A Survey on Resource Allocation Algorithms and Models in Cloud Computing

Noura AlDossary, Sarah AlQahtani, Haya AlUbaidan and Atta-ur-Rahman*
Department of Computer Science, College of Computer Science and Information Technology,
Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
Email: {2220500187, 2220500185, 2220500183,

Abstract
The cloud is an open environment where users can use the resources and services that are available to them. A cloud server or cloud service provider provides resources and services to the end user. User expectations must be met when offering these services. Since resources need to be allocated to users in a timely manner, that is where the issue resides. This research examines three main resource allocation algorithms in detail, including ant colony optimization, deadline guaranteed scheduling, and scheduling based on genetic algorithms. In this paper, a proposed cloud computing resource allocation model that can be implemented to manage resources in a cloud computing environment based on the Fuzzy Analytic Hierarchy Process (FAHP) algorithm was discussed.
Keywords: *Cloud Computing, Resource Allocation, Scheduling, Virtual Machine.*

I. INTRODUCTION

In the past decade, the computational world has rapidly developed, due to continuous expansion in demand for advanced computational devices. This development resulted in the rise of new computational models such as, cloud computing. Cloud computing is based on Service-Oriented Architecture (SOA) that adopts the concept of both virtualization and distributed computing [1]. In cloud computing SOA, a pool of computing resources is accessed and shared over the internet and can be modified according to users' demand [2].

As a result of cloud computing's growing popularity, cloud computing service providers such as Google, IBM and Amazon have started to establish an increasing number of distributed data centers for fulfilling the ever-growing customer resource demands. From cloud computing perspective, resource management is the process of allocating computing resources (e.g., CPU, Storage and Networking) with an aim of meeting performance objectives for applications, service providers and users [3]. Also, resource management in cloud computing is considered a difficult task, due to the diversity of resource types and the inconsistency and unpredictability of the load [4].In addition, utilization and consumption of available resources is the focus of the cloud. Cloud users can request access to the cloud server for the available resources to run some application. These resources are not only shared by a single user at a specific time but can be shared by many clients in the meantime. Hence, a cloud computing system must consider wider range of optimization scheduling algorithms to solve this task at hand [20][21]. The aim of this paper is to provide a survey of recent research in resource scheduling algorithms and policies in the cloud. Figure 1 shows the resource allocation in the cloud environment. The user submits the request for resource allocation to the cloud server. Based on the availability the resource is allocated to the user.

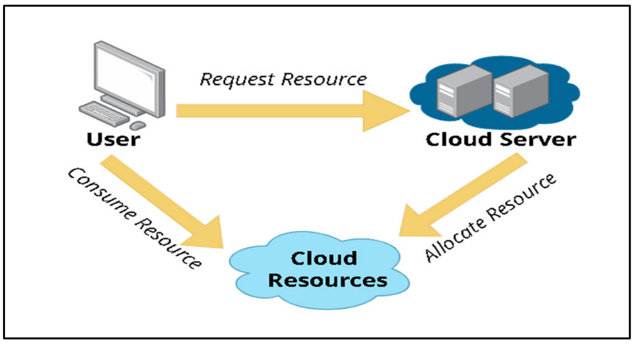

Figure 1:Resource allocation in cloud.

II. LITRATURE REVIEW

There has been extensive research on resource scheduling algorithms in cloud computing with IoT [22], edge and fog computing [23], with researchers addressing the functions and parameters of several algorithms. This section discusses related work on three main resource scheduling algorithms as shown in Figure 2.

