Computers and Electrical Engineering

Traffic Scheduling, Network Slicing and Virtualization Based On Deep Reinforcement Learning

--Manuscript Draft--

Manuscript Number:	COMPELECENG-D-21-02388R3
Article Type:	VSI:hci2(closed)
Corresponding Author:	Priyan Malarvizhi Kumar Kyung Hee University - Suwon Campus: Kyung Hee University - Global Campus KOREA, REPUBLIC OF
First Author:	Priyan Malarvizhi Kumar
Order of Authors:	Priyan Malarvizhi Kumar
	Shakila Basheer
	Bharat S. Rawal
	Fatemeh Afghah
	Gokulnath Chandra Babu
	Manimuthu Arunmozhi
Abstract:	The revolutionary paradigm of the 5G network slicing introduces promising market possibilities through multi-tenancy support. Customized slices might be provided to other tenants at a different price as an emerging company to operators. Network slicing is difficult to deliver higher performance and cost-effective facilities through render resources utilisation in alignment with customer activity. Therefore, this paper, Deep Reinforcement Learning-based Traffic Scheduling Model (DRLTSM), has been proposed to interact with the environment by searching for new alternative actions and reinforcement patterns believed to encourage outcomes. The DRL for network slicing situations addresses power control and core network slicing and priority-based sizing involves radio resource. This paper aims to develop three main network slicing blocks i) traffic analysis and network slice forecasting, (ii) network slice admission management decisions, and (iii) adaptive load prediction corrections based on calculated deviations; Our findings suggest very significant possible improvements show that DRLTSM is dramatically improving its efficiency rate to 97.32%, scalability and compatibility in comparison with its baseline.
Suggested Reviewers:	Rongbo Zhu rbzhu@mail.scuec.edu.cn
	Li Liu l.liu@derby.ac.uk
Response to Reviewers:	

Reviewer 1:

Q1: A novelty had proved and also the proposed concept of the architecture is excellent
Ans: Thanks for the review.

Reviewer 2:

Q2: The quality of the work is good and the concept is unique so accept as it is.
Ans: Thanks for the review.

Reviewer 3:

Q3: In their response to my previous comments, the authors have clarified my doubts and questions on their work so accept the paper for publication.

Ans: Thanks for the review.

Traffic Scheduling, Network Slicing and Virtualization Based On Deep Reinforcement Learning

[1]Priyan Malarvizhi Kumar*, [2]Shakila Basheer, [3]Bharat S. Rawal, [4]Fatemeh Afghah, [5]Gokulnath Chandra Babu, [6]Manimuthu Arunmozhi

[1]Department of Computer Science and Engineering, Kyung Hee University, Korea.

[2]Department of Information Systems, College of computer and Information Science, Princess Nourah bint Abdulrahman University, Saudi Arabia

[3]Department of Computer Science and Engineering, Capital Technology University, USA.

[4]Electrical and Computer Engineering Department, Clemson University, USA.

[5]Department of Computer Science & Engineering, School of Computing, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India.

[6]Nanyang Technological University, Singapore

*Mail: mkpriyan@khu.ac.kr

Abstract:

The revolutionary paradigm of the 5G network slicing introduces promising market possibilities through multi-tenancy support. Customized slices might be provided to other tenants at a different price as an emerging company to operators. Network slicing is difficult to deliver higher performance and cost-effective facilities through render resources utilisation in alignment with customer activity. Therefore, this paper, Deep Reinforcement Learning-based Traffic Scheduling Model (DRLTSM), has been proposed to interact with the environment by searching for new alternative actions and reinforcement patterns believed to encourage outcomes. The DRL for network slicing situations addresses power control and core network slicing and priority-based sizing involves radio resource. This paper aims to develop three main network slicing blocks i) traffic analysis and network slice forecasting, (ii) network slice admission management decisions, and (iii) adaptive load prediction corrections based on calculated deviations; Our findings suggest very significant possible improvements show that DRLTSM is dramatically improving its efficiency rate to 97.32%, scalability and compatibility in comparison with its baseline.

Keywords: Deep Reinforcement Learning, Network slicing, Traffic scheduling.

1. Overview of Network slicing and traffic scheduling

The 5G wireless services' architecture and study are motivated by evolving uses and diverse services [1]. Contrary to traditional networks, the latest network has varied for performance specifications, such as capacity, time and reliability [2]. The network provides 5G with usability, accessibility and cost-effectiveness to manage the traffic scheduling. The 5th revolution cellular network's fundamental improves the wireless system and is expected to be the leading infrastructure supplier for the next century [3,4]. Apart from the pure performance, cost, quality, and connectivity, 5G integrates the phone service environment's enhancement and offers a single network for heterogeneous networks [5,6].

5G can fully extract the recent developments in virtual servers and programming for the system and deliver a new network trimming technique to accomplish such an objective. Network slicing attempts to eliminate existing and increasingly centralised architectures and divide the entire network into many bits, each of which can be tailed to satisfy those service needs [7].

A network slice is created that enable providers to offer tailored network slices at varying times for multiple customers as an evolving sector [8]. Network Slicing is a successful strategy for tackling scheduling problems using technology communication and Device Components [9]. Network slices allow several functional channels to operate over a single functional system framework [10].

Network slices can be tailored to suit numerous system services' efficiency specifications and usage instances [11]. For instance, IoT resources involve large links, and low usage levels can sometimes be tailored in network slicing to avoid traffic congestion [12]. Several slices can be executed in the first period to assist delay-sensitive facilities, such as mobile enhanced perception and contact between devices. Therefore, network slicing introduces the new system and operating trends and increases system efficiency for both the content supplier and system suppliers regarding communication profit, quality of service, and delivery independence [13].

Service operators shall include network slices with efficiency and operational separation [14]. The separation of output guarantees that another network slice's activity would not impact a network slice's quality. Functional separation makes it possible to tailor the slice activities and the deployment of information in traffic scheduling [15]. The insulation between network slices decreases the capacity of multiplexing and diminishes device quality [16].

The separation is noted that when Network Services are exchanged in a short amount of time, multipathing performance increases [17]. Multipathing statement supports the adaptive network trend that can rapidly adjust the distribution of resources in network pits as per their real requirements in traffic scheduling [18]. Typically, a network slice needs resources in different technological areas, such as communication systems, transmission and edge/cloud infrastructure [19].

The use of rendering services to match it with consumer operation is hard to produce better quality and cost-effective services. The main contribution of DRLTSM is described below

- DRLTSM aims to engage with the world in search of new potential actions and improve trends that are considered encouraging responses, which has more beneficial implications.
- The goal of DRLTSM is to establish three key sliding blocks: (i) traffic analysis and network function predicting; ii) admittance control judgments for the system and iii) dynamic load prediction adjustment focused on measured variations;
- The results indicate that the DRLTSM increases its performance, scalability, and usability compared with the other methods.

The remaining article is organized as follows: Section 2 comprises various background studies concerning Network slicing and traffic scheduling. Section 3 Elaborates the proposed DRLTSM to engage with the world in search of new potential actions and improve trends that are considered encouraging responses. Section 4 constitutes the DRLTSM increases its performance, scalability, and usability. Finally, the conclusion with future perspectives is discussed in section 5.

2. Literature survey on Network slicing and traffic scheduling

This section discusses several works that various researchers have carried out; Haozhe Wang et al. [20] developed Data-driven dynamic resource scheduling (DDD-RS). DDD-RS created a new deep learning framework for the complex resource planning for network slice trimming to ensure automated and effective utilisation of services and end-to-end service quality. Thus, DRL can be utilised to collect data from expertise by communicating with the system and allowing vibrant resource alteration to various parts to enhance resource use and ensure service quality (QoS). The experiment shows that the suggested resource plan can adaptively allocate multi-slice assets and meet the appropriate quotas.

