Scheduling and Resource allocation in mobile using deep learning: An Overview

Mr. Dipak R Patil[1], Dr. Manish Sharma[2]

[1]Research Scholar, Dept. of Computer Science & Engg., SGVU, Jaipur
[2]Head, Associate Professor, Dept. of CEIT, SGVU, Jaipur

Abstract- **In today's big data environment is very essential to manage a large amount of data on mobile platforms. Specific unusable programs that are already enabled on the user's mobile phone rarely regularly connect, but certain resource capacity is dedicated to hard computer storage. We explain dynamic resource management for mobile platforms in the proposed research work using a deep learning methodology. In the real world, users of mobile devices may install numerous types of applications that need ad-hoc bases various resources generate high dimensional as well as a large amount of data which should be redundant sometimes. The mobile platform digital introduces whenever new patch has generated it automatically sinks into the data server. This process repeatedly executes according to the desired time event or execute whenever a new file system has created for updates respectively. In this paper, we propose a literature review and study of existing methodologies resource allocation and memory management techniques used by existing systems. Various machine learning as well as deep learning algorithms have been developed by existing researchers and cultivate the effectiveness of systems. This paper also investigate gap analysis and the drawbacks of those existing systems the platform which is already used in real-time mobile platforms.**

Keywords—Deep learning, Cloud computing, Resource Allocation, Scheduling

I. Introduction

The key point of deep neural systems is to rough complex capacities through an arrangement of basic and predefined tasks of units (or neurons). Such a target capacity can be nearly of any kind, for example, a mapping among pictures and their class names (arrangement), processing future stock costs dependent on recorded qualities (relapse), or in any event, choosing the following ideal chess move given the present status on the board (control). The activities performed are typically characterized by a weighted mix of a particular gathering of concealed units with a non-direct actuation work, contingent upon the structure of the model. Such tasks alongside the yield units are named "layers". The neural system engineering takes after the recognition procedure in a cerebrum, where a particular arrangement of units is initiated given the present condition, impacting the yield of the neural system model.

Building a deep learning in model without any preparation can demonstrate entangled to engineers, as this requires meanings of sending practices and slope spread tasks at each layer, notwithstanding CUDA coding for GPU parallelization. With the developing notoriety of deep learning, a few committed libraries improve this procedure. The greater part of these tool compartments work with numerous programming dialects, and are worked with GPU increasing speed and programmed separation support. This takes out the need of hand-created meaning of slope spread. We condense these libraries beneath, and give a correlation among them in Table I and Table II.

The improvement of portable innovation (for example cell phones, expanded reality, and so on.) are constraining portable administrators to advance versatile system foundations. As an outcome, both the cloud and edge side of portable systems are getting progressively advanced to provide food for clients who deliver and devour enormous measures of versatile information day by day. These information can be either created by the sensors of cell phones that record singular client practices, or from the portable system framework, which reflects elements in urban conditions. Properly mining these information can profit multidisciplinary investigate fields and the business in regions such versatile system the board, social examination, open transportation, individual administrations arrangement, etc [16]. System administrators, notwithstanding, could become overpowered while overseeing and breaking down enormous measures of heterogeneous versatile information [17].

Deep learning is presumably the most impressive philosophy that can beating this weight. We start in this way by presenting attributes of versatile large information, at that point present an all encompassing audit of deep learning driven portable information examination inquire about. Yazti and Krishnaswamy[18] propose to sort portable information into two gatherings, in particular system level information and application level information. The key distinction between them is that in the previous information is normally gathered by the edge cell phones, while in the last acquired all through system foundation

Network level versatile information produced by the systems administration framework not just convey a worldwide perspective on portable system execution (for example throughput, start to finish delay, jitter, and so on.), yet in addition log singular meeting times, correspondence types, sender and collector data, through Call Detail Records (CDRs). System level information for the most part display noteworthy spatio-fleeting varieties coming about because of clients' practices [19], which can be used for organize conclusion and the executives, client portability investigation and open transportation arranging [20]. Some system level information (for example portable traffic previews) can be seen as pictures taken by 'all encompassing cameras', which give a city-scale detecting framework for urban detecting.

