The general design of the automation for multiple fields using reinforcement learning algorithm

Vijaya Kumar Reddy Radha[1], Anantha N. Lakshmipathi [2], Ravi Kumar Tirandasu[3], Paruchuri Ravi Prakash[4]
[1]Department of IT, Lakireddy Bali Reddy College of Engineering (Autonomus), Mylavaram, India
[2]Department of CSE, Malla Reddy Engineering College (Autonomus), Secunderabad, Telangana, India
[3]Department of CSE, KoneruLakshmaiah Education Foundation, Vaddeswaram, India
[4]Department of IT, Prasad V. Potluri Siddhartha-Institute of Technology, Vijayawada, India

Article Info

Article history:

Received May 25, 2021
Revised Oct 25, 2021
Accepted Nov 21, 2021

Keywords:

AutoML
Computational graphs
Loss function
Recurrent neural network
Reinforcement learning

ABSTRACT

Reinforcement learning is considered as a machine learning technique that is anxious with software agents should behave in particular environment. Reinforcement learning (RL) is a division of deep learning concept that assists you to make best use of some part of the collective return. In this paper evolving reinforcement learning algorithms shows possible to learn a fresh and understable concept by using a graph representation and applying optimization methods from the auto machine learning society. In this observe, we stand for the loss function, it is used to optimize an agent's parameter in excess of its knowledge, as an imputational graph, and use traditional evolution to develop a population of the imputational graphs over a set of uncomplicated guidance environments. These outcomes in gradually better RL algorithms and the exposed algorithms simplify to more multifaceted environments, even though with visual annotations.

Corresponding Author:

Vijaya Kumar Reddy Radha
Department of IT, Lakireddy Bali Reddy College of Engineering (Autonomus)
Mylavaram, A. P, India
E-mail: Vijayakumarr285@gmail.com

1. INTRODUCTION

A long-standing goal of research into reinforcement learning is to blueprint of general purpose learning algorithms that can resolve an extensive array of issues. A probable resolution would be to devise a meta-learning technique that could model novel reinforcement learning algorithms that simplify to an extensive multiplicity of jobs automatically. In current years, automated machine learning (AutoML) has exposed huge success in automate the model of machine learning mechanism, such as neural networks architectures and design bring up to date rules [1], [2].

These previous procedures were intended for supervised learning but in reinforcement learning, there is additional mechanism of the algorithm that could be potential targets for model automation and it is not for all time clear with the best model, update process would be to put together these mechanism. Previous hard works for the computerization reinforcement learning algorithm detection have concentrate first and foremost on design modernize rules. These procedures learn the reinforcement learning update process itself and normally represent bring up to date rule with a neural network such as an recurrent neural network (RNN) or convolutional neural network (CNN), which can be professionally optimized with gradient-based techniques [3], [4].

There is only some profit of such an illustration. This demonstration is communicative enough to describe existing algorithms but also novel, undiscovered algorithms and also interpretable. This graph illustration can be analyze in the similar way as human intended reinforcement learning algorithms, making it more interpretable than procedures that use black box function equal for the entire reinforcement learning update process. If researchers can comprehend, why a learned algorithm is improved, then they can mutually adjust the domestic mechanism of the algorithm to develop it and transmit the helpful components to other issues. Finally, the demonstration supports general algorithms that can resolve a broad diversity of issues [5], [6]. Figure 1 shows the how reinfornmentlearning process on the raw data to generate requrienment outputs.