Mu Yan et al. [21] discussed intelligent resource scheduling strategy (iRSS). An iRSS is a key concept to leverage and enhance learning by a shared education system consisting of deep learning (DL). Primarily, DL is used to allocate large time scales and is used to plan online resources to address limited network complexities on time, namely unreliable estimation and unpredictable network status. The quantitative results indicate that iRSS consolidation meets the online planning criteria and can substantially increase resource usage, thus ensuring separation among slices related to other standard algorithms.

Fengsheng Wei et al. [22] introduced Exploiting Deep Reinforcement Learning (EDRL). EDRL uses the readjustment of a core network slice to minimise resources' long-term utilisation using deep reinforcement learning. The use of the traditional Deep Q Network (DQN) is an unsolvable issue as it has a multi-layered, distinct area that is difficult to explore. The inclusion of the action branch structure in DQN in a distinct BDQ network significantly decreases estimated activities. In an attempt to assess EDRL output, comprehensive simulation tests are undertaken, and quantitative findings indicate that EDRL can reduce its long-term asset use and obtain high efficiency in resources compared to several benchmarks.

Yu Abiko et al. [23] proposed Flexible Resource Block Allocation (FRBA). FRBA suggested a system for slicing radio access networks (RAN) that efficiently allot RAN assets using deep learning. The amount of slices operated by a basic station in RANs varies from exposure to the changing operation in the corresponding client set to the ground station. In the assessment, the test different scenarios and display that the average slice performance is around 97 %. FRBA architecture allows for the optimum allocation of resources regardless of the number of slices by adjusting the number of representatives.

Dario Bega et al. [24] developed An Artificial Intelligence -Based Framework (AI-BF). AI-BF managed the network slicing roles and assets is a demanding job that calls for effectiveness in some instances, even now in real-time judgments. Implementing AI in various stages of the slice life cycle, from entry control to dynamical resource distribution in the networking centre and radio access, is a general structure for AI-based slice administration. A responsive use of AI for the slicing of the network gives the user significant value, with projected outcomes in typical case studies up to 25 to 80 points.

Dario Bega et al. [25] introduced DeepCog. It is launched as a modern cognitive strategic planning data analytic platform for 5G networks. DeepCog estimated the capacity to satisfy potential traffic growth within the network slices and considered the operator's preferred equilibrium between over-supply and infringement of service requests. Furthermore, in the dynamic, sliced network, congested traffic DeepCog is used for a detailed first study of the compromise among the power over dimensions.

Based on the survey, to avoid traffic congestion, DRL is implemented for network slicing situations, addresses power control and core network slicing, and priority-based sizing involves radio resource.

3. Deep Reinforcement Learning-based Traffic Scheduling Model (DRLTSM)

The innovative 5G network paradigm offers many opportunities with multi-task resources in a different field. DRLTSM model has been proposed for network slicing, virtualization, and traffic management in a 5G environment. Engaging with the community to search for new alternative actions and reinforcement patterns have been discussed. Reinforcing trends that are perceived as promising alternatives provides more favourable outcomes. Radio tools for network shortening, power controls, central network shortening and priority are included in the DRL. DRLTSM aims to create three main blocks I) prediction and traffic analysis of slice network, (ii) evaluate the entry of the network slice and (iii) correction adaptive load projections based on measured differences;

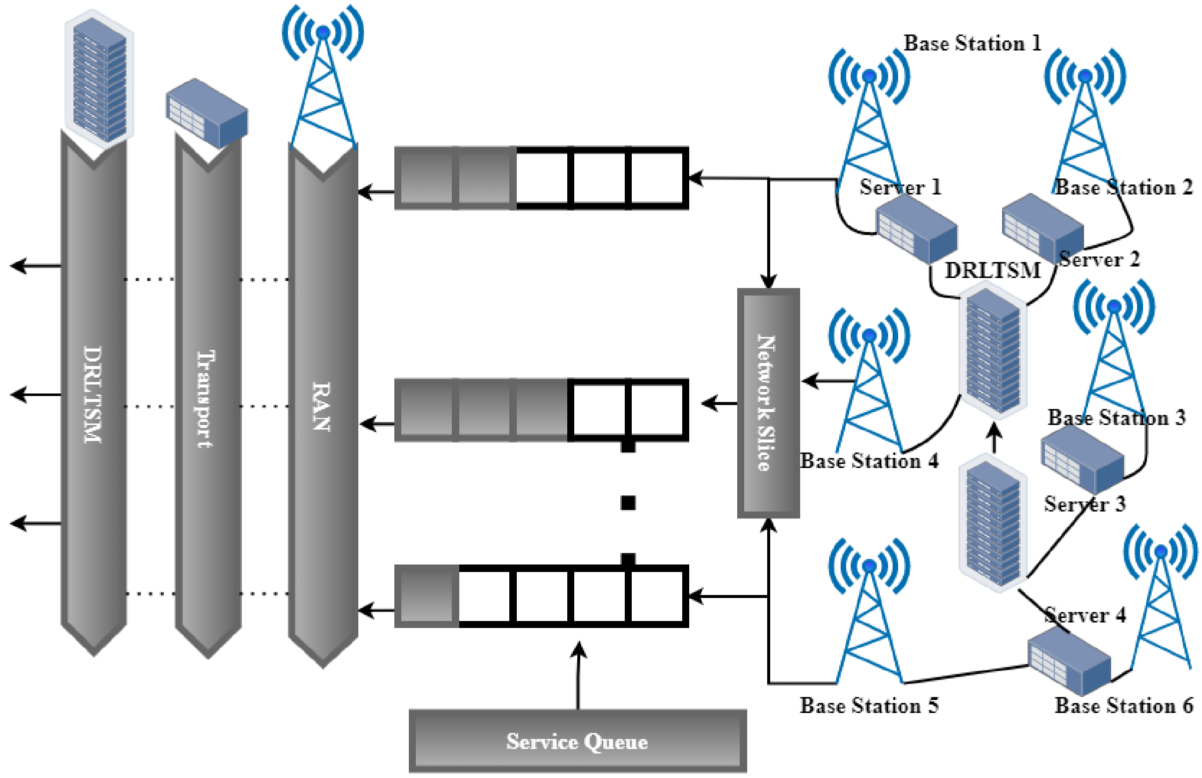

Figure 1: Proposed DRLTSM

Figure 1 described the Proposed DRLTSM. Network slices should be uniquely personalized to suit different network services' capacity criteria and IoT applications. A segment can be optimised for IoT networks that need massive connections with low data rates. A further slice may be installed in addition to supporting the late-sensitive installations. Network slicing introduces new management and operational patterns for network operators, service providers' reliability, and improved network efficiency in network sales and service consistency. The precise association between the capital and network slices' output is almost difficult to achieve. A network slice typically needs services from several technological fields, for example radio connectivity, transportation network and edge/cloud. Between these instruments, there is rather a complex negotiating and slice performance. Accelerated computing in the edge/cloud servers would compensate for a brief latency in the radio access network. Therefore, a closed mathematical expression is missing the association between resource and network slice results. The new multi-resource assignment work typically assumes that different resources are distributed according to a certain ratio unit range.

The second problem is that the range of space in mobile transport requires network divide services to be allocated correctly in various geographic areas between the base stations and the edge/cloud servers. In this paper, the proposed method utilizes the EdgeSlice, a decentralised method for the orchestration of resources, automatically slices the complex end of wireless networks. A new approach of EdgeSlice is an effective arrangement of end-to-end networking and computation outlets based on the deep reinforcement learning traffic scheduling model (DRLTSM). In the proposed DRLTSM approaches, a central performance coordination officer and several dispersed orchestration agents conduct resource orchestration.

End-to-end wireless edge calculation network consists of a multi-station (BS) and Radio Access Network (RAN), edge/cloud computer servers, and a transportation network for computer server connection. Several network sections in each RAN, as shown in Figure 1, require end-to-end resources that their user mobility is enabled and secured. Each RAN has network slices that buffer its customers' arrival traffic using the service queues' length. DRLTSM considers that the network is time-spending, and network operators can track the performance of network slices with a minimum period and dynamically adjust their resource orchestration. These managers allow end-to-end resource dynamic configuration during runtime in EdgeSlice. Experimental EdgeSlice framework prototype and implementation. Test EdgeSlice's output using both prototype device tests and network simulations powered by traces.