Then again, *App-level* information is legitimately recorded by sensors or portable applications introduced in different cell phones. This information are habitually gathered through publicly supporting plans from heterogeneous sources, for example, Worldwide Positioning Systems (GPS), versatile cameras and video recorders, and compact clinical screens. Cell phones go about as sensor center points, which are liable for information gathering and preprocessing, and in this way circulating such information to explicit areas, as required [21].

TABLE I. SUMMARY OF DEEP LEARNING LIBRARIES

Library	Platform	Written in	Interface	Recurrent nets	Convolutional nets	Mobile Supported
Torch[1]	Linux, macOS, Windows,Android,iOS	C, Lua	Lua, LuaJIT, C, utility library for C++/OpenCL	Yes	Yes	Yes
OpenNN	Cross-platform	C++	C++	No	No	No
Theano	Cross-platform	Python	Python (Keras)	Yes	Yes	No
Caffe[3]	Linux, macOS, Windows	C++	Python, MATLAB, C++	Yes	Yes	Yes
Deeplearning4j[5]	Linux, macOS, Windows, Android (Cross-platform)	C++, Java	Java, Scala, Clojure, Python (Keras), Kotlin	Yes	Yes	Yes
MXNet[6]	Linux, macOS, Windows,AWS, Android, iOS, JavaScript	Small C++ core library	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl, Clojure	Yes	Yes	Yes
Apache SINGA	Linux, macOS, Windows	C++	Python, C++, Java	Yes	Yes	Yes
Chainer[7]	Linux, macOS	Python	Python	Yes	Yes	Yes
Keras	Linux, macOS, Windows	Python	Python, R	Yes	Yes	Yes
TensorFlow[8]	Linux, macOS, Windows, Android	C++, Python, CUDA	Python (Keras), C/C++, Java, Go, JavaScript, R,Julia, Swift	Yes	Yes	Yes
BigDL	Apache Spark	Scala	Scala, Python	Yes	Yes	No
Microsoft Cognitive Toolkit (CNTK)[10]	Windows, Linux(macOS via Docker on roadmap)	C++	Python (Keras), C++, Command line,BrainScript	Yes	Yes	Yes
PyTorch[2]	Linux, macOS, Windows	Python, C, C++, CUDA	Python, C++	Yes	Yes	Yes
PlaidML[11]	Linux, macOS, Windows	Python, C++, OpenCL	Python, C++	Yes	Yes	Yes
Pylearn2	Cross Platform	Python	Python	Yes	Yes	Yes
Cuda-convnet	Cross Platform	C++, CUDA	CUDA	Yes	Yes	Yes

TABLE II. SUMMARY OF MOBILE DEEP LEARNING PLATFORM

Framework	Supported Mobile Platform	Mobile API	Is Open source	Supported Model Format	Support Training
Torch [1]	Android CPU, iOS CPU	C++	✓	customized (.dat)	✓
PyTorch [2]	Android CPU, iOS CPU	C++	✓	customized, pickle (.pkl)	✓
Caffe [3]	Android CPU, iOS CPU	C++	✓	customized, json (.caffemodel, .prototxt)	✓
Caffe2 [4]	Android CPU, iOS CPU	C++	✓	ProtoBuf (.pb)	✓
DeepLearning4J [5]	Android CPU	Java	✓	customized (.zip)	✓
MxNet [6]	Android CPU, iOS CPU	C++	✓	customized, json (.json, .params)	✓
Chainer [7]	iOS CPU,GPU	C,C++	✓	customized (.chainermodel)	✓
TensorFlow [8]	Android CPU, iOS CPU	Java, C++	✓	ProtoBuf (.pb, .pbtxt)	✓
TF Lite [9]	Android CPU, iOS CPU	Java, C++	✓	FlatBuffers (.tflite)	×
CNTK [10]	Android CPU, iOS CPU	C,C++	✓	ProtoBuf (.proto)	✓
CoreML [11]	iOS CPU, GPU	Swift, OC	×	customized, ProtoBuf (.proto, .mlmodel)	✓
FeatherCNN [12]	Android CPU, iOS CPU	C++	✓	customized (.feathermodel)	×
PaddlePaddle [13]	Android CPU, iOS CPU & GPU	C++	✓	customized (.tar)	✓
OpenCV [14]	Android CPU, iOS CPU	C++	✓	TesnorFlow, Caffe, etc	×
ncnn [15]	Android CPU, iOS CPU	C++	✓	customized (.params, .bin)	×