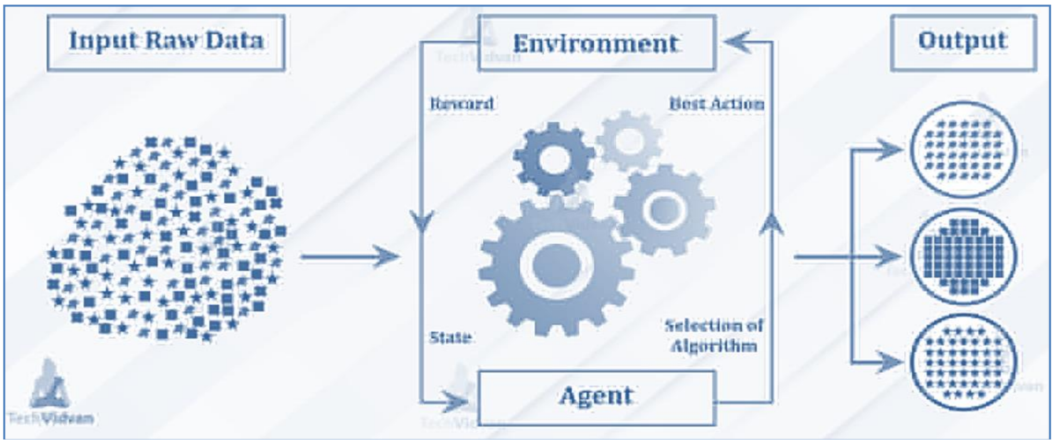

Figure 1. Reinforcement learning in machine learning

Reinforcement learning is part of learning technique in machine learning. They are supervised, unsupervised and reinforcement learning's [7], [8]. We discuss a reinforcement learning algorithm for an automatic prediction. The paper is presented in the following manner: the next section discuss the background analysis. Section 3 describes reinforcement learning algorithm mutation graph. An environmental reinforcement learning algorithm describes in section 4. Section 5 describes environmental learning algorithms and conclusion in section 6.

2. THE PROPOSED METHOD AND BACKGROUND ANALYSIS

In deep neural network, multi-agent types of environments are extremely dynamic, they impact on neighbors for alters rapidly. This operation is tough to learn the interpretation between the elements. The convolutional reinforcement learning graph accommodates the dynamics of the graph of the numerous agent environments, and this dynamic knowledge imprisons the relative between agents by their representation. Dormant features generated by convolutional layers from accessible fields are oppressed to learn teamwork; finally describe the proposed method substantially performs existing techniques in a diversity of cooperative scenarios [9], [10].

In recent years, reinforcement learning algorithms has gained rising attention and efforts to get better it have grown-up significantly. A set of measurements that quantitatively compute dissimilar aspects of reliability and we spotlight on variety and risk factor during training and after learning. These metrics are designed to be general purpose with statistical tests to allow meticulous comparisons on these metrics. We apply our metrics to a set of common reinforcement learning algorithms and their environments for comparison and analyze the output [11]. Generally, reinforcement learning agents with two crucial objectives. Primary one is to bring together obvious, revealing and scalable issues that imprison key problem intend of general and well-organized learning algorithms. The second objective to learn agent behaviour through their throughput on these communal benchmarks [12], [13].

In deep reinforcement learning algorithms handle with robust value functions for unprocessed clarification and rewards for model-free and model-based learning algorithms. In these algorithms, successor representations are decomposes the value function into 2 mechanisms; these mechanisms are reward predictor and successor map. The reward predictor maps describe to scalar rewards and the successor map presents the predictable future situation tenure from any given condition. In this concept, the value function of a condition can be calculated as the inner product between the map and the weights of reward points. Most of these types of algorithms used deep successor reinforcement learning (DSR) they generalize successor representations (SR) within a back-to-back deep reinforcement learning framework [14], [15].

2.1. Reinforcement algorithm as imputation graph

The memory and computation required for the Q-value algorithm would be too high. Thus, a deep network Q-Learning function approximator is used instead. This learning algorithm is called deep Q-network (DQN). The key idea in this development was thus to use deep neural networks to represent the Q-network and train this network to predict total reward. DQN is Q-learning of neural networks, the motivation at the back is merely connected to big state space environments where vital a Q-table would be a tremendously complex, difficult and protracted task. As an alternative of a Q-table neural networks estimated Q-values for each exploit based on the condition.

Graphs representing neural network architectures inspired by over the space, in reinforcement learning algorithms by on behalf of the loss function of a reinforcement learning algorithm as accusation as a directed acyclic graph for the loss function, with nodes on behalf of inputs, operators, parameters and outcome. For example, in the processing graph for DQN, input nodes contain data from the repeat barrier, operative nodes comprise neural network operators and fundamental math operators, and the outcome node represent the loss, which will be minimize with gradient descent. Figure 2 show how the squared Bellman Error will be used to get thr required output.