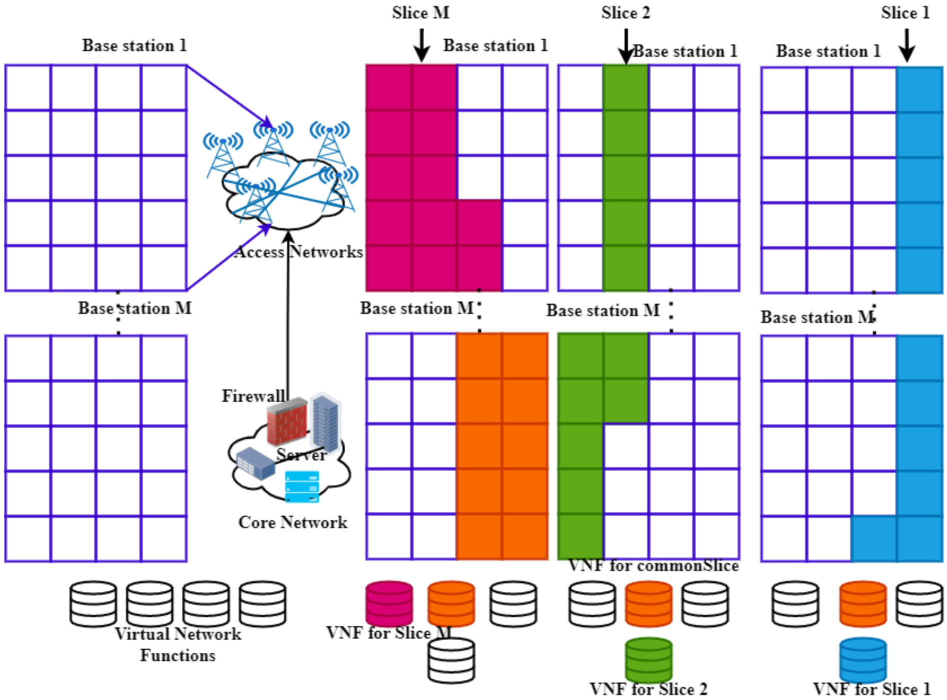

Figure 2: Network Slicing with virtual network function

Figure 2 signifies the Network Slicing with virtual network function. Management of capital is a permanent trend during wireless networking capacity evolution. Resource management should be viewed from multiple viewpoints for network slicing. Resource control for the network slice includes both the radio access portion and the main network part, which have slightly different optimization objectives, as in figure 2. Due to the extreme minimal spectrum space available, radio access management makes significant attempts to allocate one slice of Resource Blocks (RBs) to ensure appropriate when seeking a reminder and a slight delay. The commonly implemented optical transmission in central networks has led to core networking optimisation by designing a standard virtualized network functions (VNF) method with a minimum scheduling time to transmit packets correctly from a specific slice. The resource management dilemma can be formulated $O = \xi . SE + \gamma . QoE$ All tools and by satisfying Quality of experience (QoE), where γ and ξall resources are called system engineering (SE) and QoE.

The summation of these variables may be viewed as a help to the learning agent. The objective of resource management for network slicing could take account of many variables.

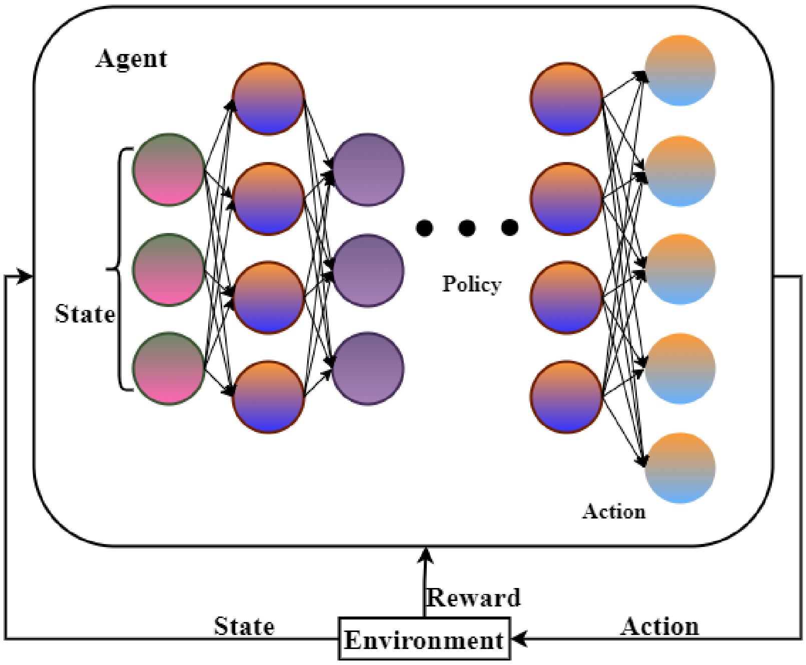

Figure 3: Deep Reinforcement learning

Figure 3 explores the Deep reinforcement learning for decision-making in resource scheduling using a combination of DNN and RL, known collectively as deep reinforcement learning. The DRL syncretization of the absolute IoT-resource scheduling method integrates the traditional policy reinforcement learning to model the system context and deep neural networks to model the policy network. Reinforcement learning (RL) is a machine-learning field where intelligent agents can take steps to optimise the concept of incremental rewards. Reinforcement learning, along with supervised and unattended learning, is one of three basic machine-learning paradigms. This is known as an incentive mechanism that allows AI platforms to conclude rather than a forecast. For improving learning structures, incentive functions are used. Engineering the incentive function defines the benefits of actions. The State value marks the cumulative incentive to be earned when this state is selected to be a point of departure. The principles are important for future decisions to be taken. A protocol determines the actions of the learner at a specific time. In general, a strategy is a mapping of perceived environmental states to steps that are to be taken in these states.

