IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Smart Reinforcement Learning Algorithms with Controlled-Complexity using Segmented and Recursive Methodology

[1] Modalavalasa Hari Krishna, [2] Makkena Madhavi Latha

[1] Full-time Ph.D Scholar, [2] Professor in ECE

[1,2] Dept. of Electronics and Communications Engineering,

[1,2] Jawaharlal Nehru Technological University-Hyderabad (JNTUH-CEH), Hyderabad, India

Abstract: Machine Learning algorithms play very crucial role in decision automation process of Artificial Intelligent systems. Machine Learning algorithms learn from the hidden structures present inside the data without the need of any traditional programming. Many advanced algorithms are developed for Machine Learning to handle complex datasets. These algorithms provide better performance but require huge amount of training time. This requirement of huge training time is manageable in Supervised and Unsupervised Machine Learning algorithms as their models are trained before deploying them into application. But, in Reinforcement learning, the effect of huge training time is very significant problem as their models are trained after deploying them into application environment. To solve this problem, new Reinforcement algorithms with controlled complexity need to be developed without compromising the performance of the model. This paper aims at four new Reinforcement Learning algorithms with controlled complexity to reduce the training time. The proposing algorithms are developed using MATLAB software and validated by employing them for automated parameter tuning in image denoising technique using Double Density Dual-tree Discrete Wavelet Transform. These proposing algorithms are compared against standard Markov Decision Process based Reinforcement Algorithm in terms of model accuracy and model training times.

Index Terms **- Complexity Controlled Learning, Double Density Dual-tree Discrete Wavelet Transform, Intelligent parameter tuning, Hybrid Thresholding, Segmented Recursive Reinforcement Learning, Segmented Adaptive Reinforcement Learning.**

I. INTRODUCTION

Machine Learning (ML) algorithms understands the data and explore the hidden structures and create models using which can handle the future data without need of manual coding [1]. Machine learning algorithms broadly classified into 3 categories as Supervised, unsupervised any reinforcement machining learning. Supervisor Machine Learning (SML) algorithms are very powerful among the all and provide best performance but requires pre-labelled data [2]. In case of lack of labelled data, Unsupervised Machine Learning (UML) algorithms can classify the available data by exploring the hidden relations inside it and train model which can classify the future data into same categories [3]. Supervised and unsupervised algorithms are not applicable if the statistical model of the environment is unknown or model response is not available until interacting with real environment. Reinforcement Learning (RL) algorithms can serve this situation which train the models after deployment. RL algorithms search for entire solution space and identify the optimal solution with best reward. Advanced RL algorithms are developed using the gaming theory and can serve in complex environments [1][4].

Reinforcement algorithms are further classified into two categories as Model-free and Model-based. Model-based algorithms such as Given-The-Model algorithm estimate the reward of next state before the calculation the next state values [5]. In Model-free algorithms like Q-notation next state values are calculated without any estimation of its reward [6].

II. OBJECTIVES AND GOALS

All RL algorithms provide better performance by searching the entire solution space in their own ways but searching entire solution space is very complex procedure [7][8]. Particularly in case of multi-dimensional solution space with more tuning parameters, these existing RL algorithms requires very large amount of training time to search the entire solution space. This training time can be reduced by increasing the step size but large step size reduces the accuracy of the model and miss the best reward [8]. The smaller steps provide precise optimal value but huge training times due to these small steps not suitable for real-time applications. Advanced hardware resources like Graphical Processing Units (GPUs) with high computational power can reduce the training time but training times of still these algorithms are large enough for complex applications like multidimensional parameter tuning and advanced image processing. To solve this problem, the complexity of the RL algorithms needs to be reduced in terms of number of iterations. The number of iterations have inverse relationship with the step-size in most of RL algorithms like Markov

Decision Processes (MDPs) [9][10][11]. To reduce the number of iterations without compromising the model accuracy, a novel segmented-recursive methodology is proposed in this paper.

III. METHODOLOGY

In this methodology algorithm first segments the entire solution space then starts searching solution space with gradient step. This step gradient is non-uniform and independent of reward value for entire training epoch. This intelligent training methodology is implemented in 4 different ways in order to serve different application. The 4 algorithms are Blind Segmented Recursive Reinforcement (SRRL-B) Algorithm, Unidirectional Segmented Recursive Reinforcement (SRRL-Ud) Algorithm, Bidirectional Segmented Recursive Reinforcement (SRRL-Bd) Algorithm, Blind Segmented Recursive Reinforcement (SARL) Algorithm. SRRL-Blind algorithm is the simplest form of Segmented Recursive methodology with controlled complexity and uniform gradient. SRRL-Unidirectional is similar to SRRL-Blind but the step is unidirectional incremental or decremental for all parameters. SRRL-Bidirectional is enhanced form of SRRL-Unidirectional with bi-directional incremental or decremental step gradient. All SRRL algorithms train the model with lesser iteration to provide more precise optimal value. For continuous training models, SRRL algorithms are best suited for initial training with a smaller number of iterations in complex solution space. But, after initial training SRRL is not required and simple algorithm is enough to maintain the optimal reward. For this purpose, SARL algorithm is developed which provides or track optimal value in continuous training models with very smaller number of iterations.

SRRL Algorithm

All the SRRL algorithms have same architecture except for few changes in gradient step calculation phase. So, common algorithm has two major steps such as segmentation and recursive loops. The main logic behind the controlled complexity is in travelling towards the optimal solution in each segment instead of travelling through entire solution space. The SRRL algorithm updates the step size after each epoch irrespective of reward in each segment as written below.

1. Start
2. Define tunable parameters
3. Set defaults solution space boundaries
4. Set SRRL Hyper parameters
5. Segment the Solution Space
6. Recursive Learning with Gradient
7. Set initial step Gradient, Parameters
8. Loop: Determine segment, step and parameters
    i. Interact with environment
    ii. Calculate reward
    iii. Update segment data
    iv. Update epoch data / direction flag
    v. Update step and parameters
9. Define Output values / Optimum Solution / Reward
10. Stop

SRRL Algorithm

SARL algorithm structure is almost similar to SRRL algorithms but as its name says, the step calculation is depending on best reward after each epoch. In SARL also updates the step once for each epoch in order to maintain uniform search in all segments. The SARL algorithm is given below.