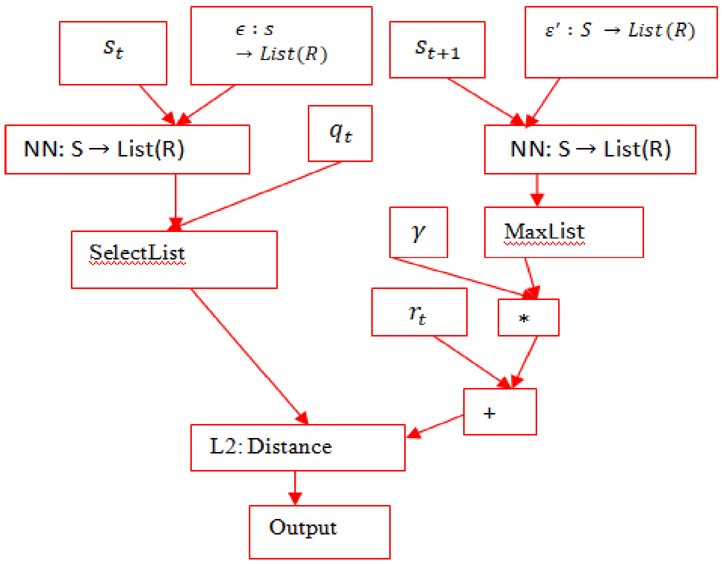

Figure 2. Example of squared Bellman error (16)

We can recognize, why a learned algorithm is enhanced, and then they can together adjust the domestic mechanism of the algorithm to advance it and transfer the helpful mechanism to additional issues. Finally, the illustration supports general algorithms that can resolve an extensive variety of issues in Figure 2 [16]. We developed this representation using the python PyGlove library, which appropriately turns the exceeding graph into a investigate space that can be optimized with standardize development into (1).

$$L_{DQN} = (Q_{\varepsilon}(S_t, a_t) - (r_t + \gamma * max_a\, Q_{\varepsilon'}(S_{t+1}, a)) \quad (1)$$

Model-based reinforcement learning has a actually influential from control theory, and the intention is to graph through an f(s, a) control function to choose the most excellent probable actions. It is similar as reinforcement learning fields where the laws of physics are contributed by the originator. The difficulty of model-based methods is that although they have extra supposition and estimate on a particular job, but may be incomplete only to these correct types of tasks. There are two main approaches: learning the model or learn given the design.

2.2. Environment of reinforcement learning

We utilize an evolutionary based procedure to optimize the reinforcement learning algorithms of attention [17], [18]. First, we initialize a populace of training agents with randomized graphs. This populace of agents is trained in equivalent over a set of training environments. The agent's first train on a difficulty environment projected to rapidly out with poor performing issues. If an agent can't crack the difficulty environment, the training is stopped up near the beginning with a score of zero. Otherwise, the training proceeds to more hard environments. The algorithm throughput is evaluated and used to bring up to date the populace, where more talented algorithms are further mutated. To decrease the search space, then we use a functional correspondence manager which will bounce over recently projected algorithms. These algorithms are same as previously practical examined algorithms. This loop continues as novel mutate agent algorithms are trained and evaluate. At the ending of training, we choose the most excellent algorithm and appraise its throughput over a set of hidden test environments. Figure 3 show how the the meta-learning metod will be used for training and testing it for multiple times.

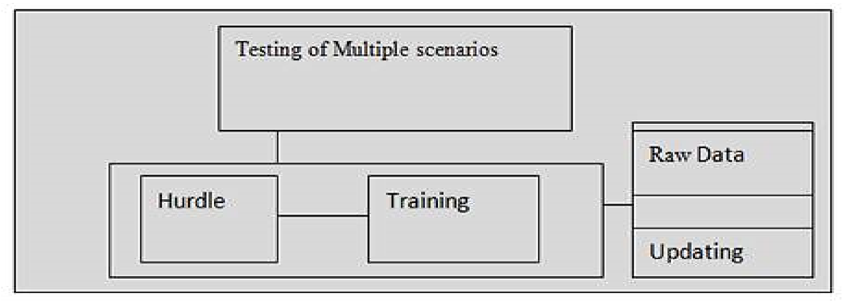

Figure 3. Overview of meta-learning method

3. METHODOLOGY AND RESULTS

We expose two finding algorithms that show high-quality generalization throughput. The primary deep algorithm is DQNReg, which build on DQN by addition heaviness on the Q-values based on standard squared Bellman error [19]. The next learned loss function, DQNClipped, is additional multifaceted, and it's dominate term has a straightforward form-the maximum of the Q-value and the squared Bellman error that means modulo a constant. Two algorithms can be view as a manner to normalize the Q-values. While DQNReg add a soft constriction, DQNClipped can be interpreting as a type of constrained optimization that will reduce the Q-values, if they become too hefty. We demonstrate that this learned constriction kicks in during the near the beginning phase of training when overestimate the Q-values is a potential problem. Once this constraint is pleased, then the loss will minimize instead of the original squared Bellman error [20], [21].