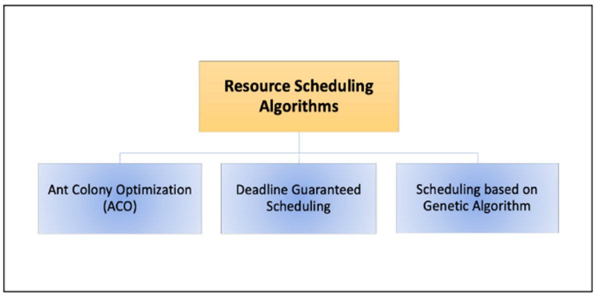

Figure 2:Resource Scheduling Algorithms

Manuscript received March 5, 2022
Manuscript revised March 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.3.102

A. Ant-colony Optimization

Ant colony optimization (ACO) was used by the authors in [5] to optimize the scheduling of cloud resources based on actual QoS (quality of service) parameters in the environment. Prioritizing tasks is the first step in an algorithmic process that identifies the most important tasks for each user. Classes represent the various QoS requirements imposed by different user tasks. Second, use ant colony optimization to allocate and schedule the resources according to Quality of Service (QoS). To complete the process, once the QoS criteria have been met and the shortest path has been taken, the task can be bound to the resource and run. In their experiment, they used several parameters, including executing time and network bandwidth. As a result, the colony optimization method takes less time than using a standard approach.

Authors in [6] presented a technique based on the ant colony optimization algorithm to improve the allocation of virtual machines to their physical counterparts in a more efficient manner. In fact, combining the ant colony optimization algorithm with the local search method can significantly improve its performance. The local search algorithm is used to solve computationally difficult problems. Modifications to the solution space at a local level are investigated until an optimal solution is found or the timer runs out. The proposed algorithm works as follows: the existing solution's bins are destroyed, and their contents are transferred to other bins. There are fewer bins required if the existing solution is found to be more optimum. During the experiment's implementation, the number of virtual machines and the host's utilization rate were used as factors.

Authors in [7] developed an enhanced differential evolution ant colony algorithm that combines the ant colony method that preserves its advantages with the differential evolution algorithm. Moreover, changing the size of the mutation operator F in the differential evolution algorithm prevents the algorithm from falling into a local search state and premature convergence. Local search optimality and better global evolution of the differential evolution method are also properties of the new approach. During their experiment, different task numbers were used to test the proposed algorithm. Unbalanced load and extended processing times in cloud computing resource scheduling can be effectively addressed by combining these two algorithms.

Authors in [8] provide an enhanced ant colony optimization algorithm for cloud computing task scheduling optimization. The ant colony algorithm's pheromone update formula now includes reward and punishment coefficients. As soon as an ant discovers the shortest path, it promptly "reward" the path it is presently on according to the reward and punishment coefficient. The pheromone gains the corresponding value in this way. Current paths that are longer than the previous ones are "punished" according to a coefficient of reward and punishment. During their experiment, they set the following algorithm parameters: number of tasks, length of tasks, number of virtual machines, and execution speed. As a result, the proposed algorithm has the fastest convergence speed. It also has the fastest completion time, the most balanced load, and the most efficient use of virtual machine resources.

Load balancing based on the ant colony algorithm for cloud resource optimization scheduling was proposed by the authors in [9]. Server load factor, a new concept in the traditional ant colony algorithm, is added to allow for dynamic changes in the cluster's server load to achieve an optimally balanced schedule. Their technique considers the utilization of CPU, memory, and network bandwidth as three key metrics. Moreover, the enhanced ant colony algorithm is capable of global searching for the optimal solution based on the load factor. In terms of job completion time, CPU utilization, and load balancing degree, their suggested technique outperforms other algorithms.

B. Deadline guaranted scheduling

Authors in [10] presented a proposed scheduling algorithm to improve resource utilization and deadline guarantee. They enhanced the conservative backfilling algorithms by using largest weight first (LWF) algorithm and the earliest deadline first (EDF) algorithm. The suggested algorithms initially score any jobs that reached the Datacenter (DC). Also, all jobs will be stored ascending to deal with high-priority jobs first. The suggested algorithm has chosen the largest backfill jobs based on the guaranteed deadline. This experiment result shows that the performance has enhanced significantly in both deadline guarantee and resource utilization.

Authors in [11] presented a new form of SLAs, helps each user (tenant) reserve a percentage to provide within the required deadline. The current system considers an unlimited number of tenants and serving provides a capability in the two different side serving and storage; however, it delivers the same storage service. That means the request reaches the front-end server of the cloud as the first stage after that is the base of load balancing policy the request redirected to the server. All hosts consider as replicas of the requested data partition. A request's service latency is the response time that takes the longest across all destination servers. To improve that, they were proposed a new SLA for cloud storage service. It is for any tenant specified percent of the number of requests and deadline that will be agreed in their SLA with the cloud provider.