II. DEEP LEARNING DRIVEN NETWORK CONTROL

Right now, direct our concentration toward portable system control issues. Because of incredible capacity guess component, deep learning has made amazing achievements in improving customary support learning [22] and impersonation learning [23]. These advances can possibly take care of versatile system control issues which are intricate and recently thought to be unmanageable [24], [25]. Review that in fortification learning, a specialist persistently connects with the earth to gain proficiency with the best activity. With steady investigation and abuse, the operator figures out how to expand its normal return. Impersonation learning follows an alternate learning worldview called "learning by exhibit". This learning worldview depends on an 'instructor' who mentions to the specialist what activity ought to be executed under specific perceptions during the preparation. After adequate shows, the operator learns an arrangement that copies the conduct of the instructor and can work independent without supervision. For example, a specialist is prepared to copy human conduct (e.g., in applications, for example, game play, selfdriving vehicles, or mechanical autonomy), rather than learning by communicating with the earth, as on account of unadulterated support learning. This is on the grounds that in such applications, committing errors can have lethal outcomes [26].

Past these two methodologies, investigation based control is picking up footing in portable systems administration. In particular, this conspire utilizes ML models for organize information investigation, and in this way abuses the outcomes to help arrange control.

In contrast to fortification/impersonation learning, investigation based control doesn't straightforwardly yield activities. Rather, it extricates valuable data and conveys this to an operator, to execute the activities

Principles of three control approaches applied in mobile and wireless networks control, namely

a) Reinforcement learning ,
b) Imitation learning , and
c) Analysis based control

III. SCHEDULING

There are a few examinations that research booking with deep learning. Zhang et al. present a deep Q learning-controlled half and half unique voltage and recurrence scaling booking instrument, to decrease the vitality utilization continuously frameworks (for example Wi-Fi, IoT, video applications) [27]. In their proposition, an AE is utilized to rough the Q work and the system performs experience replay [28] to balance out the preparation procedure and quicken intermingling. Reproductions exhibit that this strategy lessens by 4.2% the vitality utilization of a customary Q learning based technique. Likewise, the work in [29] utilizes deep Q learning for planning in side of the road correspondences systems. Specifically, connections between vehicular situations, including the arrangement of activities, perceptions, and prize signs are defined as a MDP. By approximating the Q esteem work, the operator learns a planning strategy that accomplishes lower inertness and active time, and longer battery life, contrasted with customary booking strategies.

All the more as of late, Chinchali et al. present an approach inclination based scheduler to streamline the cell organize traffic stream [30]. In particular, they give the booking issue a role as a MDP and utilize RF to anticipate arrange throughput, which is hence utilized as a segment of a prize capacity. Assessments with a practical system test system show that this proposition can progressively adjust to traffic varieties, which empowers versatile systems to convey 14.7% more information traffic, while outflanking heuristic schedulers by more than 2×. Wei et al. address client planning and substance reserving all the while [31]. Specifically, they train a DRL operator, comprising of an entertainer for choosing which base station should serve certain substance, and whether to spare the substance. A pundit is additionally utilized to gauge the worth capacity and convey criticism to the on-screen character. Recreations over a bunch of base stations show that the operator can yield low transmission delay. Li et al. shed light on asset assignment in a multi-client versatile figuring situation [32]. They utilize a deep Q learning system to mutually advance the offloading choice and computational asset assignment, in order to limit the entirety cost of postponement and vitality utilization of all client hardware. Reproductions show that their proposition can lessen the all out expense of the framework, when contrasted with completely nearby, completely offloading, and guileless Q-learning draws near. Rebel et. al[33] builds up a fake specialist conveyed at the RSU, which will take in a planning approach from high-dimensional continuous inputs utilizing start to finish profound support learning. This operator infers effective portrayals of the earth, gain from past understanding, and progress towards the acknowledgment of a fruitful planning strategy all together to establish a green and safe vehicular system which accomplishes adequate degrees of QoS.