The following algorithm DQNReg for reinforcement learning in better way to analyze the accurate predictions. Along with this algorithm, other algorithms are working for better learning for accurate estimation of values. These reinforcement learning algorithms every step updates the data using buffer space. Check the samples for compute the target values, for this we perform the gradient descent step and update the target network parameters. This learning consists of multiple concepts for accurate learning for generating the accurate results for end users. The following algorithm given reinforcement learning of DQNReg.

Algorithm for DQNReg

```
Step 1: Initialize the networks with buffer
Step 2: for each iteration do
Step 3: for each environment step do
Step 4: Observe the state of the element and then select
Step 5: Execute that state and move to next state
Step 6: Store the information in buffer
Step 7: for each update step do
Step 8: Check the samples
Step 9: Compute the target Value
Step 10: Perform Gradient descent step
Step 11: Update the target network parameters
Step 12: end
```

A quicker analysis shows that while fundamentallines like DQN frequently overestimate R-values, our learned algorithms deal with this problem in dissimilar methods [22]. DQNReg underestimate the R-values, while DQNClipped has alike performance todouble DQN in that it gradually slows procedures the ground reality without overestimating it. We demency a dataset of top 2000 performing algorithms exposed during progress. Inquisitive reader could further examine the property of these learned loss functions. Our technique learns algorithms that have establish a way to regularize the Q-values and thus decrease overestimation [23], [24]. Figure 4 is used to show the results of minigrid-doorkey for value based RL method.

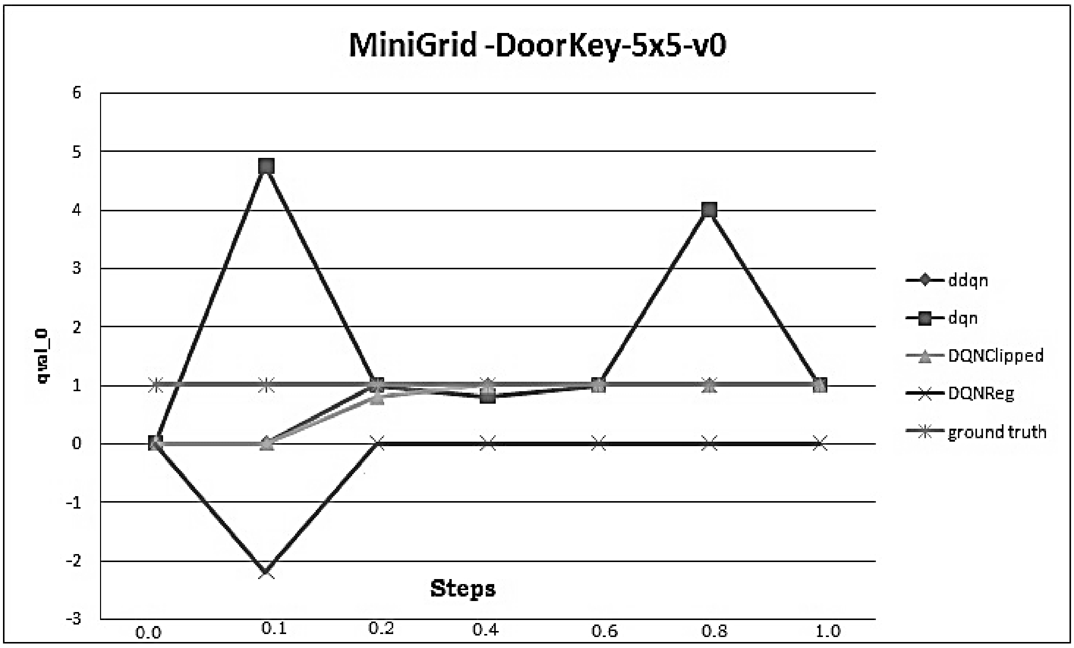

Figure 4. Overestimated values issues in value-based RL

Even on game based environments, we observe better throughput, even though training was onimage-based environments. This suggests that meta-training on a set of simple but diverse training environments with a generalized algorithm illustration could enable radical algorithmic generalization [25], [26]. Thus, it is a generalization of multiclass classification, where the classes involved in the problem are hierarchically structured, and each example may simultaneously belong to more than one class in each hierarchical level, e.g., multi-level text classification.