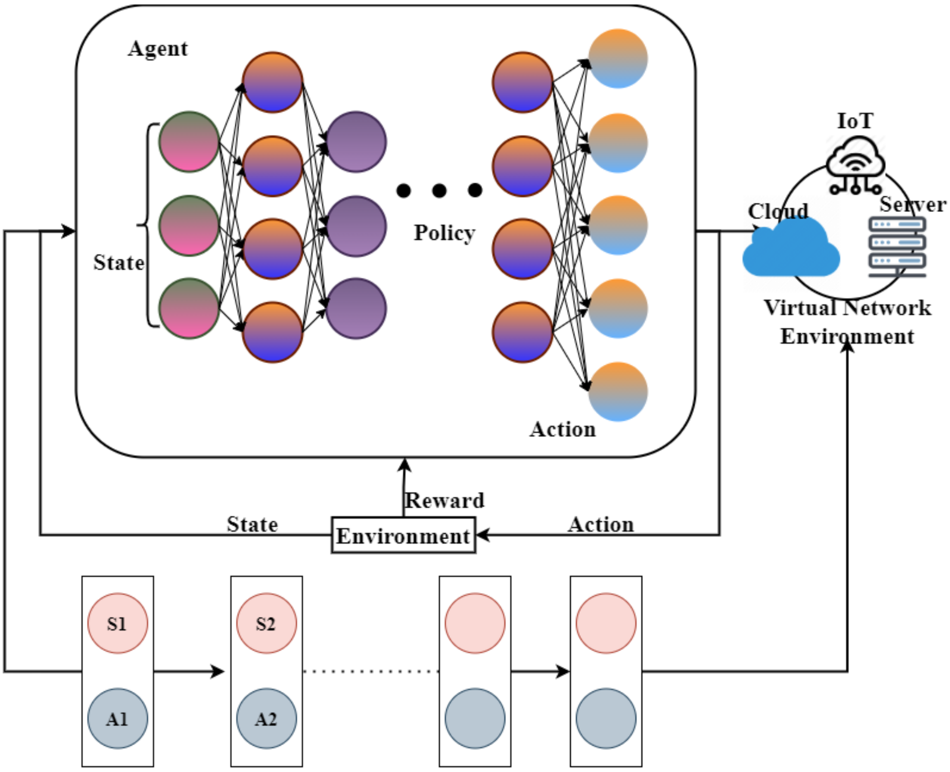

Figure 4: DRLTSM based policy generation

Figure 4 shows the DRLTSM based policy generation. A state $T_s$is information completely encapsulated about an IoT ecosystem that demonstrates the reasons to decide. Model the state as a device instance that comprises the existing allocation of resources and pending service demands. The method would result in a series in which the services are allocated to the service request. Ideally require a proper state representation for the neural network. Although several service requests can be sent, address first N requests as a mini-batch for planning to ensure a coherent state representation. In addition, there are many potential variations of allocation sequences that lead to a different number of states and to a TSM of multiple states (which depends on the current state, not on the history). In this way, reinforcement learning can be achieved by taking a sequenceT_s into account at times. Action is to devote resources to incoming demands for IoT services. After each operation, the transition from the current status T_s to new state T_{s+1} $s+1$ is performed. An operation contributes to assigning the services needed for each service request from the N request mini-batch. This vast number of actions complicates the whole resource planning task.

It has been used to minimise the size of the intervention area using several allocations at a time S. The DRL scheduler, during preparation, attempts to learn how the acts are distributed along with the called rewards. Reward $O_s = O(T_s, b_s, T_{s+1})$is the scalar system input after the system shifts. Due to network efficiency impacts, an agent takes action any time a favourable or negative award is earned. It then shows how good an agent in state S. Our goal is to reduce the mean reaction time and energy usage overall to a minimum. The immediate reward received by the agent with the equation reward function. The overall objective of the strategy is to optimise potential accrued rewards. The behavioural role of an agent is a strategy. It is a map that describes a distribution of probability overactions. Each supervision offers abstract knowledge on such indicators within an IoT ecosystem ability. As the structure is passed to the next state, these parameters' values will be modified until intervention is carried out. The effects of measurement can be interpreted as an input from the actual state of the operation. Scalar feedback can be registered for any resource allocation action and is the ultimate incentive for this unique action.

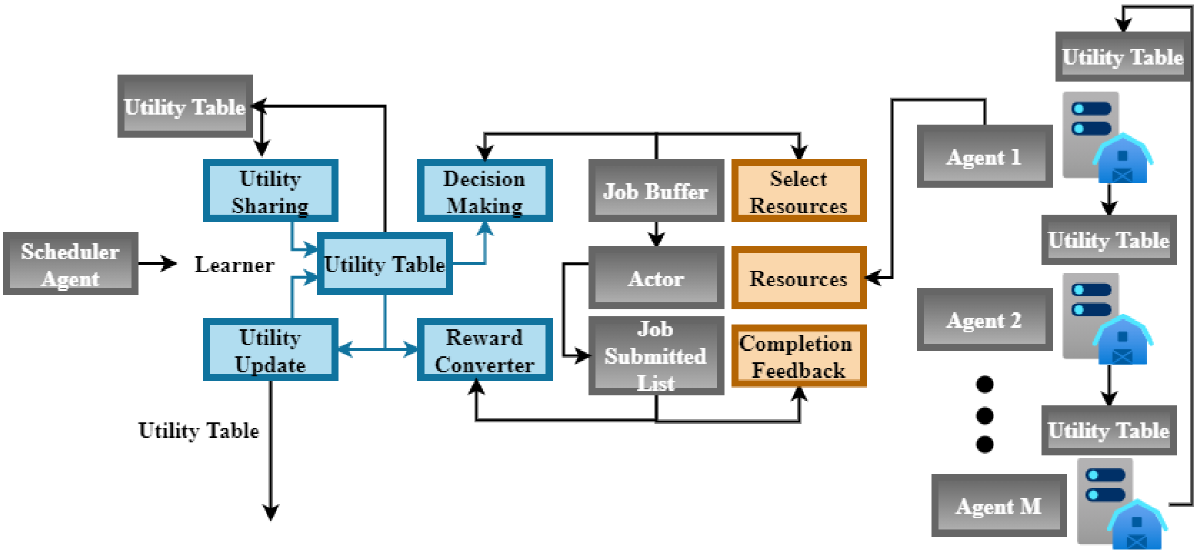

Figure 5: Traffic Scheduling

Figure 5 explored Traffic Scheduling. The new approach involves simplified decision-making in the schedule of work using a utility-based learning technique. The new approach will prevent problems of scalability and alignment with task planning. Using an ordinary distributed learning strategy eliminates the scalability issue and ensures ongoing multi-agent collaboration focused on knowledge exchange by restricted agent communication. As in figure 5, the design of the use of the OSL network schedule system is demonstrated. The top loop reflects the schedulers or the different agents sharing the utility tables. One scheduler is drawn in-depth within the lower loop. Each planning agent is comprised of two components, the Learner and the Actor. The learner generally collects the utility table from retired agents and selects resources to file subsequent queuing jobs in the work buffer. The Reward Converter analyses and converts the completion signal into reward signals to upgrade the utility table. The actor collects the new tasks and arranges them to put them into the work buffer, send them by the learner's choices, and log the entry on the job list requested. To finish, the actor modifies the submitted job list by task fulfilment, and once the task is finished, it is removed from the submitted worklist. The OSL solution can achieve efficient load balance according to the simulation data, and the results have shown that its effectiveness often resembles some centralised programming algorithms. The evaluation results indicate the convergence property and stability of OSL in various gird environments. Future research work could involve the creation and deployment to actual grid environments of the proposed system.

The TSM is the simplified paradigm for modelling decision-making issues in situations where the decision's outcome is partly random and is influenced. Mathematical, RL fits the common idea of a Markov decision-making (MDP). An MDP may be defined with five-fold $N = \langle T, B, Q(T'|T, B), S, \beta \rangle$Where T and B are indications of finite state and action. Ultimately, MDP's objective is to find the policy $b \in B$that defines the chosen behaviour under state Tin order to optimise the value function usually described by Bellman's equation as the discounted cumulative reward in equation (1):

$$U^{\pi}(\hat{T}) = F_{\pi}\left[\sum_{L=0}^{\infty} \beta^{l} O\left(\left(T^{(l)}, \pi(T^{(l)})\right) \middle| T^{(0)} = \hat{T}\right)\right]$$

$$= F_{\pi}\left[O\left(\hat{T}, \pi(T^{(l)})\right) + \beta \sum_{T' \in T} Q\left(\left(T' \middle| \hat{T}, \pi(\hat{T}) U^{\pi}(T')\right)\right)\right] \tag{1}$$

As described in equation (1), bellman’s cumulative reward has been calculated. The Bellman equation could benefit from dynamic programming if a priori is understood without any random variables for the $Q(T'|T, b)$ State transformation probability. However, influenced by both control theory and behavioural psychology, RL needs to achieve optimized policyπ^{*} For unfamiliar, partly arbitrarily random dynamics under some conditions. Since RL does not know specifically whether it close to its target, it needs a compromise between testing new possible behaviour and taking advantage of the previously learned knowledge. The most upfront method for feature estimate is a linear approach. Enchanting the sample of Q-learning feature could be estimated by a linear arrangement of m orthogonal bases $\varphi(T,b) = \{\varphi_1(T,b),\cdots \varphi_2(T,b)\}$ that is $P(T,b) = \theta_0.1 + \theta_1.\varphi_1(T,b) + \cdots + \theta_0.\varphi_m(T,b) = \theta^S\varphi(T,b)$, where θ_0 is a biased duration through one absorption addicted to the φ for the straightforwardness of demonstration, and θ is a vector using the dimension of m. The feature estimation in the Q-learning denotes that $Q(s,a) = \theta^S\varphi(T,b)$ should be as nearby as the learn S target quantity $P^+(T,b) = \sum_T Q(T'|T,B)\,[O(T,b) + \delta max_{b'}P^+(T',b')]$Overall the state-action pairs. Since all the states can not be transverted, the "target" value may be calculated by the samples from the minibatch and $P^+(T,b) >\approx O(T,b) + \delta max_{b'}P^+(T',b')$. To make $P(T,b) - \theta^S\varphi(T,b)$ the method the target charge$P^+(T,b)$, the objective function could be represented as

$$K(\theta) = \frac{1}{2}\left(P^+(T,b) - P(T,b)\right)^2$$

$$= \frac{1}{2}\left(P^+(T,b) - \theta^S\varphi(T,b)\right)^2 \tag{2}$$

As calculated in equation (2), the objective function has been expressed. The parameter θ minimizing $K(\theta)$ could be achieved by a gradient-based approach in equation (3):

$$\theta^{(j+1)\leftarrow}\theta^{(j)} - \sigma\nabla K\left(\theta^{(j)}\right) = \theta^{(j)} - \sigma\left(P^+(T,b) - \theta^S\varphi(T,b)\right)\varphi(T,b) \tag{3}$$