A. Dynamic Scheduling

We first give a brief analysis for a real world mobile app usage trace [34] to show the dynamics and unpredictabilities of mobile users' job requests.

Wang et al. [35] proposed VariedLen algorithm includes three scheduling policies, i.e., threshold-based policy for fronthaul links, JSQ policy for the BBU dispatching and greedy strategy for cloud server scheduling. They compare these methods to three classic scheduling methods as follows,

- ✓ *Best-effort* (B). For the fronthaul links scheduling, compare the threshold-based (T) with this method which transfers job requests as much as possible.
- ✓ *Round-robin* (R). For the BBU dispatching, compare the JSQ (J) policy with this classic scheduling method which dispatches job requests to servers in circular order.
- ✓ *First-come-first-served*, FCFS (F). For the mobile server scheduling, compare the greedy (G) strategy with this method which runs the job requests waited in the queue one by one.

The above classic scheduling algorithms have been widely used in the literature [36]–[38]. Meanwhile, when utilizing the Lyapunov technique, the state-of-the-art research often derive or compare with the above classic scheduling algorithms [39], [40], [41]. For example, Zhou et al derived the JSQ scheduling when using Lyapunov technique in SaaS cloud [39]. Nan et al used Lyapunov technique to design algorithms in Cloud of Things system and compared them with the round-robin algorithm [40], [41].

IV. Resource Allocation

Different unusable applications which are already installed in mobile platform user never access frequently but it allocates some memory space on hard device storage. In the proposed research work describe dynamic resource allocation for mobile platforms using deep learning approach [42].

The versatile figuring world is moving from 4G to 5G and one of the significant contributions of 5G is the consistent registering force and it is the significant slowed down in the present situation. The significant troubles that should be tended to are registering, nature of administrations. Speed, force and security. This examination paper points in tending to the issue of assignment the executives in the portable frameworks that is straightforwardly identified with quality. The article proposes a profound learning-based calculation that performs dynamic assignment offloading in the versatile cloudlet since cloudlet helps in the decrease of the defer that happen in the WLAN. The postponement in performing errands is one of the significant disadvantages of cloudlet that it is denied of assets when contrasted with cloud server because of which the undertakings that are to be performed are isolated and is assigned to cell phones, distinctive cloud servers and cloudlet itself. Hence, to decide the mix of gadgets required to perform various undertakings, profound learning calculations are considered. The calculation is dependable to distinguish the subtasks, the subtasks that must be figured/executed in which gadget or cloudlet or cloud server. The proposed calculation is named Deep Learning based Dynamic Task Offloading in Mobile Cloudlet (DLDTO). The calculation is actualized and contrasted and Cloudlet based Dynamic Task Offloading (CDTO). The general examination and correlation with the current CDTO for work distribution demonstrated that the exhibition of the proposed DLDTO calculation is better as far as vitality utilization and finish time [43]. Table III shows comparison of resource allocation using deep learning.