1. Start
2. Define tunable parameters
3. Set defaults solution space boundaries
4. Set SARL Hyper parameters
5. Segment the Solution Space
6. Adaptive Learning with Gradient
7. Set initial step Gradient, Parameters
8. Loop: Determine segment, step and parameters
    i. Interact with environment
    ii. Calculate reward and differential reward
    iii. Calculate adaptive step in cyclic loop
    iv. Update segment data
    v. Update epoch data
    vi. Update step and parameters
9. Define Output values / Optimum Solution / Reward
10. Stop

Hybrid Thresholding based Image denoising using wavelet Transform (DDDT-DWT) method is considered as test application to verify the accuracy and training periods of designed algorithms. Hybrid Thresholding calculation has 3 independent parameters (Direct noise coefficient – Cd, Feed through coefficient-Cf, Gain controller coefficient -Cg) which decide the noise reduction performance of overall system [12].

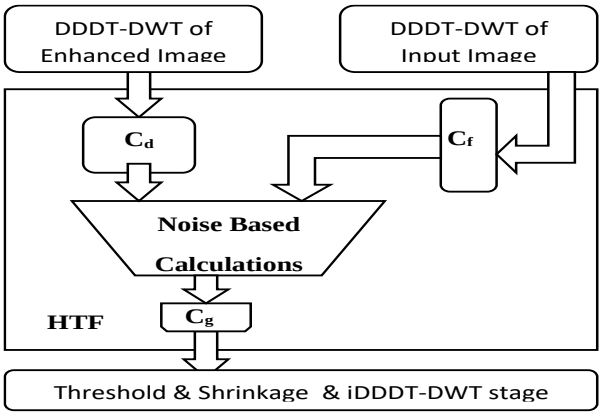

Fig 1. Block diagram of Noise based Hybrid Threshold factor Calculation

IV. Simulation and Results

The Base MDP RL algorithm and proposing SRRL and SARL algorithms are developed and simulated using MATLAB-R2020B software. The computational platform is a core-i7 Laptop with 8GB GTX 1660 Ti GPU card and 32GB DDR5 RAM. In this work, 10 different images have been considered as source images. Accuracy and training times of all algorithms are validated for Hybrid threshold-based noise reduction against input noisy image with three different noise types such as Gaussian, Speckle and Salt & Pepper noises and each at 6 different noise levels (variances levels 0.01, 0.05, 0.1, 0.3, 0.5, 0.8). All considered 10 test images used to validation of RL algorithms are shown in fig.2.

Fig.2. Test Images (10 different images with different test scenarios like low frequency, high frequency, high brightness, low brightness images from different application areas like medical field, satellite, etc.)

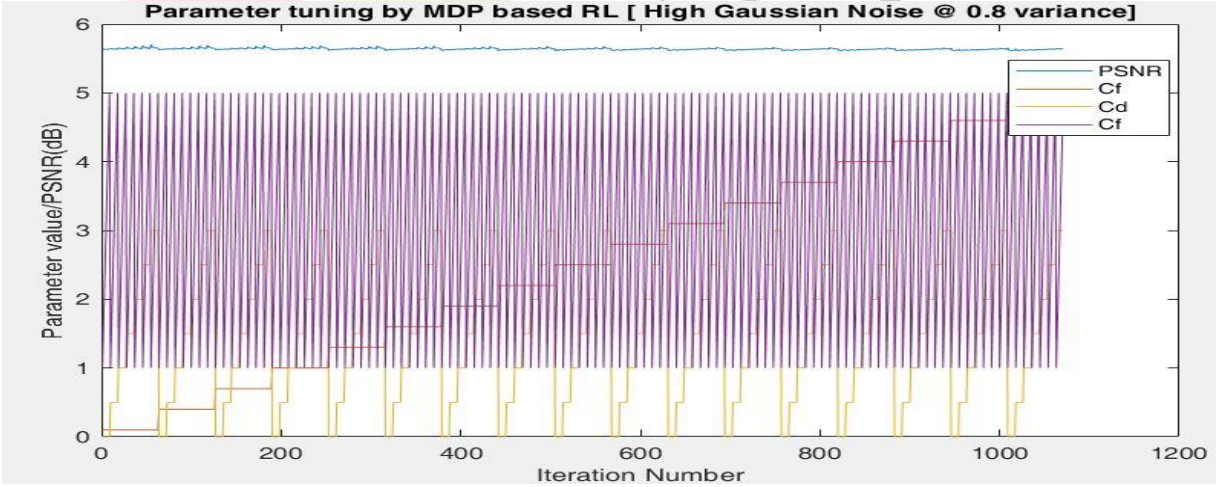

Fig.3. Training/ tuning of hybrid threshold factor parameters using standard MDP based RL algorithm

These 10 images with three different noise levels and each at 6 different noise levels forms 180 test instances. At each test instance all 5 RL algorithms are applied and performance parameters like best reward (Best PSNR), worst reward (worst PSNR), number of iterations and training times (Including image denoising procedure) are measured and analyzed.

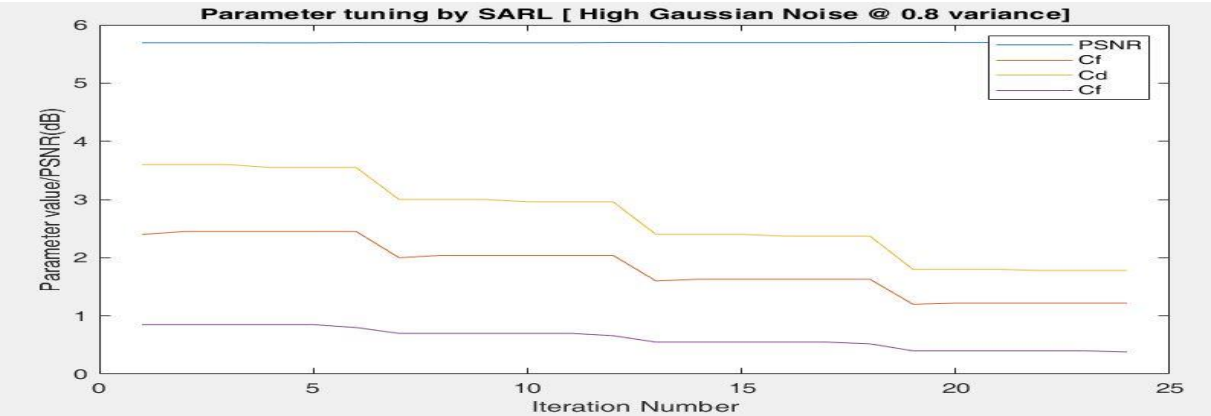

Fig.4. Training/ tuning of hybrid threshold factor parameters using SARL algorithm