As expressed in equation (3) gradient-based approach has been explored. The approximation of the function reduces the unknown parameter to a dimensional m vector for a broad state-action space. The associated form of the gradient thus provides a computationally efficient solution to the parameter approximation. The calculated value function could not be correctly modelled using the linear approximation function. Researchers then suggested that certain non-linear means substitute the approximation $P(T,b{:}\,\theta)$.

This segment deals with how DRL is used for the slicing of radio property. Mathematically, given a list of current slices, $1,\cdots,M$ sharing the aggregated bandwidth Z and with fluctuating requirements $C = (C_1,\ldots,C_M)$, DQL tries to provide sharing solution for bandwidth $Z = (Z_1, \dots, Z_M)$ the bandwidth-sharing solution, to optimise the medium to long term $F\{O(Z, C)\}$ where $F(\cdot)$notes that the claim is predicted in equation (4):

$$arg_Z maxF\{O(Z, C)\} = arg_Z maxF\{\tau. TF(Z, C) + \gamma. QoE(Z, C)\}$$

$$Z = (Z_1, \dots, Z_M)$$

$$Z_1 + \cdots + Z_M = Z$$

$$C = (C_1, \dots, C_M)$$

$$C_j \sim Certain\, Traffic\, Model, \forall j \in [1, \dots, M] \tag{4}$$

As deliberated in equation (4), long term maximize reward expectation has been performed. The main problem (4) is to overcome fluctuations in demand without first knowing about traffic models. Therefore, DQL is the perfect solution for problem resolution.

Traffic projections shall be carried out for each occupant on an aggregate basis. Each locator j might ask for another custom-designed request for the network slice. The forecasting method can conveniently categorise traffic demands based on the corresponding service demand, such that a prediction can be performed separately. First, presume that traffic requests are consistently spread across the whole network. To allow for tenant j traffic volumes for class l (i.e. meeting basic service requirements) to be a point operation $\xi_j^{(l)} = \sum_{s=0}^{S} \lambda_s\, O_j^{(l)}(s)$where λ_sIndicates the Dirac calculation for sample s is not a single point. Appropriate services express traffic requests from $O_j^{(l)}(s)$. Based on the traffic queries' regular structure, the traffic estimates are based on the S_{OBS} time scale observed. This is given by the $O_j^{(l)} = (O_j^{(l)}(s)\ (s - S_{OBS}), O_j^{(l)}(s)\ (S - (S_{OBS} + 1))\ \cdots, O_j^{(l)}(s)$ variable. The mechanism has a regular behaviour, which turns into WS replicated seasons over time in fixed traffic patterns. Assume the state is stationary and ergodic within a single season. Therefore it can simulate and model possible traffic demands for a given network slice using the Holt-Winter (HW) predictive method. HW indicate a particular traffic query $O_j^{(l)}(s) by\ O_{j,s}^{(l)}(s)$. DRL focus on the HW adjustment variant. The seasonal effect does not depend on the observed time window's average traffic level; instead is taken into account by level and pattern effects expected values. Forecast these demands based on the K_S, pattern A_s seasonal and T_S variables as follows, normal practice in equation (5):

$$K_S = \sigma\left(O_{j,s}^{(l)} - T_s - Z\right) + (1-\sigma)(K_{s-1} + A_{s-1})$$

$$A_s = \gamma(K_s - K_{s-1}) + (1-\gamma)A_{s-1}$$

$$T_S = \delta\left(O_{j,s}^{(l)} - K_{s-1} - A_{s-1}\right) + (1-\delta)T_{s-Z}$$

$$\hat{O}_{j,s}^{(l)} + S_{Window} = K_s + A_s g + T_s + S_{Window} - Z \quad (5)$$

As found in equation (5), HW normalized practice has been explored. While the optimum HW parameters σ, γ and δ can be obtained during training to concentrate on existing strategies on predictive errors and how the provision's inaccuracy will impact our network trimming approach. The above equation (5) to determine the forecast HW standard practice with pattern and seasonal factors. Define $F_{j,s}^{(l)}$ the one-phase prediction error in equation (6)

$$F_{j,s}^{(l)} = O_{j,s}^{(l)} - \hat{O}_{j,s}^{(l)} = O_{j,s}^{(l)} - (K_{s-1} + A_{s-1} + T_{s-1}) \quad (6)$$

As determined in equation (6), one phase prediction error has been formulated, calculated during our forecast algorithm preparation. The above equation (6) to predict one state error are compared with the observed ones.

Provided that our method $\lambda_j^{(l)}$ is ergodic and assumes optimum HW, at any expected value at time W can extract a forecast interval $\left[\widehat{KK}_{j,W}^{(L,S)}, \widehat{gg}_{j,W}^{(L,S)}\right]$ where potential traffic requests lie with a certain $Y_j^{(l)}$. Then this applies in equation (7):

$$\widehat{gg}_{j,W}^{(L,S)} or \widehat{KK}_{j,W}^{(L,S)} = O_{j,s}^{(l)} + (-)\vartheta_y \sqrt{var\left(F_{j,W}^{(l)}\right)}$$

$$Var\left(F_{j,s}^{(l)}\right) \approx \left(\left(1 + (W-1)\sigma^2 \left[1 + W\gamma + \frac{W(2W-1)}{6}\gamma^2\right] \alpha_F^2\right)\right)$$

$$QO\left\{\widehat{KK}_{j,W}^{(L,S)} \leq O_{j,W}^{(l)} \leq \widehat{gg}_{j,W}^{(L,S)}\right\} = O_j^{(l)} \forall_W \in [S+1, S+S_{Window}] \quad (7)$$

The forecast interval obtained in equation (7).In the above equation, ϑ_y is the value of the one-step normal distribution is $Y_j^{(l)}$*the* likelihood and the uncertainty of one-stage estimation forecasting error $\alpha_F^2$is $\alpha_F^2 = var\left(F_{j,s}^{(l)}\right)$ in the observed time window. Because of traffic SLAs actions concentrate on the higher limit of the interval of estimation, as it accounts for 'the worst-case' of a projected volume of traffic equation (7) contributes to lower precision and is similar to actual network slice demand a bigger time forecast window of S_{Window}A greater number of expected W values. Instead, an exact prediction with a lower likelihood of error could lead to extreme fines if the expected SLA is not guaranteed. The variance likelihood of Forecast error ($Y_j^{(l)}$) to satisfy the service specifications and several predictions point the prediction process would carry out. For example, requests for best-effort traffic without strict criteria will tolerate a long forecast with an imprecise value. This means that, irrespective of the probability $\widehat{gg}(_{j,W}^{(L,y)})$ is similar to the actual (future) values $O_{j,s}^{(l,y)}$as the number of W values to be predicted is restricted. Therefore capture the $Y_j^{(l)}$Low likelihood of forecast error for this form of operation. On the other hand, SLA related has done in a shorter time frame when assured bit rate traffic is considered, making our predictive model considerably more complicated and includes substantially more expected values W. Our device models traffic of this kind with a higher predictive error $Y_j^{(l)}$. Traffic class $L = 0$has a forecast horizon less than certain traffic classes must then be predicted according to the traffic classes and a higher number of W values. The maximum benefit between the tranche request and the need for traffic is calculated expected as $\hat{C}_j^{(L=0)} = \max_{W \in S_{Window}} \left(O_j^{(l)} - \hat{O}_{j,s}^{(l)}\right)$Then DRL calculates the likelihood of a predicted error accordingly in equation (8):

$$Y_j^{(L=0)} : \vartheta_y \sqrt{var\left(F_{j,Z}^{(L=0)}\right)} = \hat{C}_j^{(L=0)} \quad (8)$$

