Joint channel selection and cluster-based routing scheme based on reinforcement learning for cognitive radio networks," in International Conference on Computer, Communications, and Control Technology, 2015.

[12] B.Debowski, P. Spachos, S. Areibi, "Q-learning enhanced gradient based routing for balancing energy consumption in WSNs," in 21st IEEE International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), 2016.

[13] W.-S. Jung, J. Yim, Y.-B. Ko, "QGeo: Q-Learning-based geographic ad hoc routing protocol for unmanned robotic networks," IEEE Communications Letters, vol. 21, no. 10, pp. 2258-2261, 2017.

[14] Z. Zheng, A. K. Sangaiah, T. Wang, "Adaptive communication protocols in flying ad hoc network," IEEE Communications Magazine, vol. 56, no. 1, pp. 136–142, 2018.

[15] VarshiniVidyadhar, Nagaraj R and D V Ashoka, "NetAI-Gym: Customized Environment for Network to Evaluate Agent Algorithm using Reinforcement Learning in Open-AI Gym Platform" International Journal of Advanced Computer Science and Applications(IJACSA), 12(4), 2021.

[16] VarshiniVidyadhar and R. Nagaraja, "Evaluation of Agent-Network Environment Mapping on Open-AI Gym for Q-Routing Algorithm" International Journal of Advanced Computer Science and Applications(IJACSA), 12(6), 2021.

[17] X.-J. Shen, Q. Chang, L. Liu, J. Panneerselvam, Z.-J. Zha, "CCLBR: Congestion control-based load balanced routing in unstructured P2P systems," IEEE Systems Journal, vol. 12, no. 1, pp. 802-813, 2018.

[18] T. Hendriks, M. Camelo, S. Latre, "Q2-routing: a Qos-aware Qrouting algorithm for wireless ad hoc networks," in 5th International Workshop on Cooperative Wireless Networks, Cartagena, Spain, 2018.

[19] M. Johnston, C. Danilov, K. Larson, "A reinforcement learning approach to adaptive redundancy for routing in tactical networks," in IEEE Military Communications Conference, Los Angeles, CA, 2018.

[20] C. Wang, L. Zhang, Z. Li, C. Jiang, "SDCoR: Software defined cognitive routing for Internet of vehicles," in IEEE Internet of Things Journal, 2018.

[21] S.-C. Lin, I. F. Akyildiz, P. Wang, M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in IEEE International Conference on Services Computing, 2016.

[22] K. Tang, C. Li, H. Xiong, J. Zou, P. Frossard, "Reinforcement learning-based opportunistic routing for live video streaming over multi-hop wireless networks," in IEEE 19th International Workshop on Multimedia Signal Processing, 2017.

[23] K. Arulkumaran, M. P. Deisenroth, M. Brundage, A. A. Bharath, "Deep reinforcement learning, a brief survey," IEEE Signal Processing Magazine, vol. Nov., pp. 26-38, 2017.

[24] C. Yu, J. Lan, Z. Guo, Y. Hu, "DROM: Optimizing the routing in SOftware-Defined Networks with deep reinforcement learning," IEEE Access, vol. 6, no. 18, pp. 64533-54539, 2018.

[25] R. Sutton and A. Barto. Reinforcement Learning: An Introduction. Bradford Books, 1998.