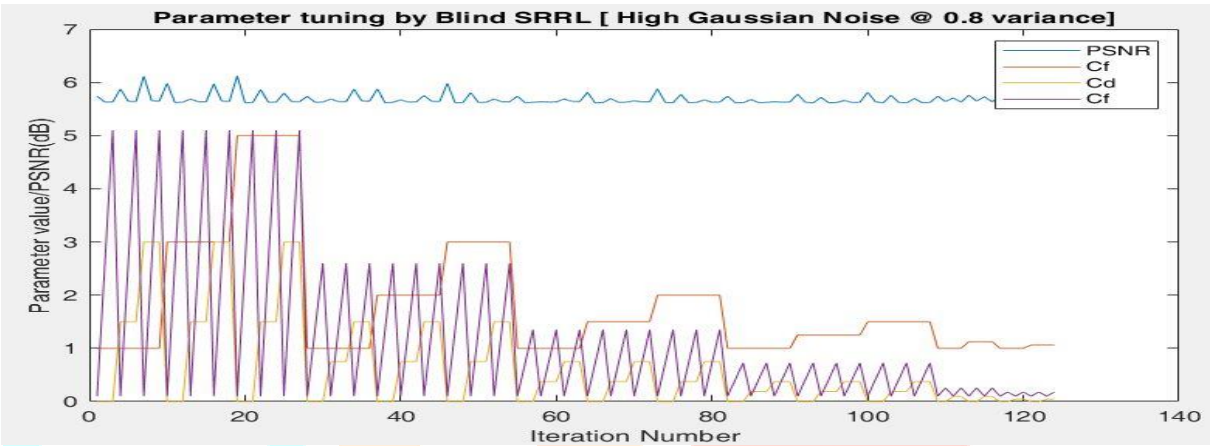

Fig.5. Training/ tuning of hybrid threshold factor parameters using Blind SRRL algorithm.

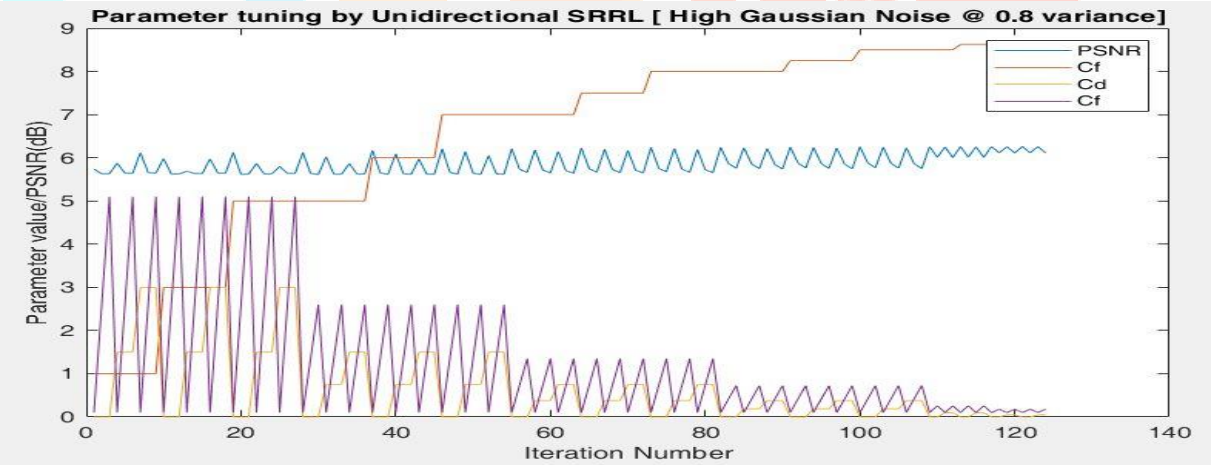

Fig.6. Training/ tuning of hybrid threshold factor parameters using Unidirectional SRRL algorithm

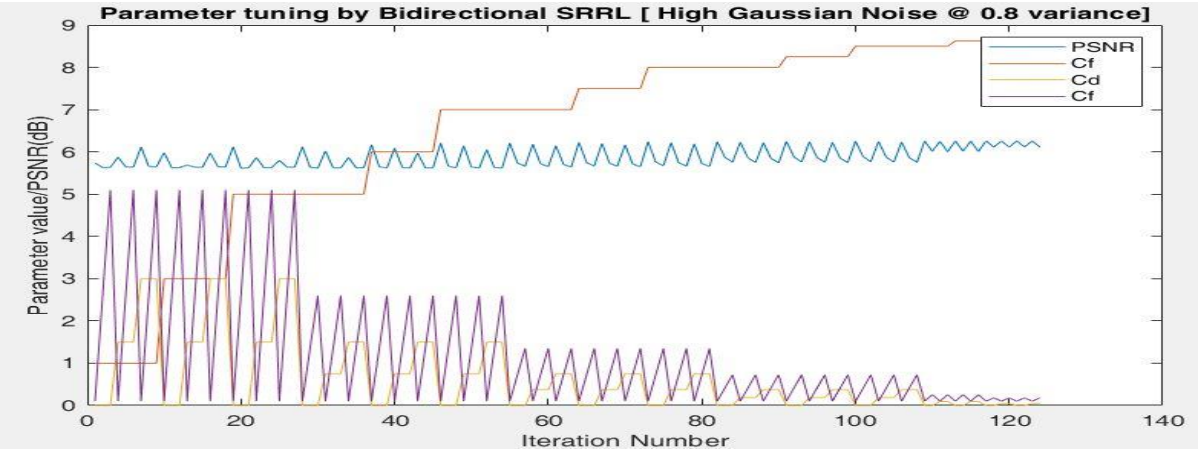

Fig.7. Training/ tuning of hybrid threshold factor parameters using Bidirectional SRRL algorithm

All the algorithms are applied on same image denoising application to tune 3 parameters of hybrid threshold factor and tuned states (parameter values) along with their reward (PSNR) are analyzed. Base MDP based RL algorithm search for entire solution space using the standard solution space using the tunable range of all 3 parameters. During the training process the variation in

reward (PSNR) is also varying and finally the best reward is determined after 1071 iterations as shown in fig.3. During the training period of SARL algorithm, the parameters are tuned very smoothly along the maximum reward (PSNR) so the variations in reward is very less and the tuning is gradually increasing in manner as shown in fig.4. SRRL algorithms tune the parameters in 124 iterations. The tuning process have variations in initial epochs and gradually smoothens the tuning and maximizes the reward by travelling towards the best rewards in all segments. But Blind SRRL algorithm, reward is not in increasing manner after each epoch as shown in fig-5 because of its blind step updating nature. Fig.6 and fig.7 show the smooth and directional training procedures along with their rewards in unidirectional and bidirectional SRRL algorithms respectively.