Table1 show the tabular format of the results of different performances of DQNReg against the games that tested aginst the baselines. The reinforcement learning algorithms using in different aspects of automation against multiple fields of multiple games with different environment. Mostly recent days using rapid changes occurred during the generation of accurate results. They can more helpful to the future development of automation system.

Table 1. Performance of DQNReg, against baselines on several games

Environment	DQN	DDQN	PPO	DQNReg
Space game	1464.5	754.7	2197.3	2490.2
Tenpin bowling	52.4	69.1	42.1	81.5
Kick Boxing	89.0	92.5	95.6	101.0
Running Race	40544.0	45127.0	35496.0	65816.0

4. CONCLUSION

In this paper, we learn novel accountable Reinforcement learning algorithms by on behalf of loss functions as computation graphs and enhance of agents over this progression. The computation graph formulation obey to both construct upon human-designed and learned algorithms using the same statistical toolset of extant algorithms. We analyzed a not more than of the learned algorithms and can construe them as a form of entropy regularization to avoid value of overestimation. These learned algorithms can perform fundamentallineslines and facilitate to hidden environments. We hope that future work will extend to more diverse of reinforcement learning algorithms settings such as actor critic algorithms.

REFERENCES

[1] S. Shaik, "Analysis of Photo Plethysmography Signals with Artificial Neural Networks Using Curvelet Transform," *Journal of Advanced Research in Dynamical & Control Systems*, vol. 11, no. 1, pp. 363-370, 2019.

[2] S. Shaik, "Pattern Recognition of Speech Signals Using Curvelet Transform and Artificial Intelligence," *International Journal of Creative Research Thoughts*, vol. 6, no. 2, pp. 718-723, April 2018, doi: 10.6084/m9.doi.one.IJCRT1892450.

[3] Ch. Shravya, Pravallika, and S. Shaik, "Prediction of Breast Cancer Using Supervised Machine Learning Techniques," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 6, pp. 1106-1110, April 2019.

[4] J. S. Keerthan, Y. Nagasai, and S. Shaik, "Machine Learning Algorithms for Oil Price Prediction," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 8, pp. 958-963, June 2019.

[5] S. Butada and S. Shaik, "IPL Match Prediction using Machine Learning," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 3438–3448, April 2020.

[6] Mamatha, S. Datta, and S. Shaik, "Fake Profile Identification using Machine Learning Algorithms," *International Journal of Engineering Research and Applications*, vol. 11, no. 7, pp. 60-65, July-2021.

[7] K. P. Surya Teja, V. Reddy, and S. Shaik, "Flight Delay Prediction Using Machine Learning Algorithm XGBoost," *Journal of Advanced Research in Dynamical & Control Systems*, vol. 11, no. 5, pp. 379-388, 2019.

[8] W. Qiang and Z. Zhongli, "Reinforcement learning model, algorithms and its application," *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 2011, pp. 1143-1146, doi: 10.1109/MEC.2011.6025669.

[9] J. Jiang, C. Dun, T. Huang, and Z. Lu, "Graph Convolutional Reinforcement Learning," *International Conference on Learning Representations*, 2020, pp. 1-13.

[10] J. A. Salvador, "Reinforcement learning: a literature review," Thesis, Computer Science and Business Management, Iscte - Instituto Universitário de Lisboa, Lisbon, Portugal, pp. 1-36, 2021.

[11] S. C. Y. Chan, S. Fishman, J. Canny, A. Korattikara, and S. Guadarrama, "Measuring the reliability of Reinforcement Learning Algorithms," *International Conference on Learning Representations*, pp. 1-36, 2020.

[12] I. Osband *et al.*, "Behaviour Suite for Reinforcement Learning," *International Conference on Learning Representations*, 2020, pp. 1-19.

[13] C. D'Eramo, D. Tateo, A. Bonarini, M. Restelli, and J. Peters, "MushroomRL: Simplifying Reinforcement Learning Research," *Journal of Machine Learning Research*, pp.1-5, 2021.

[14] T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman, "Deep Successor Reinforcement Learning," *Cornell University*, 2016.

[15] P. Klink, H. Abdulsamad, B. Belousov, C. D'Eramo, J. Peters, and J. Pajarinen, "A Probabilistic Interpretation of Self-Paced Learning with Applications to Reinforcement Learning," *Journal of Machine Learning Research*, pp. 1-52, 2021.

[16] R. Bellman, "On the Theory of Dynamic Programming," *Proceedings of the National Academy of Sciences America*. U. S. A., vol. 38, no. 8, 1952, pp. 716-719, doi: 10.1073/pnas.38.8.716.