As initialized equation (8) likelihood of a predicted error has been computed. Our complex scenario predictive model considers consumer mobility and no longer retains the expectation of traffic frequency. The proposed DRL consider that the whole region is a multi-cell system. Traffic scheduling depends on human mobility patterns to develop predictive algorithms which are reliable in practical conditions. According to the least action trip plan (LATP), with σ_{SLAW} , users start to move between the subset of paths within the chosen clusters. On the user drive, traffic demands arrive arbitrarily. If users would avoid hitting a pause point, an altered value from a hard customized distributions feature described in Fourier transformations as flight time value $Y_K$and pause-time Y_Q as a random value in equation (9) :

$$E_K(y) = E_Q(y) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-jVy-|\rho V|^{\sigma}} DISTR \; dv \tag{9}$$

As inferred in equation (9), flight time and pause time has been derived. Where ρ is the factor of scale (pause period or flight-time) and σ_{DISTR} it depends on the distribution considered. User speed distribution is governed by a heavy-duty distribution of the user concerned traffic model. The proposed DRLTSM method has been proposed to achieve high efficiency, scalability, reliability, improve traffic scheduling, enhance system ability, network capacity, length of the queue.

4. Experimental Results and discussion

DRLTSM has been evaluated based on efficiency and scalability. To test Network Slice in optimising and working with various training methods and functionality, DRL creates network implementations. Six slices and three types of services are available in the result section. The network modifies implementations dynamically and chooses frame dimensions, for instance, 300x 300 and computer versions. DRLTSM use the traces of the network from phone service to reduce network traffic. In the evaluation of network slicing and traffic scheduling, the movement of blocks must be identified. In-network slicing, the total data traffic is received and used to transport network tranches under various geographical areas. The duration span of traffic scheduling for the calculation and the period is used to evaluate the efficiency, as shown in figure.6.

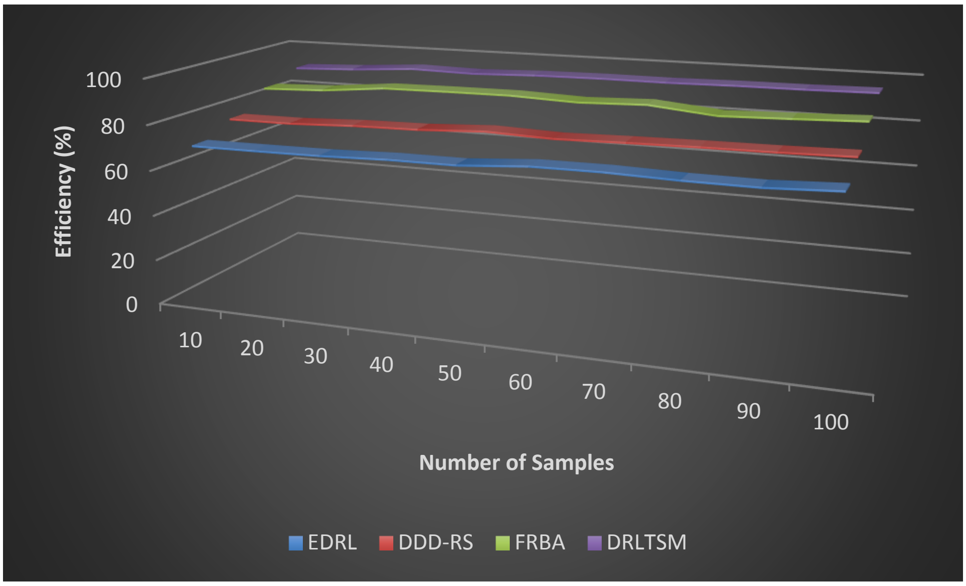

Figure.6. The efficiency of DRLTSM

In assessing the correlation among the number of slices and output, assess the developed system's scalability. The accumulation of assessed data is determined based on the number of network slices set during the development of the situation and the number of slices working simultaneously. The accumulation is listed as the network slices if the amount of slices is set to maximum. Still, the number of slices running at the same time evaluated based on the scalability. Suppose the amount of network slices increases, the system's efficiency is decreased by the greater capital requirement and the network slices' lower average assets. However, the device Slice is indeed capable of performing better than some others. The scalability of the proposed DRLTSM is shown in figure.7.

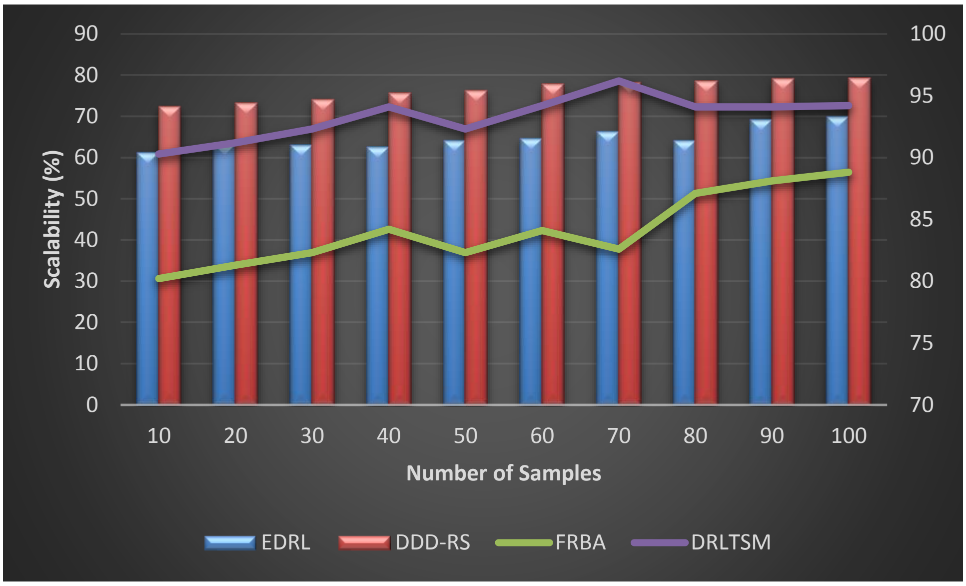

Figure.7. The scalability of the proposed DRLTSM

DRLTSM assess EdgeSlice efficiency under various network slice reliability functions. The significance of traffic in the output feature is different, and the reliability is shown in Figure.8. The broad document suggests a poorer output in the similar network lifetime for slice studies. Slice significantly outperformed everyone in all circumstances, meaning that Slice would learn advanced resource management policy immediately under different performance features. Besides, DRLTSM describes another feature as an unfavourable remaining contract of slice consumers without considering Slice wait traffic. The reliability function of DRLTSM is shown in figure.8.