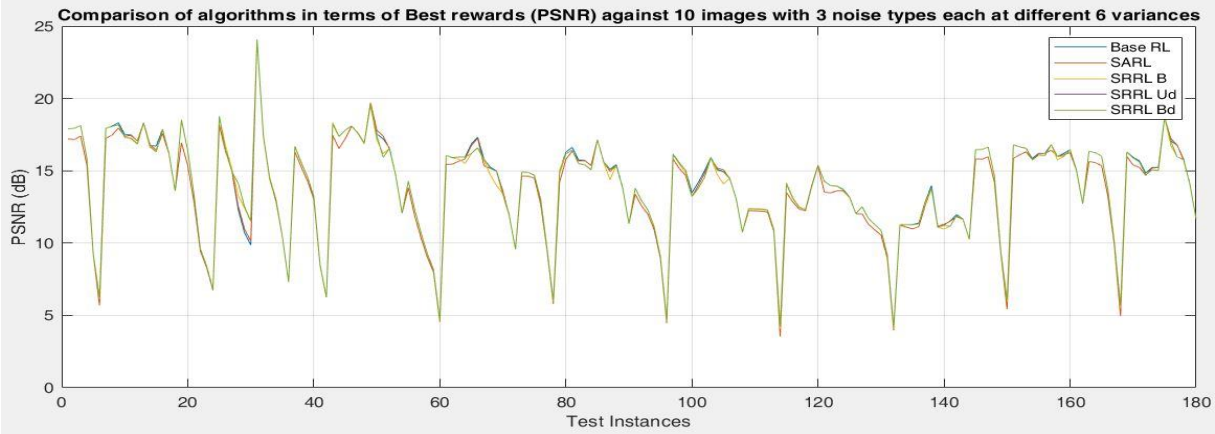

Fig.8. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm at each test instance.

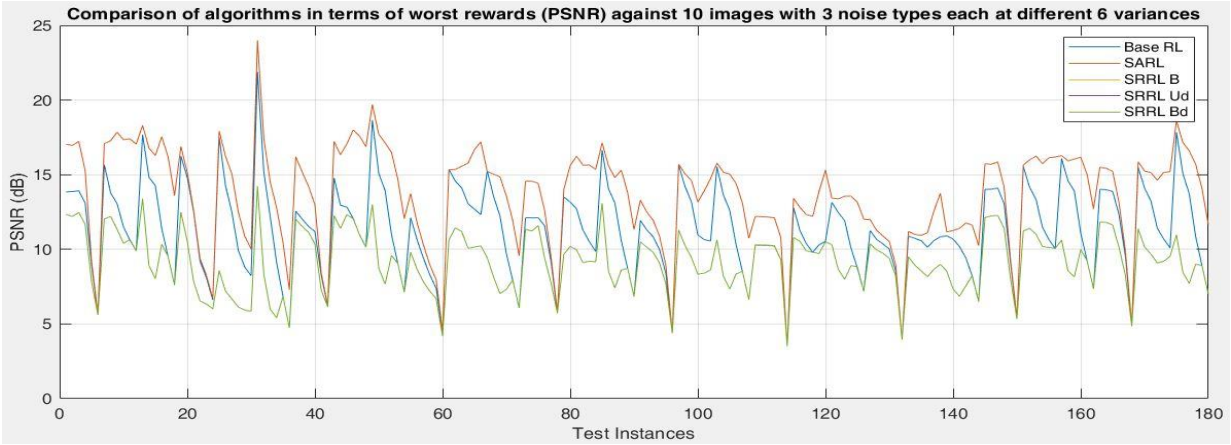

Fig.9. Comparison of all 5 RL algorithms in terms of worst reward provided by each algorithm at each test instance

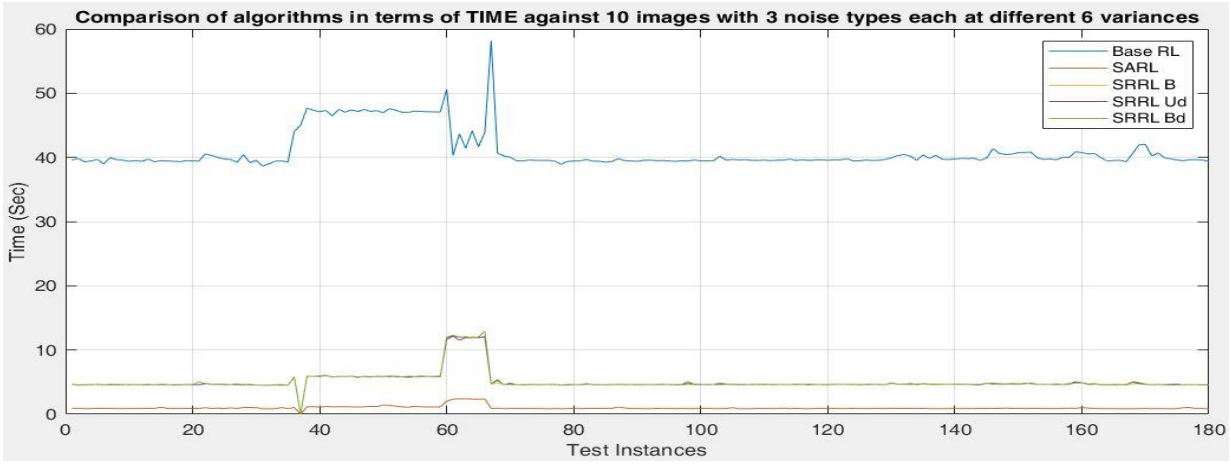

Fig.10. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm at each test instance

The existing MDP based standard RL algorithm learns and tunes all three control parameters in 1071 iterations up to one decimal point precision with step size 0.5 units. Whereas all the three SRRL algorithms (SRRL-B, SRRL-Ud, SRRL-Bd) tune the parameters more precisely up to 3 decimal points within 124 iterations. The SARL algorithm tunes all 3 parameters up to 3 decimal points precision within 30 iterations (most of times 24 iterations).