To explain it more, they take all similar SLAs for all requesters from the tenant and then divide them into multiple SLAs for different demands from tenants. If a tenant makes various requests with different SLAs, the tenant divides SLAs into several sub-tenants. They assume that data request answers are independent, meaning that the servers handle data requests separately. The proposed parallel deadline guaranteed scheme for cloud storage systems uses a variety of approaches to ensure SLAs when addressing many objectives such as low traffic load, high resource utilization, and quick scheme execution. As the result of the proposed SLA enhanced resource utilization, and prioritized data reallocation algorithm.

The author in [12]suggested an algorithm named just in time (JIT-C), which makes suitable scheduling decisions just before workflow tasks are available for execution. The presented algorithm utilizes a monitor control loop to make timely scheduling/provisioning decisions in the case of performance variance. The status of ongoing tasks is continually monitored within each loop, and resource provisioning/scheduling decisions are made based on the most up-to-date information. The proposed algorithm comparison with baseline methods is conducted to satisfy workflow deadline constraints at lower costs. the proposed scheduling algorithm addresses three primary concerns with cloud platforms: VM performance variability, resource acquisition delays, and the diverse nature of cloud resources.

The authors in [13] suggested an algorithm focus on the best way to utilize resources effectively within their deadlines. This proposed algorithm prioritizes all jobs based on deadlines and time executions. First, the jobs that the users have submitted will be stored in the global queue. Then the classified will separate all jobs in the global queue into three dynamic priority queues: high, medium, and low. The high priority may face issues with the missed deadline because the high priority jobs must be executed first to reach their deadline.

Moreover, all high-priority jobs should assign to high-priority VMs to be capable of completing their deadlines, all that will impact the performance positivity and reduce the chances of losing deadline. All VMs need to arrange in ascending order based on their weight in this stage. However, the classified VMs will be the same way used to classify the jobs. The VMs priority is as follows: low, medium, and high priority level queue of VMs. This experiment has been run on top of cloud to ensure the quality of the suggested scheduling algorithm. The proposed scheduling algorithm has been compared with the First Come First Serve (FCFS) algorithms and Earliest Deadline (EDF). As a result, of their experiment, the proposed algorithm meets both the needs of cloud customers and cloud service providers.

C. Scheduling based on Genetic Algorithm

Genetic Algorithms (GA) is a powerful heuristic scheme optimization algorithm inspired by Darwin's theory of natural evolution. The term "Survival of the fittest" in his theory corresponds to method used for scheduling tasks. Tasks are assigned to resources based on the value of fitness function for each parameter of the task scheduling procedure [14].Genetic algorithms are commonly used to solve both search and optimization problems. There are several genetic operations in GA which are crossover, mutation, selection, and elitist selection. These activities contain a set of solutions that can be evolved by reproduction, mutation, and other functions. Hence, most fit possibilities will be chosen, and worst possibilities will be disregarded after several stages of evolution. The genetic scheduler in [15] gave better solutions to Hadoop framework, which is an implementation of map-reduce [16]. This resulted in an overall enhancement of the global execution time.

Authors in [15]introduced genetic based task scheduling algorithm for cloud computing. The main idea of their proposed method is that after each selection in the population, there exists a solution that might fulfill good fitness function, but it is not chosen to crossover process. The method they proposed includes this solution and doesn't remove it from the population. The solution is added up to the population when the next iteration starts. This is considered a good step as some iterations can produce the best solution. Also, the suggested strategy aims to minimize cost and completion time and maximize utilization of resources. Their experimental results proved that the utilization rate of Tournament Selection Genetic Algorithm (TS-GA) is higher rather than the default GA and the round robin (RR) algorithm.