[17] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction," *2nd edition. Cambridge, MA: MIT Press*, 2017.

[18] V. Krishnamurthy and G. Yin, "Langevin Dynamics for Adaptive Inverse Reinforcement Learning of Stochastic Gradient Algorithms," *Journal of Machine Learning Research*, pp. 1-49, 2021, doi: 10.1109/TAC.2021.3079280.

[19] D. Silver, "Deep reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2016.

[20] R. E. Bellman and S. E. Dreyfus, "Applied Dynamic Programming," *Annals of Mathematical Statistics*, vol. 33, no. 2, pp. 719-726, 1962, doi: 10.1214/aoms/1177704593.

[21] Y. Nasir, J. He, C. Hu, S. Tanaka, K. Wang, and X. Wen, "Deep Reinforcement Learning for Constrained Field Development Optimization in Subsurface Two-phase Flow," *Cornell University*, 2021, doi: 10.3389/fams.2021.689934.

[22] D. Silver *et al.*, "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm," *arXiv preprint arXiv:1712.01815*, *London*, 2017.

[23] A. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st edition. Cambridge, MA: MIT Press, 2016.

[24] S. Shaik, "DM Algorithms Based Clustering for Road Accident Data Analysis," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 9, pp. 160-167, Sept 2018, doi: 10.26438/ijcse/v6i9.160167.

[25] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *Spriner SN Computer Science*, vol. 2, no. 3, pp. 1-21, 2021, doi: 10.1007/s42979-021-00592-x.

[26] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484-489, 2016, doi: 10.1038/nature16961.

BIOGRAPHIES OF AUTHORS

Vijaya Kumar Reddy Radha Associate Professor in Information Technology at Lakireddy Bali Reddy College of Engineering state of Andhra Pradesh. Started is carrier in teaching from 2011 and worked as Assistant Professor in Computer Science and Engineering at Mother Theresa Educational Society Group of Institutions, Nunna in the state of Andhra Pradesh. And later worked as Assistant Professor in Electronics and Computer Engineering and Information Technology at Prasad V Potluri Siddhartha Institute of Technology from May 2012 to August 2021. He recivied is B.Sc. in 2005 from Acharya Nagarjuna University, M.C.A in 2009 from JNTU, Kakinada and M.Tech Information Technology in 2011 from Vignan University and Stood University Topper and PhD Computer Science and Engineering in 2019 from Acharya Nagarjuna University. He published 6 Patents and He has authored 34 International Journal Publications and Conferences of which 12 are Scopus Indexed and 22 UGC Indexed. His publications have a total of 120+ Google Scholar citations with h-index-7, i10 index-4. He is also rendering his services as a reviewer for two SCI/Scopus Indexed Journals. He is Life Member in ISTE, ISSE. He can be contacted at email: vijayakumarr285@gmail.com.

Anantha N. Lakshmipathi currently working as Head and Professor of CSE Department, Malla Reddy Engineering College (Autonomous), completed Ph.D. from Acharya Nagarjuna University. My Ph.D. Work was on A Deep Learning Framework based on Convolution Neural Network for Product Recommendation. My research interests are Machine Learning, Deep Learning in the field of Recommender Systems. My recent work focuses on application of deep learning/machine learning based approaches for object detection, image segmentation, identify patterns in sequential data, ranking in recommendation data. In my research work, we proposed a model which used Convolution Neural Network with Transfer Learning to classify the images and gives ton –N recommendations to the users. IEEE Senior Member (id: 97639856), Lifetime member of IAENG (International Association of Engineers), Member of (SDIWC) The Society of Digital Information and Wireless Communications. He can be contacted at email: anlakshmipathi@gmail.com.

Ravi Kumar Tirandasu Working as Assistant Professor in Computer Science and Engineering. K L Deemed to be University, India. His Research areas are Software Engineering, Machine Learning and Datamining. He got Best paper publication awards in international conference ICRTC-2021. Two patens are published. He can be contacted at email: travikumar@kluniversity.in.

Paruchuri Ravi Prakash is Assistant Professor Departemtn of Information Technology at Prasad V Potluri Siddhartha Institute of Technology, Kanuru Vijayawada.He holds a M.Tech Degree from Acharya Nagarjuna University.His reaserach areas are Data Analytics, Blockchain Technologies. He can be contacted at email: raviprakashparuchuri@pvpsiddhartha.ac.in and prppvpsit@gmail.com.