Fig.8 shows the best Rewards / PSNR values provided by all 5 algorithms at all 180 test instances. In all 180 test instances proposed SRRL algorithms exhibits equal or better performances than standard MDP algorithms. SARL algorithm also provides optimal or nearly optimal solutions in all cases. Worst case performance is also very important to analyze any algorithm. Fig.9 shows the worst-case performances from 5 algorithms in all 180 instances. Here SARL outperforms in all scenarios and proves its capability for continuous training of pre- trained models without losing the optimal track in short training period. SRRL algorithms also follow the standard MDP RL in most of cases. The training including the image denoising time is calculated at all instances.

Fig.10 shows total training times taken by all 5 algorithms during each test instance. MDP based RL took 40 seconds to 50 seconds whereas all SRRL algorithms took 4 to 5 seconds. Here SARL completed training within 2 seconds. The rise in time plot after 60 test instance is due to time taken to refresh the system RAM.

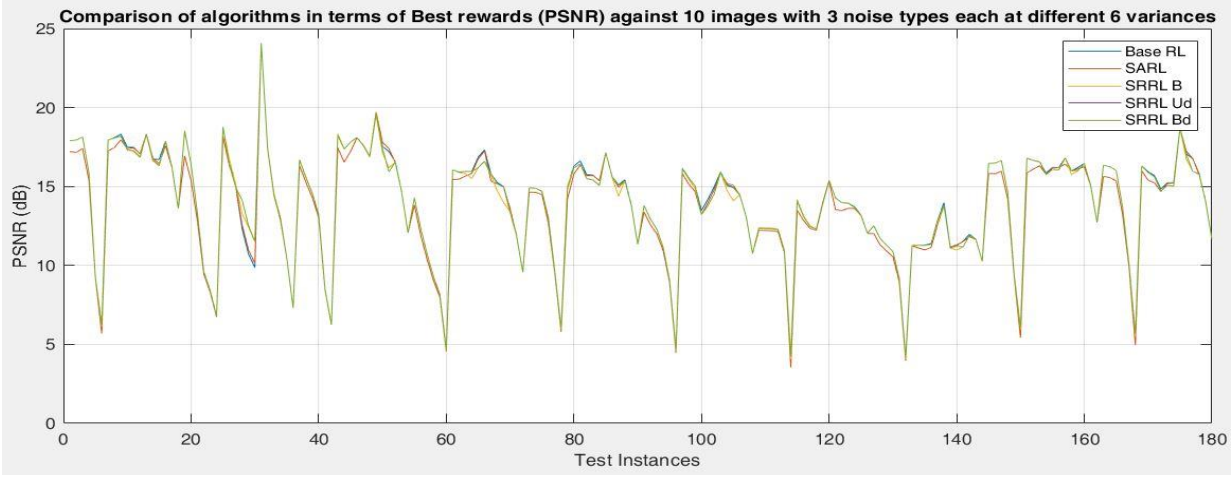

Fig.11. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm under Gaussian noise environment

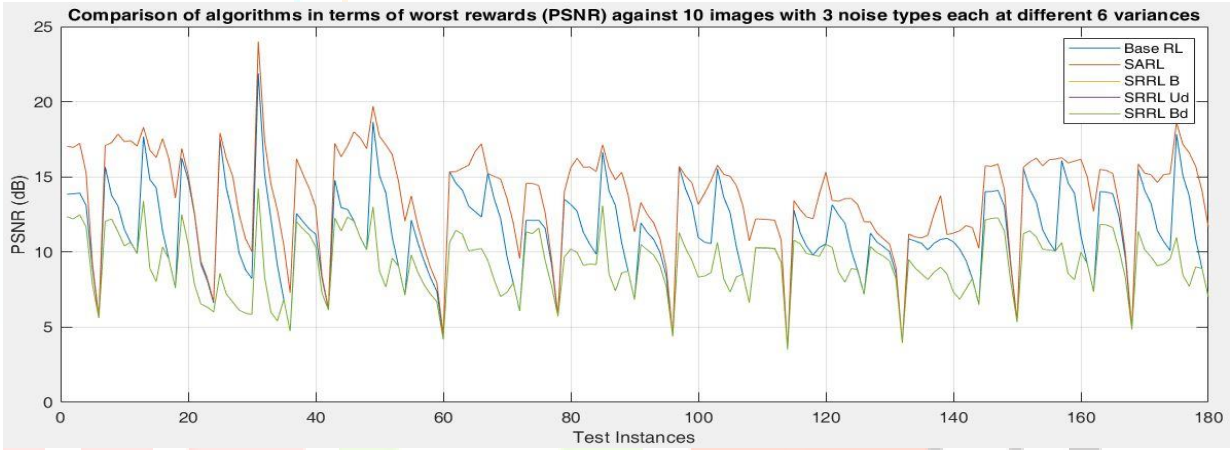

Fig.12. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm under Speckle noise environment

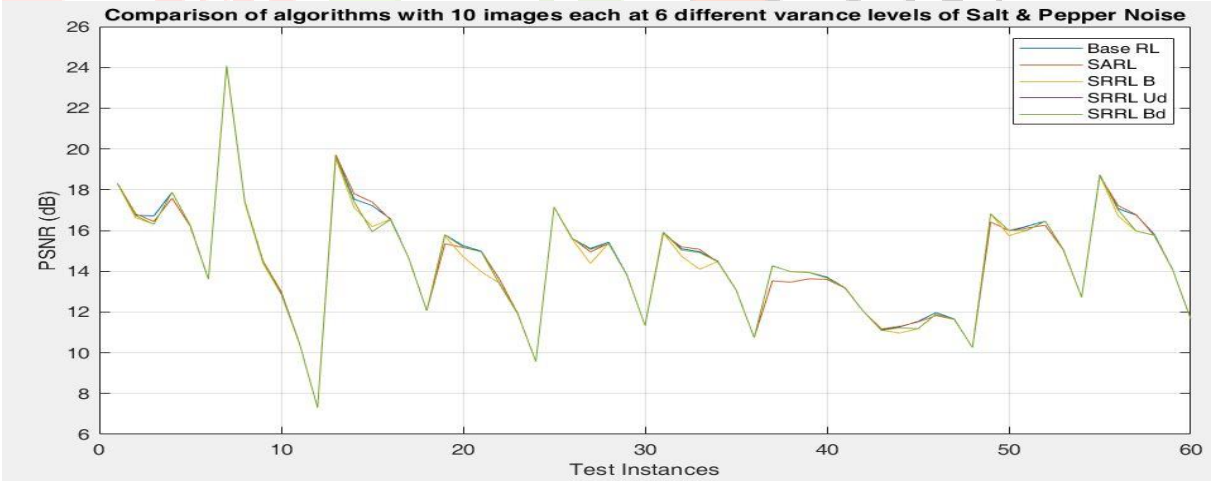

Fig.13. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm under gaussian noise