TABLE III. A SUMMARY OF WORK ON RESOURCE ALLOCATION USING DEEP LEARNING

Reference	Application	Deployment	Learning Scenario	Model
J. Liu et al [44]	CORP: Cooperative Opportunistic Resource Provisioning	Cloud based	Unsupervised	HMModel
M. Hassan et al [45]	DEARS	Cloud based	Unsupervised, Reinforcement	LSTM model
J. Koo et al [46]	Time Varying Traffic Dynamics	Cloud based	Reinforcement Learning	RNN
S. Baer et al [47]	Flexible Manufacturing Systems	Cloud based	Reinforcement Learning	CNN
D. Yi et al [48]	Green Data Centers	Cloud based	Reinforcement Learning	CNN
W. Liu[49]	Software-Defined Data-Center Networks	Cloud based	Reinforcement Learning	Deep Q-Network (DQN) and Deep Deterministic Policy Gradient (DDPG)
D. Yi et al [50]	Data Centers	Cloud based	Reinforcement Learning	deep Q-network
Sun et al. [51]	Resource management over wireless networks	Cloud based	Imitation learning	MLP
Xu et al. [52]	Resource allocation in cloud radio access networks	Cloud based	Reinforcement learning	Deep Q learning
Ferreira et al. [53]	Resource management in cognitive communications	Cloud based	Reinforcement learning	Deep SARSA
Ye and Li [54]	Resource allocation in vehicle-to-vehicle communication	Cloud based	Reinforcement learning	Deep Q learning
Challita et al. [55]	Proactive resource management for LTE	Cloud based	Reinforcement learning	Deep policy gradient

V. CONCLUSION

In mobile, deep learning is playing an important role. In this paper, we provided comparative survey of recent work between mobile and deep learning. We discussed the scheduling and resource allocation in mobile using deep learning. After an investigation of entire research finally, we conclude in any file system whenever system sync data to cloud server, it generates heavy network overload during data transmission. Such a systems also utilize heavy resources and band with other consumable parameters. Sometimes such a kind of heavy overload also generates single-point bottleneck attacks. To eliminate such problems we need to optimize search file systems or meta-data using various machine learning as well as deep learning algorithms. Implementation of different kinds of deep learning algorithms utilization of data Optimization using the data syncing will improve the effectiveness of such systems that will be interesting future work for such kind of systems.

REFERENCES

1. 2018, torch. http://torch.ch/
2. 2018, pytorch. http://pytorch.org/
3. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the ACM International Conference on Multimedia, MM*, USA, November 03- 07, 2014**.** 675–678, 2014.
4. 2018, Caffe2 https://github.com/caffe2/caffe2.
5. 2018, Deep Learning for Javahttps://deeplearning4j.org/.
6. Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang, MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. CoRR abs/1512.01274 (2015).
7. 2018,Chainer. https://chainer.org/.
8. 2018,TensorFlow. https://www.tensorflow.org/
9. 2018,TensorFlowLite.https://www.tensorflow.org/mobile/tflite/
10. 2018,CNTK https://github.com/Microsoft/CNTK
11. 2018, https://developer.apple.com/documentation/coreml
12. 2018,FeatherCNN.https://github.com/Tencent/FeatherCNN
13. 2018,https://github.com/baidu/mobile-deep-learning
14. 2018,https://opencv.org/
15. 2018, https://github.com/Tencent/ncnn
16. Xiang Cheng, Luoyang Fang, Xuemin Hong, and Liuqing Yang. Exploiting mobile big data: Sources, features, and applications. *IEEE Network*, 31(1):72–79, 2017.
17. Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning, *In Proc. 28th international conference on machine learning*, pp. 689–696, 2011.
18. Imad Alawe, Adlen Ksentini, Yassine Hadjadj-Aoul, and Philippe Bertin. Improving traffic forecasting for 5G core network scalability: A Machine Learning approach. *IEEE Network*, 32(6):42–49, 2018.
19. Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, and Antonio Pescape. Mobile encrypted traffic classification using deep learning. *In Proc. 2nd IEEE Network Traffic Measurement and Analysis Conference*, 2018.
20. Shiva Navabi, Chenwei Wang, Ozgun Y Bursalioglu, and Haralabos Papadopoulos. Predicting wireless

channel features using neural networks. arXiv preprint arXiv:1802.00107, 2018.