The authors in [17]proposed a scheduling approach on load balancing of Virtual Machine (VM) resources based on GA. Their method depends on the historical data and current state to calculate in advance the influence it will have on the overall system. This will occur when the current VM service resources that require deployment are arranged to every physical node, and then it selects the solution which will have the minimum effect on the system after arrangement. In this manner, the strategy achieves the best load-balancing, and diminishes or prevents dynamic migration. Based on the results, this method can improve load balancing and resource utilization.

The authors in [4]presented a collaborative optimization scheduling approach of cloud resources based on improved GA (OSIG). They considered a wider range of cloud service resource types and collaborative optimization scheduling problems to solve the available tasks. The experimental results verified the effectiveness and reliability of the OSIG algorithm with regards to computing resource scheduling. Also, it was shown that the convergence rate of the OSIG algorithm is faster because of the enhancement of mutation and crossover operations.

OSIG algorithm can effectively reduce the task execution time, thereby enhancing the overall system's quality of service (QoS) and improving the user experience. Due to their universal optimization nature, the evolutionary computing algorithms are famous for solution to many problems in engineering, communications, behavioral studies, healthcare and many more [24][25][26][27][28][29][30]. Moreover, they are proven best for resource allocation in the continuous spaces optimally. Table 1 shows an overview of a comparison between on resource scheduling algorithms in cloud computing in term of proposed approach, algorithms' factors, and function of each algorithm.

III. CLOUD COMPUTING RESOURCE ALOCATION MODEL

Cloud computing technology is a game-changer, the main action from the user's side to submit their jobs for execution. Moreover, the cloud service provider handles scheduled jobs and manages resources professionally[18]. Resource management is well known problem in many other areas as well like [31-45].

The proposed model as shown in Figure 3 is to manage incoming jobs that come into cloud. Moreover, the proposed model uses the fuzzy analytic hierarchy process (FAHP) to rank the jobs based on run time and length. If jobs have been ranked individually, they will join the jobs queue. We can consider the first stage of this proposed completed. As the second stage, (BATS) and BAR optimized allocation methodology is implemented on cloud data center resources such as CPU, Memory, and bandwidth.

Table 1: Comparison between resource scheduling algorithms.

Algorithm	Ref.	Proposed approach	Factors	Function
Ant colony optimization	[5]	ACO virtual machine oriented	Executing time and network bandwidth.	Prioritizing tasks for each user to allocate the resources according to QoS. Once the QoS criteria have been met, and the shortest path has been taken, the task can be bound to the resource and run.
	[6]	Local Search based Ant Colony Optimization	Number of virtual machines and host's utilization rate.	Modifications to the solution space at a local level are investigated until an optimal solution is found.
	[7]	Differential evolution ant colony algorithm	Task numbers	Combines ant colony method that preserves its advantages with the differential evolution algorithm.
	[8]	Ant colony optimization algorithm for cloud computing task scheduling optimization	Number of tasks, length of tasks, number of virtual machines, and execution speed.	Reward the shortest path, and punishment longer path coefficients.
	[9]	Load balancing based on Ant colony algorithm	Server load factor, utilization of CPU, memory, and network bandwidth.	Global searching for the optimal solution based on the load factor.
Deadline guaranteed scheduling	[10]	Proposed scheduling algorithm to improve resource utilization and deadline guarantee.	Order of jobs	Chosen the largest backfill jobs based on the guaranteed deadline.
	[11]	Presented a new form of SLAs, helps each user (tenant) reserve a percentage to provide within the required deadline.	Load balancer	Uses a variety of approaches to ensure SLAs addressing many objectives.
	[12]	Suggested an algorithm named just in time JIT-C	Monitor control loop to make timely scheduling/provisioning decisions.	The proposed algorithm comparison with baseline methods is conducted to satisfy workflow deadline constraints at lower costs.
	[13]	Proposed algorithm prioritizes all jobs based on deadlines and time executions.	Deadline high priority.	The proposed scheduling algorithm has been compared with the First Come First Serve (FCFS) algorithms and Earliest Deadline (EDF).
Scheduling based on Genetic algorithm	[15]	Scheduling using genetic algorithms.	Minimize global execution time.	Improve Hadoop framework functionality using a scheduler based on GA.
	[15]	Scheduling on load-balancing of VM resources using genetic algorithms.	Reduce or prevent dynamic migration.	Improve resource utilization and load-balancing using proposed method.
	[17]	Scheduling using Tournament Selection Genetic Algorithm (TS-GA)	Minimize cost and completion time.	Improve genetic algorithm for task scheduling to maximize resource utilization.
	[4]	Scheduling based on improved Genetic Algorithm (OSIG)	Mutation and crossover operations.	Improve overall system QoS and user experience using (OSIG) algorithm.