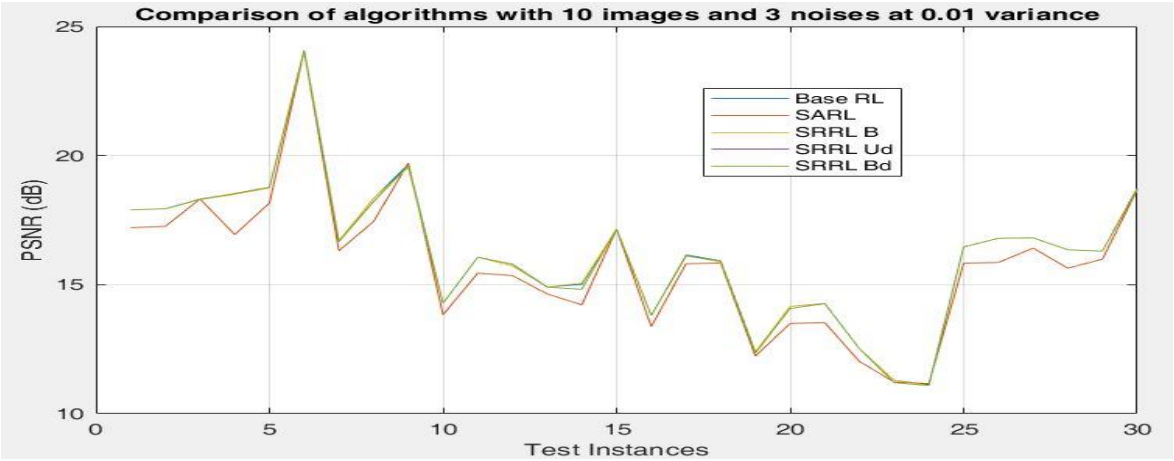

Fig.14. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm at noise variance 0.01 level

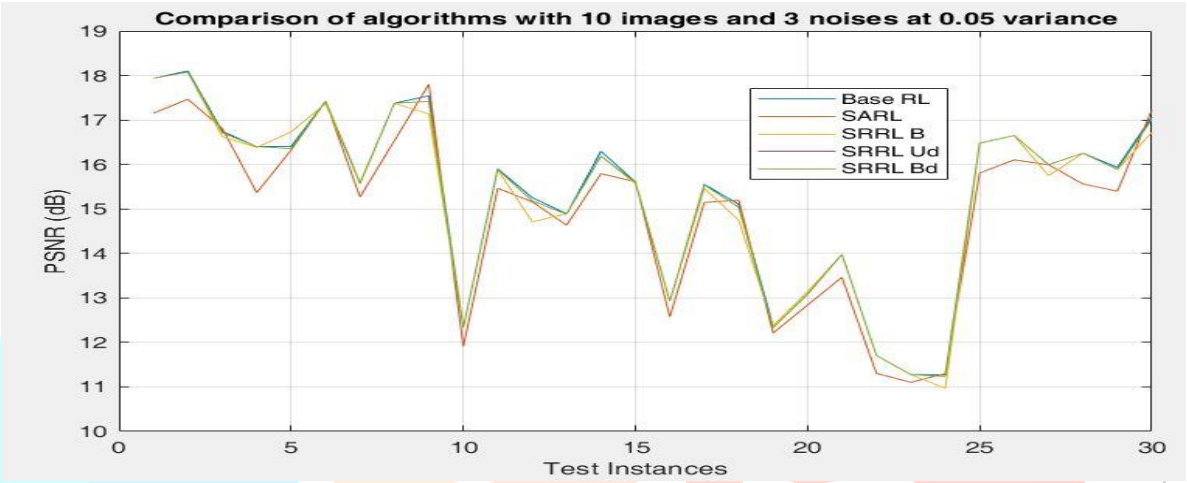

Fig.15. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm at noise variance 0.05 level

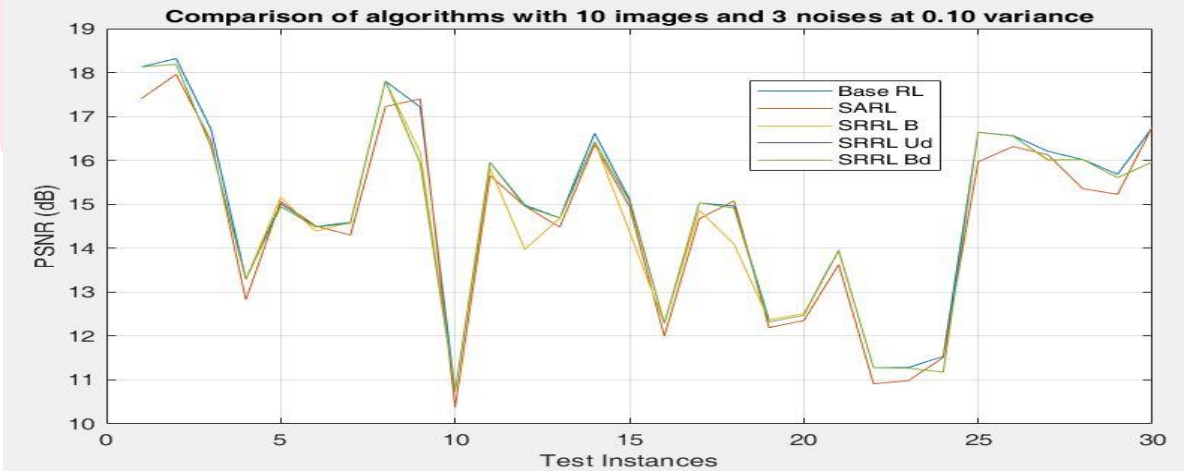

Fig.16. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm at noise variance 0.1 level

The algorithms are compared separately at each noise type (Gaussian, Speckle and Salt & Pepper). In all the cases the rewards from proposed algorithms are equal or higher than standard MDP based RL. Fig.11 shows the comparison of best rewards from 5 algorithms against Gaussian noise environment at various noise levels. Fig.12 and Fig.13 represent the same in Speckle noise and Salt & Pepper noise environments. For all considered 10 images at all noise levels the proposed algorithms outperform the standard MDP RL. To evident this statement these algorithms are compared at different noise levels separately. Fig.14 and Fig.15 represent the comparison at low noise levels such as 0.01 and 0.05 variance levels respectively. Fig.16 and Fig.17 represents the comparison of algorithms in terms of best rewards at medium noise levels at medium noise levels such as 0.1 and 0.3 variance levels and Fig.18 and Fig.19 represents the same in high noise environments with 0.5 and 0.8 variance levels respectively.