21. Xiang Cheng, Luoyang Fang, Xuemin Hong, and Liuqing Yang. Exploiting mobile big data: Sources, features, and applications. *IEEE Network,* 31(1):72–79, 2017.
22. Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine,* 34(6):26–38, 2017
23. Paulo Victor R Ferreira, Randy Paffenroth, Alexander M Wyglinski, Timothy M Hackett, Sven G Bilen, Richard C Reinhart, and Dale J Mortensen. Multi-objective Reinforcement Learning for Cognitive Satellite Communications using Deep Neural Network Ensembles, *IEEE Journal on Selected Areas in Communications,* 2018
24. Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, *In Proc. International Conference on Learning Representations (ICLR)*, 2016.
25. Qingjiang Shi, Meisam Razaviyayn, Zhi-Quan Luo, and Chen He. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel. *IEEE Transactions on Signal Processing,* 59(9):4331–4340, 2011.
26. Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR),* 50(2):21:1–21:35, 2017.
27. Yifei Wei, Zhiqiang Zhang, F Richard Yu, and Zhu Han. Joint user scheduling and content caching strategy for mobile edge networks using deep reinforcement learning. *In Proc. IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018.
28. Ji Wang, Jianguo Zhang, Weidong Bao, Xiaomin Zhu, Bokai Cao, and Philip S Yu. Not just privacy: Improving performance of private deep learning in mobile cloud. *In Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2407–2416, 2018.
29. Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nikos D Sidiropoulos. Learning to optimize: Training deep neural networks for wireless resource management. *In Proc. 18th IEEE International Workshop on SPAWC*, pp. 1–6, 2017.
30. Zhiyuan Xu, Yanzhi Wang, Jian Tang, Jing Wang, and Mustafa Cenk Gursoy. A deep reinforcement learning based framework for power efficient resource allocation in cloud RANs. *In Proc. IEEE ICC*, pp. 1–6. 2017.
31. Paulo Victor R Ferreira, Randy Paffenroth, Alexander M Wyglinski, Timothy M Hackett, Sven G Bilen, Richard C Reinhart, and Dale J Mortensen. Multi-objective reinforcement learning-based deep neural networks for cognitive space Comm., *In Proc. IEEE CCAA*, pp. 1–8., 2017.
32. Bomin Mao, Zubair Md Fadlullah, Fengxiao Tang, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. A Tensor Based Deep Learning Technique for Intelligent Packet Routing. In Proc. IEEE Global Comm. Conference, pp. 1–6, 2017.
33. Ribal Atallah, Chadi Assi, and Maurice Khabbaz. Deep reinforcement learning-based scheduling for roadside communication networks. *In Proc. 15th IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pp. 1–8, 2017.
34. C. Shepard, A. Rahmati, C. Tossell, L. Zhong, and P. Kortum, "Livelab: Measuring wireless networks and smart phone users in the field," *SIGMETRICS Perform. Eval. Rev*., vol. 38, no. 3, pp. 15–20, 2011.
35. Xinhou Wang, Kezhi Wang, Song Wu, Sheng Di, Hai Jin, Kun Yang, Shumao Ou, Dynamic Resource Scheduling in Mobile Edge Cloud with Cloud Radio Access Network", *IEEE Transaction on Parallel Distribution and Systems,* 2018.
36. D. Hayes, D. Ros, A. Petlund, and I. Ahmed, "A framework for less than best effort congestion control with soft deadlines," *in Proc. Of IFIP Networking,* pp. 1–9, 2017.
37. A. Sirohi, A. Pratap, and M. Aggarwal, "Improvised round robin (cpu) scheduling algorithm," *International Journal of Computer Applications*, vol. 99, no. 18, pp. 40–43, 2014.
38. R. Frijns, S. Adyanthaya, S. Stuijk, J. Voeten, M. Geilen, R. Schiffelers, and H. Corporaal, "Timing analysis of first-come first-served scheduled interval-timed directed acyclic graphs*," in Proc. of date,* pp. 288:1–288:6, 2014.
39. Z. Zhou, F. Liu, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in SaaS clouds," *in Proc .of INFOCOM*, pp. 872–880, 2013.
40. Y. Nan, W. Li, W. Bao, F. Delicato, P. Pires, Y. Dou, and A. Zomaya, "Adaptive energy-aware computation offloading for cloud of things systems," *IEEE Access*, vol. 5, pp. 947–957, 2017.
41. Y. Nan, W. Li, W. Bao, F. Delicato, P. Pires, and A. Zomaya, "A dynamic tradeoff data processing framework for delay-sensitive applications in cloud of things systems," *Journal of Parallel and Distributed Computing,* vol. 112, pp. 53–66, 2017.
42. D.R Patil, Rajesh Purohit, Dynamic Resource Allocation and Memory Management using Deep Convolutional Neural Network, *International Journal of Engineering*