After that, the proposed model uses preemption methodology to see the load of the virtual machine (VM). Also, the proposed model uses the VMs table to check the status of VM. If the VM in running or executing status, the idle VM only can be utilized. In this case, the proposed model helps cloud providers manage resources effectively.

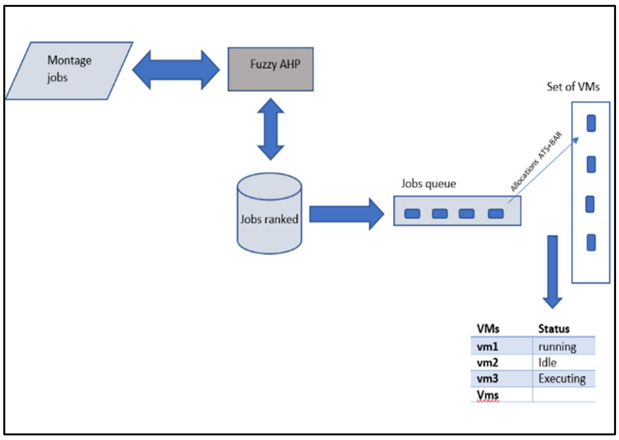

Figure 3: Proposed model flowchart.

The proposed model uses Montage scientific workflows as input. Figure 4 shows a visualization of Montage scientific workflows. Cloud computing technology is a game-changer, the main action from the user's side to submit their jobs for execution. Moreover, the cloud service provider handles scheduled jobs and manages resources professionally. According to our information, the user's interaction ended once they submitted their jobs. Then the service provider addressed scheduling tasks and resource management [18]. Montage created by the NASA/IPAC Infrared Science Archive. It is a toolkit (open source) that utilizes input images in the Flexible Image Transport system (FIST) format to produce custom mosaics of the sky. Also, it is used to generate the geometry of the output image. Montage workflow. The circles represent the various computational workflow tasks. The arrows represent the data dependencies between the tasks. Different colors represent different job types [19].

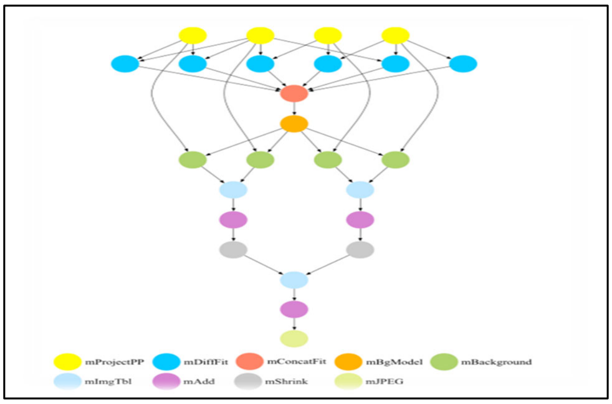

Figure 4: Visualization of Montage scientific workflows[18].

IV. Conclusion

The distribution of cloud computing resources plays an important role, as cloud computing relies heavily on the availability of resources. This paper discusses and compares several resource scheduling algorithms in a cloud computing environment in terms of function and effect factors. Ant colony optimization, deadline guaranteed scheduling, and scheduling based on genetic algorithms were investigated and compared. More importantly, bandwidth is the most important influence factor in the resource scheduling algorithms that were compared in this paper, and it is critical in the cloud environment since it allows data to be sent quickly and efficiently. This research proposed a cloud computing resource allocation model that can be implemented to manage resources in a cloud computing environment based on the Fuzzy Analytic Hierarchy Process (FAHP) algorithm. This model could enhance the process of allocating different resources in a cloud environment. In the future, a comprehensive implementation of the suggested resource allocation graph model can be developed and evaluated.