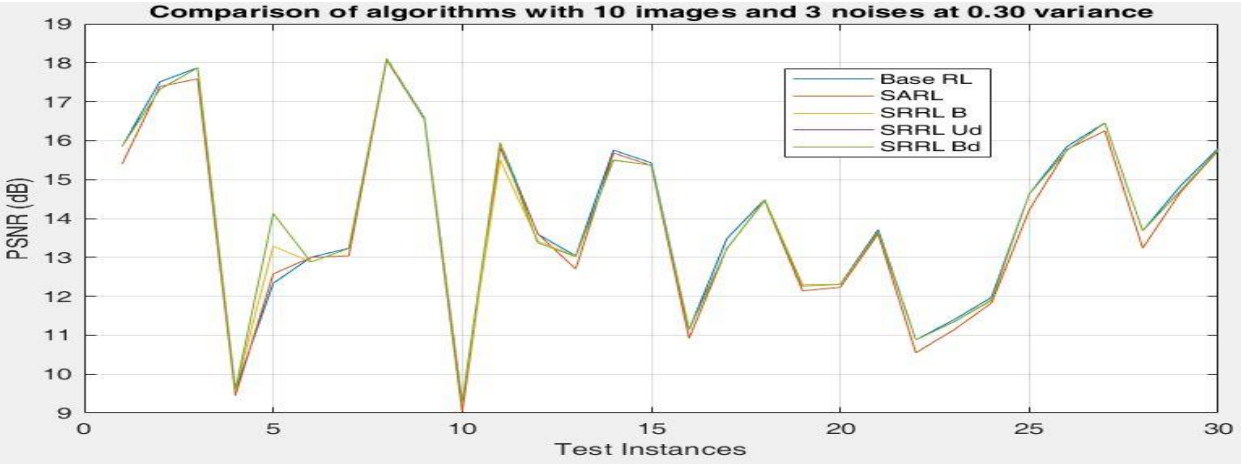

Fig.17. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm at noise variance 0.3 level

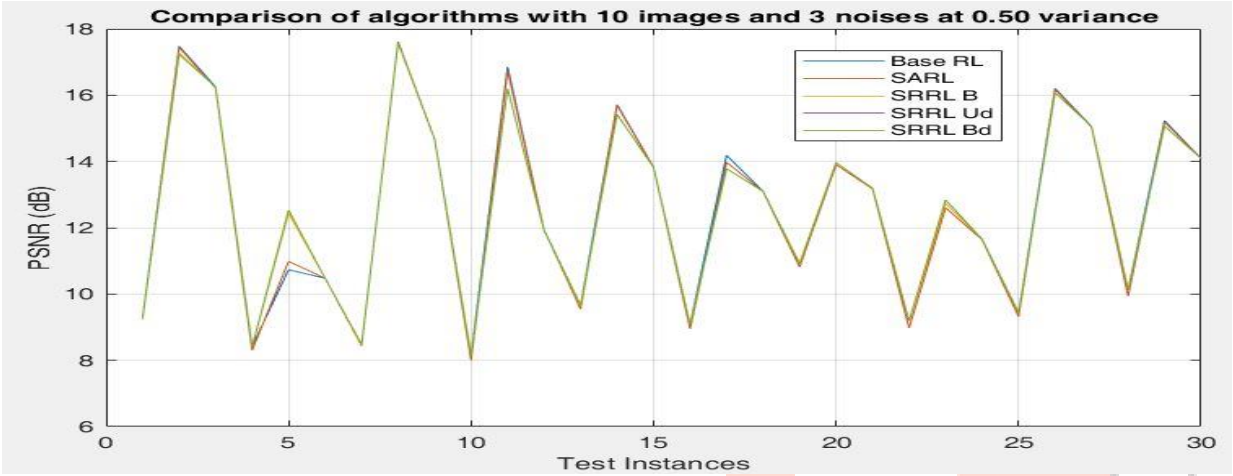

Fig.18. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm at noise variance 0.5 level

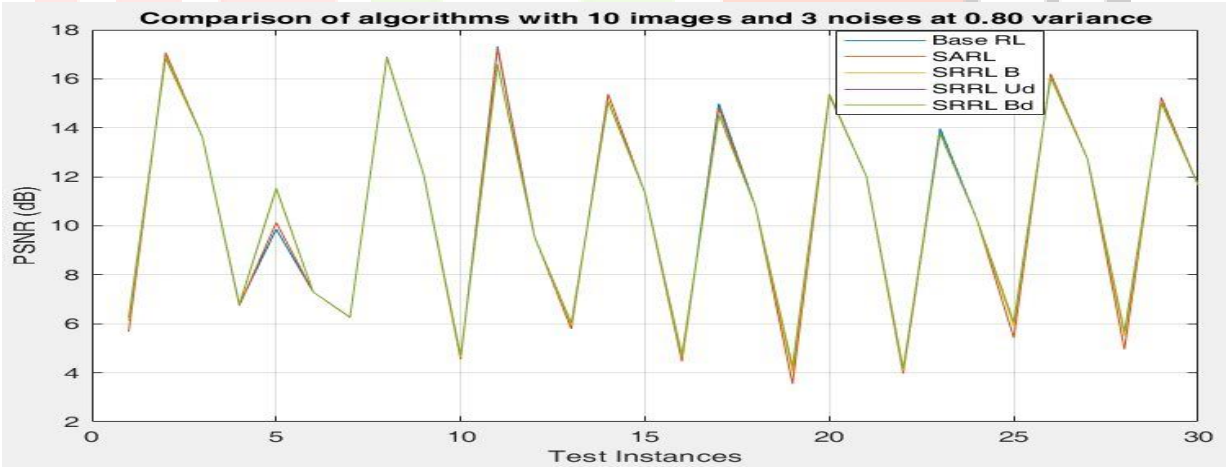

Fig.19. Comparison of all 5 RL algorithms in terms of best reward provided by each algorithm at noise variance 0.8 level