and Advanced Technology, ISSN:2249–8958,Volume-9Issue-2,2019.

43. Rani, D.S., Pounambal, M. Deep learning based dynamic task offloading in mobile cloudlet environments. Evol. Intel. 2019.
44. J. Liu, H. Shen and L. Chen, "CORP: Cooperative Opportunistic Resource Provisioning for Short-Lived Jobs in Cloud Systems," *IEEE International Conference on Cluster Computing (CLUSTER)*, Taipei, 2016, pp. 90-99,2016.
45. M. Hassan, H. Chen and Y. Liu, "DEARS: A Deep Learning Based Elastic and Automatic Resource Scheduling Framework for Cloud Applications," *IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications* pp. 541-548, 2018.
46. J. Koo, V. B. Mendiratta, M. R. Rahman and A. Walid, "Deep Reinforcement Learning for Network Slicing with Heterogeneous Resource Requirements and Time Varying Traffic Dynamics," *15th International Conference on Network and Service Management (CNSM)*, Halifax, NS, Canada, pp. 1-5. 2019.
47. S. Baer, J. Bakakeu, R. Meyes and T. Meisen, "Multi-Agent Reinforcement Learning for Job Shop Scheduling in Flexible Manufacturing Systems," *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, Laguna Hills, CA, USA, pp. 22-25, 2019.
48. D. Yi, X. Zhou, Y. Wen and R. Tan, "Toward Efficient Compute-Intensive Job Allocation for Green Data Centers: A Deep Reinforcement Learning Approach," *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, pp. 634-644, 2019.
49. W. Liu, "Intelligent Routing based on Deep Reinforcement Learning in Software-Defined Data-Center Networks," *2019 IEEE Symposium on Computers and Communications (ISCC)*, Barcelona, Spain, pp. 1-6, 2019.
50. D. Yi, X. Zhou, Y. Wen and R. Tan, "Efficient Compute-Intensive Job Allocation in Data Centers via Deep Reinforcement Learning," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 6, pp. 1474-1485, 1 June 2020.
51. Hao Ye and Geoffrey Ye Li. Deep reinforcement learning for resource allocation in V2V communications. *In Proc. IEEE International Conference on Communications*, pp 1–6, 2018.
52. Hao Ye and Geoffrey Ye Li. Deep reinforcement learning for resource allocation in V2V communications. *In Proc. IEEE International Conference on Communications (ICC)*, pages 1–6, 2018.
53. Oshri Naparstek and Kobi Cohen. Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks. *In Proc. IEEE Global Communications Conference*, pp. 1–7, 2017.
54. Timothy J O'Shea and T Charles Clancy. Deep reinforcement learning radio control and signal detection with KeRLym, a gym RL agent. arXiv preprint arXiv:1605.09221,
55. Michael Andri Wijaya, Kazuhiko Fukawa, and Hiroshi Suzuki. Intercell-interference cancellation and neural network transmit power optimization for MIMO channels. *In Proc. IEEE 82nd Vehicular Technology Conference*, pp. 1–5, 2015.