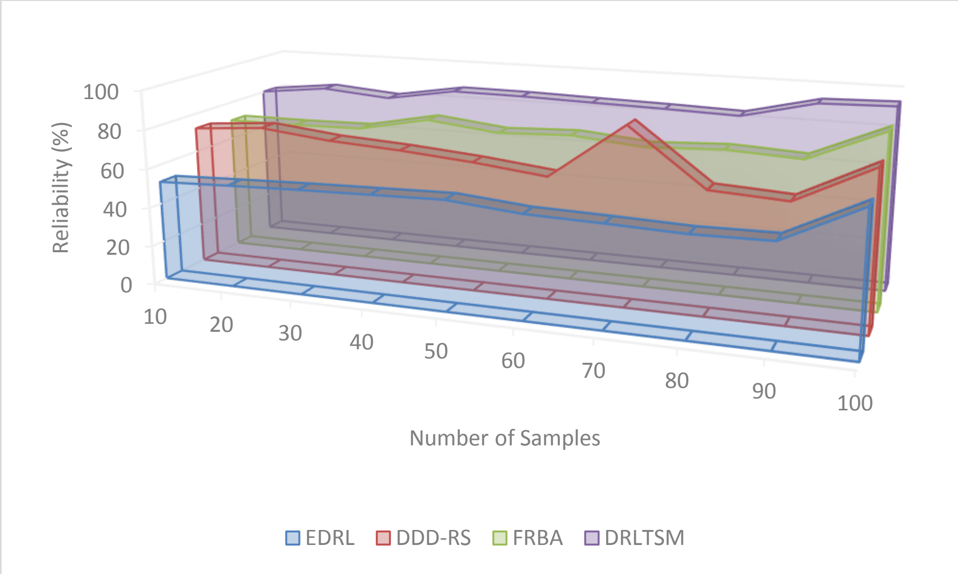

Figure. 8. The reliability function of DRLTSM

Network slice requests can vary from 5% to 25% of the total device power, while the length of applications may be from various stages. Each Slice determines the likelihood that an application in a period window enters the system. The access management procedure is called up at the start of each window in the network. Depending on prediction, the next time, the timeframe is given for the network dividing queries and corresponding traffic. However, the predicted data estimate the actual traffic rate leading to a network slice breach because of unpredictable traffic queries. The accuracy for the traffic scheduling rate is shown in Figure.9.

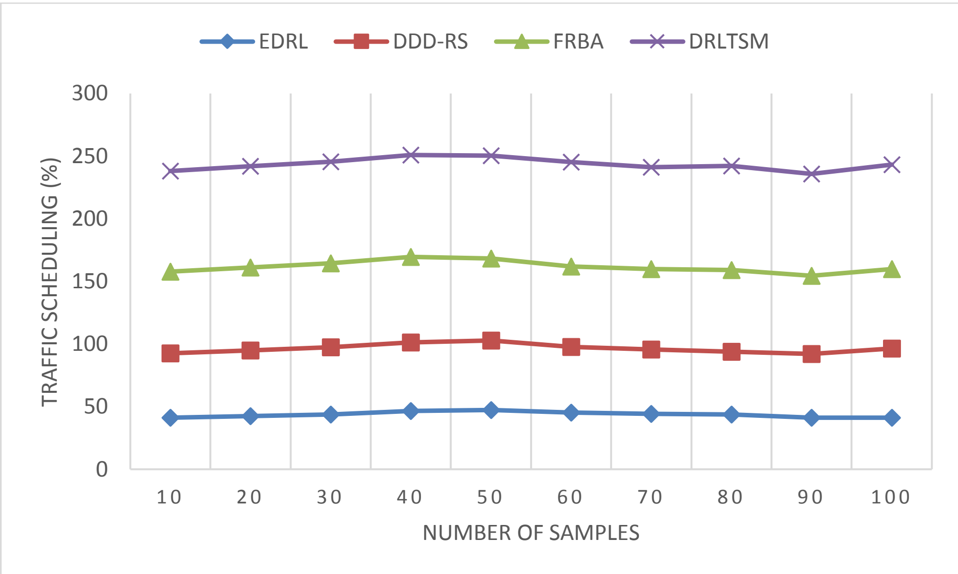

Figure.9. The accuracy for traffic scheduling rate

A variety of network models assess the efficacy of the forecasted knowledge. In each case, DRLTSM recognizes the decision to lease various parts of our usable capital, namely, the supplier of facilities between 100 and 250 networks. DRLTSM note that proportional gains are increased by the numbers of units and the system's ability. In reality, a tiny handful of rents include several network sections that can be easily compensated and have a small absolute profit. The system ability is shown in figure.10.

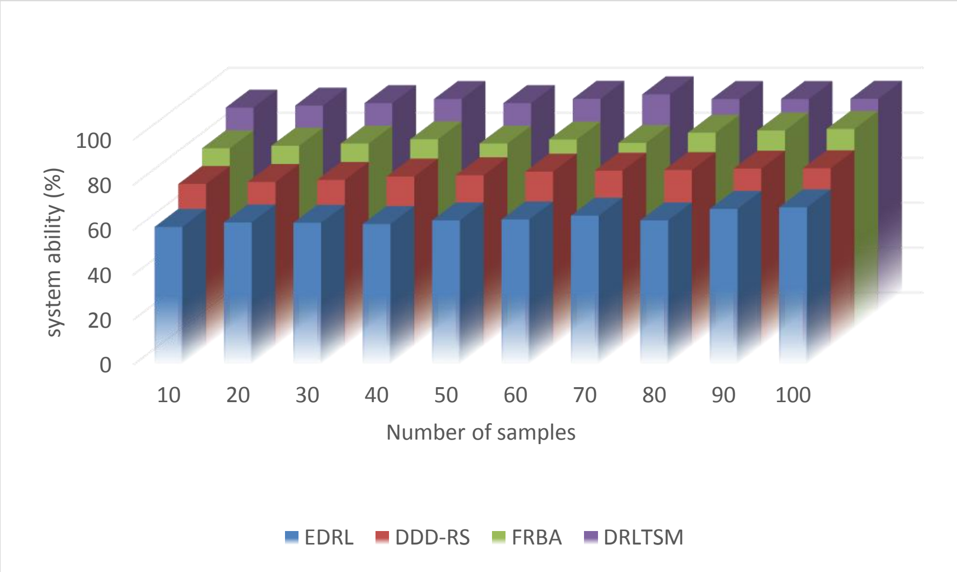

Figure.10. The system ability rate

Network slicing results are reported after study updates in a quantitative correlation among the traffic scheduling rate. From LSTM, a drop in slicing networks leads to reduced network capacity and increased possible crashes between the parts. The network capacity is re-allocated in line with the content yet action slice. When the traffic scheduling is altered and adequate transmission capacity and a scheduling time, it can be seen. Simultaneously, each Slice's assigned frequency is adapted per moment and therefore weaker to achieve fluctuations in request. The network capacity is shown in Table.1.

Table.1. The network capacity of DRLTSM

Number of samples	FRBA	EDRL	DRLTSM
20	51.3	65.2	80.2
40	52.6	66.2	80.8

60	53.8	67.1	80.9
80	54.7	68.2	81.4
100	55.6	65.3	82.3

The Network Slice exceeds those in all circumstances, meaning that each Slice may acquire superior resource management policy immediately under different output features. Besides, DRLTSM describes another feature as an unfavourable processing period of slice customers without considering slice queue activity. The queue length is used to determine the length of the slicing of the network. The length of the queue in slicing is shown in table.2.

Table.2. The length of the queue in slicing

Number of samples	FRBA	EDRL	DRLTSM
20	33.3	21.2	10.9
40	32	21.4	10.4
60	31.6	21	10.5
80	34.1	20.5	10.7
100	38.2	20.4	10.1

The proposed method achieves the highest efficiency when compared to other existing Exploiting Deep Reinforcement Learning (EDRL), Data-driven dynamic resource scheduling (DDD-RS), Flexible Resource Block Allocation (FRBA).

5. Conclusion

This paper presents DRLTSM to find new alternative measures and reinforce trends that are considered reasonable solutions, producing more rewarding results. The ground-breaking 5G-Network paradigm helps to cut off with multi-tenant help and promising industry possibilities. As an emerging business to operators, customised slices could be delivered to other clients at a cost difference. Remote resources are included in the DRL for network cutting situations that tackle power management and central network sizing. The goal of DRLTSM is to establish three key sliding blocks: (i) traffic assessment and network slice predicting; (ii) admittance management judgments for the network and (iii) dynamic predictive control adjustment based on measured variations. The experimental result obtained achieves the highest efficiency of 97.32% when compared to other existing.