Table-I represents the training time required for all 5 algorithms and the controlled complexity of proposed segmented algorithms can be clearly observed in terms of number of iterations and training times. PSNR/Training time is value represents the advantage and importance of SARL algorithm with 13.85 units. SRRL algorithms also has more than 2.7 value where as standard MDP algorithm has 0.3 units. MDP RL taken around 50 seconds of training time with 1071 iteration whereas SRRL algorithms have taken around 5 seconds to complete 124 iterations of training and SARL algorithm has taken less than a second to train the model in 24 iterations. The performance of all 5 algorithms is analyzed in terms of mean, maximum and minimum PSNRs both in best and worst conditions as noted in table-II. In worst case condition SARL provides best performance and exhibit PSNR up to 24dB and average PSNR 13.71dB. In best case condition, Bidirectional SRRL algorithm exhibits best performance with reasonable training time. The SRRL algorithms are best suits for initial training of RL models and SARL algorithms best suits for continuous training of RL after pretrained models with SRRL algorithms.

TABLE I. COMPARISON OF TRAINING TIME

parameter	Reinforcement Learning Algorithms				
	MDPRL	***SRRL-B***	***SRRL-Ud***	***SRRL-Bd***	***SARL***
Iterations	1071	124	124	124	24
Training Time	43.0817	5.0508	5.0583	5.0749	0.9898
PSNR/Time	0.3231	2.7468	2.7492	2.7402	13.8521

TABLE II. PERFORMANCE COMPARISON IN TERMS OF PSNR

PSNR In dB		Reinforcement Learning Algorithms				
		MDPRL	***SRRL-B***	***SRRL-Ud***	***SRRL-Bd***	***SARL***
Best -Case performance in terms of PSNR	Mean	13.92	13.87	13.90	13.93	13.71
	Max	24.07	24.05	24.07	24.09	24.04
	Min	3.545	4.0612	4.171	4.173	3.542
Worst-Case performance in terms of PSNR	Mean	11.28	11.165	11.165	11.166	13.64
	Max	21.89	20.22	20.22	20.23	24.00
	Min	3.509	3.510	3.509	3.511	3.536

Note:
MDPRL = Markov Decision Process based Reinforcement Learning,
SRRL_B = Blind Segmented and Recursive Reinforcement Learning,
SRRL-Ud=Unidirectional Segmented and Recursive Reinforcement Learning,
SRRL-Bd=Bidirectional Segmented and Recursive Reinforcement Learning,

V. CONCLUSION

Novel Reinforcement Learning algorithms are designed using Segmented and recursive methodology with controlled complexity. SRRL algorithms provides better performance than MDP based standard RL algorithms with training time savings up to 90% in all the cases. SARL best suits for continuous algorithm with 98% lesser training time and exhibits best performance in worst case conditions. The SRRL algorithms are highly suitable for initial training of RL models. Proposed controlled complexity models meet the time requirements of the real time intelligent systems.

References

[1] John D. Kelleher; Brendan Tierney, "MACHINE LEARNING," in Data Science, MITP, 2018, pp.97-150.
[2] R. Saravanan, Pothula Sujatha, "A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification", Second International Conference on Intelligent Computing and Control Systems (ICICCS-2018), IEEE, doi: 10.1109/ICCONS.2018.8663155,2018.
[3] Alloghani M., Al-Jumeily D., Mustafina J., Hussain A., Aljaaf A.J. (2020) A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. In: Berry M., Mohamed A., Yap B. (eds) Supervised and Unsupervised Learning for Data Science. Unsupervised and Semi-Supervised Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-22475-2_1
[4] Jun Yu; Dacheng Tao, "Modern Machine Learning Techniques," in Modern Machine Learning Techniques and Their Applications in Cartoon Animation Research, IEEE, 2013, pp.63-104
[5] Polydoros, Athanasios & Nalpantidis, Lazaros. (2017). Survey of Model-Based Reinforcement Learning: Applications on Robotics. Journal of Intelligent & Robotic Systems. 86. 153-. 10.1007/s10846-017-0468-y.
[6] S. Çalışır and M. K. Pehlivanoğlu, "Model-Free Reinforcement Learning Algorithms: A Survey," 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 2019, pp. 1-4, doi: 10.1109/SIU.2019.8806389.
[7] Juan Cruz Barsce, Jorge A. Palombarini, Ernesto C. Mart´ınez , "Towards Autonomous Reinforcement Learning: Automatic Setting of Hyper-parameters using Bayesian Optimization", doi : 978-1-5386-3057-0/17,IEEE,2017.
[8] Chenyang Shen, Yesenia Gonzalez, Liyuan Chen, Steve B. Jiang, Xun Jia, "Intelligent Parameter Tuning in Optimization-based Iterative CT Reconstruction via Deep Reinforcement Learning", IEEE Transactions on Medical Imaging, doi: 10.1109/TMI.2018.2823679, 2018.
[9] V. Solo, "Selection of tuning parameters for support vector machines," Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., Philadelphia, PA, 2005, pp. v/237-v/240 Vol. 5
[10] Chenyang Shen, Yesenia Gonzalez, Liyuan Chen, Steve B. Jiang, Xun Jia, "Intelligent Parameter Tuning in Optimization-based Iterative CT Reconstruction via Deep Reinforcement Learning", IEEE Transactions on Medical Imaging, doi: 10.1109/TMI.2018.2823679, 2018
[11] Parag Kulkarni, "Reinforcement and Systemic Machine Learning for Decision Making", IEEE Books, Wiley-IEEE Press, doi: 10.1002/9781118266502, e-ISBN: 9781118266502, 2012.
[12] M. Hari Krishna, G. Sateesh Kumar, "A Noise Basesd Hybrid Thresholding For Enhanced Noise Reduction Using Double Density Dual Tree Complex Wavelet Transform", in 13th International Conference on Electromagnetic Interference and Compatibility (INCEMIC), 978-1-5090-5350-6/15, IEEE, 2015.