Acknowledgment:

This research is supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R195), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

Reference

1. Xiong, Z., Zhang, Y., Niyato, D., Deng, R., Wang, P., & Wang, L. C. (2019). Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges. *IEEE Vehicular Technology Magazine*, *14*(2), 44-52.
2. Yang, S., Ma, W., Pi, X., & Qian, S. (2019). A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transportation Research Part C: Emerging Technologies*, *107*, 248-265.
3. Dragičević, T., Siano, P., & Prabaharan, S. R. (2019). Future generation 5G wireless networks for smart grid: A comprehensive review. *Energies*, *12*(11), 2140.
4. Qiu, J., Grace, D., Ding, G., Zakaria, M. D., & Wu, Q. (2019). Air-ground heterogeneous networks for 5G and beyond via integrating high and low altitude platforms. *IEEE Wireless Communications*, *26*(6), 140-148.
5. Banchs, A., Gutierrez-Estevez, D. M., Fuentes, M., Boldi, M., & Provvedi, S. (2019). A 5G mobile network architecture to support vertical industries. *IEEE Communications Magazine*, *57*(12), 38-44.
6. Walia, J. S., Hämmäinen, H., Kilkki, K., & Yrjölä, S. (2019). 5G network slicing strategies for a smart factory. *Computers in industry*, *111*, 108-120.
7. Gokulnath, C. B., & Shantharajah, S. P. (2019). An optimized feature selection based on genetic approach and support vector machine for heart disease. *Cluster Computing*, *22*(6), 14777-14787.

8. Barakabitze, A. A., Ahmad, A., Mijumbi, R., & Hines, A. (2020). 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. *Computer Networks*, *167*, 106984.
9. Baskar, S., & Dhulipala, V. S. (2018). Secure and compact implementation of optimized Montgomery multiplier based elliptic curve cryptography on FPGA with road vehicular traffic collecting protocol for VANET application. International Journal of Heavy Vehicle Systems, 25(3-4), 485-497.
10. Song, C., Zhang, M., Zhan, Y., Wang, D., Guan, L., Liu, W., ... & Xu, S. (2019). Hierarchical edge cloud enabling network slicing for 5G optical fronthaul. *Journal of Optical Communications and Networking*, *11*(4), B60-B70.
11. Molinos-Senante, M., Muñoz, S., & Chamorro, A. (2019). Assessing the quality of service for drinking water supplies in rural settings: A synthetic index approach. *Journal of environmental management*, *247*, 613-623.
12. Afolabi, I., Taleb, T., Frangoudis, P. A., Bagaa, M., & Ksentini, A. (2019). Network slicing-based customization of 5G mobile services. *IEEE Network*, *33*(5), 134-141.
13. Kotulski, Z., Nowak, T. W., Sepczuk, M., & Tunia, M. A. (2020). 5G networks: Types of isolation and their parameters in RAN and CN slices. *Computer Networks*, *171*, 107135.
14. Kumar, P. M., Babu, G. C., Selvaraj, A., Raza, M., Luhach, A. K., & Díaz, V. G. (2021). Multi-criteria-based approach for job scheduling in industry 4.0 in smart cities using fuzzy logic. Soft Computing, 25(18), 12059-12074.
15. Obaid, A., Hanna, M. E., Wu, Y. W., Kollo, M., Racz, R., Angle, M. R., ... & Melosh, N. A. (2020). Massively parallel microwire arrays integrated with CMOS chips for neural recording. *Science advances*, *6*(12), eaay2789.
16. Fu, X., Fortino, G., Pace, P., Aloi, G., & Li, W. (2020). Environment-fusion multipath routing protocol for wireless sensor networks. *Information Fusion*, *53*, 4-19.
17. Xiao, J., Chen, S., & Sui, M. (2019). The strategy of path determination and traffic scheduling in private campus networks based on SDN. *Peer-to-Peer Networking and Applications*, *12*(2), 430-439.
18. Liu, B., Zhang, Y., Zhang, G., & Zheng, P. (2019). Edge-cloud orchestration driven industrial smart product-service systems solution design based on CPS and IIoT. *Advanced Engineering Informatics*, *42*, 100984.

19. Datta, P. P. (2020). Hidden costs in different stages of advanced services–A multi-actor perspective of performance based contracts. *Journal of Business Research*, *121*, 667-685.
20. Selvaraj, A., Selvaraj, J., Maruthaiappan, S., Babu, G. C., & Kumar, P. M. (2020). L1 norm based pedestrian detection using video analytics technique. *Computational Intelligence*, *36*(4), 1569-1579.
21. Yan, M., Feng, G., Zhou, J., Sun, Y., & Liang, Y. C. (2019). Intelligent resource scheduling for 5G radio access network slicing. *IEEE Transactions on Vehicular Technology*, *68*(8), 7691-7703.
22. Wei, F., Feng, G., Sun, Y., Wang, Y., Qin, S., & Liang, Y. C. (2020). Network slice reconfiguration by exploiting deep reinforcement learning with large action space. *IEEE Transactions on Network and Service Management*, *17*(4), 2197-2211.
23. Abiko, Y., Saito, T., Ikeda, D., Ohta, K., Mizuno, T., & Mineno, H. (2020). Flexible resource block allocation to multiple slices for radio access network slicing using deep reinforcement learning. *IEEE Access*, *8*, 68183-68198.
24. Bega, D., Gramaglia, M., Garcia-Saavedra, A., Fiore, M., Banchs, A., & Costa-Perez, X. (2020). Network slicing meets artificial intelligence: an AI-based framework for slice management. *IEEE Communications Magazine*, *58*(6), 32-38.
25. Bega, D., Gramaglia, M., Fiore, M., Banchs, A., & Costa-Perez, X. (2019, April). DeepCog: Cognitive network management in sliced 5G networks with deep learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications* (pp. 280-288). IEEE.

Biography

1. Priyan Malarvizhi Kumar

Dr. Priyan Malarvizhi kumar is currently working in Department of Computer Science and Engineering, Kyung Hee University, South Korea. Before that he is Postdoctoral Research Fellow in Middlesex University, London, UK. He had completed his Ph.D. in the Vellore Institute of Technology University. He received his Bachelor of Engineering and Master of Engineering degree from Anna University and Vellore Institute of Technology University, respectively.

E-mail id: mkpriyan@khu.ac.kr

2. Shakila Basheer

Dr. Shakila Basheer is an Assistant Professor in Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint AbdulRahman University in Riyadh, Saudi Arabia. She has more than 10 years of teaching experience and has published more technical papers in international journals/ proceedings of international conferences/ edited chapters of reputed publications.

3. Bharat S. Rawal

Dr Bharat S. Rawal is full Professor & Chair Computer and Data Science at Capitol Technology University, USA. His research focuses on network security, cloud computing and security, blockchain, Big Data and analytical modeling, smart grid and health informatics for the development of next-generation cyber defense and operation technologies.

4. Fatemeh Afghah

Dr. Fatemeh Afghah is a tenured Associate Professor in the Department of Electrical and Computer Engineering at Clemson University. Prior to joining NAU, she was an Associate Professor (with tenure) in the School of Informatics, Computing and Cyber Systems at Northern Arizona University. She is an IEEE Senior Member and the Director of Intelligent Systems and Wireless Networking (IS-WiN) Laboratory.

5. Gokulnath C

Dr. Gokulnath C is currently working in the Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, India. He had completed his Ph.D. at the Vellore Institute of Technology. He received his Bachelor of Engineering and Master of Engineering degree from Anna University and VIT University, respectively.

6. Manimuthu Arunmozhi

Manimuthu Arunmozhi a Research Fellow working in the field of Autonomous Vehicle Security at CYBERSECURITY RESEARCH CENTER (CYSREN), Nanyang Technological University (NTU), Singapore. Ph.D. in the topic of AI, Machine Learning, IoT, and Embedded application under the faculty of Electrical Engineering in 2019. Bachelor's in Electrical Engineering and Master's in Embedded and Real-Time systems in the years 2013 and 2015 respectively.

Conflict of Interest:

There Is no conflict of interest between the authors.