References

[1] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," *NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC*, pp. 44–51, 2009, doi: 10.1109/NCM.2009.218.

[2] P. Mell and T. Grance, "The NIST-National Institute of Standars and Technology- Definition of Cloud Computing," *NIST Special Publication 800-145*, p. 7, 2011.

[3] B. Jennings and R. Stadler, "Resource Management in Clouds : Survey and Research Challenges," pp. 567–619, 2015, doi: 10.1007/s10922-014-9307-7.

[4] R. Liu, S. Liu, and N. Wang, "Optimization Scheduling of Cloud Service Resources Based on Beetle Antennae Search Algorithm," *Proceedings of the 2020 International Conference on Computer, Information and Telecommunication Systems, CITS 2020*, vol. 8, 2020, doi: 10.1109/CITS49457.2020.9232458.

[5] Y. Zhu and H. Liang, "Research for the virtual machine-oriented cloud resource scheduling algorithm," in Proceedings of 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2013, 2013, vol. 1, pp. 133–136. doi: 10.1109/ICIII.2013.6702893.

[6] N. K. Gondhi and A. Sharma, "Local Search Based Ant Colony Optimization for Scheduling in Cloud Computing," in *Proceedings - 2015 2nd IEEE International Conference on Advances in Computing and Communication Engineering, ICACCE 2015*, Oct. 2015, pp. 432–436.

[7] X. Xie, K. Xu, and X. Wang, "Cloud computing resource scheduling based on improved differential evolution ant colony algorithm," in *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, Apr. 2019, pp. 171–177. doi: 10.1145/3335656.3335706.

[8] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[9] N. Tang and H. Zhang, "A Strategy of Cloud Resource Load Balancing Enhancement Based on Ant Colony Optimization," in *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, 2020, pp. 172–178.

[10] S. Shin, Y. Kim, and S. Lee, "Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing," in *2015 12th Annual IEEE Consumer Communications and Networking Conference, CCNC 2015*, Jul. 2015, pp. 814–819. doi: 10.1109/CCNC.2015.7158082.

[11] G. Liu, H. Shen, and H. Wang, "Deadline Guaranteed Service for Multi-Tenant Cloud Storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2851–2865, Oct. 2016.

[12] J. Sahni and P. Vidyarthi, "A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 2–18, Jan. 2018, doi: 10.1109/TCC.2015.2451649.

[13] S. K. Grewal and N. Mangla, "Deadline based energy efficient scheduling algorithm in cloud computing environment," in *Proceedings - 2021 4th International Conference on Computational Intelligence and Communication Technologies, CCICT 2021*, Jul. 2021, pp. 383–388.

[14] T. Harada and E. Alba, "Parallel Genetic Algorithms: A Useful Survey," *ACM Computing Surveys*, vol. 53, no. 4, 2020, doi: 10.1145/3400031.

[15] F. A. Emara, A. A. A. Gad-Elrab, A. Sobhi, and K. R. Raslan, "Genetic-Based Multi-objective Task Scheduling Algorithm in Cloud Computing Environment," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 571–582, 2016.

[16] "Apache Hadoop." https://hadoop.apache.org/ (accessed Dec. 28, 2021).

[17] J. Gu, J. Hu, T. Zhao, and G. Sun, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment," *Journal of Computers*, vol. 7, no. 1, pp. 42–52, 2012, doi: 10.4304/jcp.7.1.42-52.

[18] M. B. Gawali and S. K. Shinde, "Task scheduling and resource allocation in cloud computing using a heuristic approach," *Journal of Cloud Computing*, vol. 7, no. 1, Dec. 2018, doi: 10.1186/s13677-018-0105-8.

[19] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and Profiling Scientific Workflows".

[20] A. Rahman, S. Dash, M. Ahmad, T. Iqbal, "Mobile Cloud Computing: A Green Perspective," Intelligent Systems, Lecture Notes in Networks and Systems book series (LNNS, volume 185), pp. 523-533, 2021.

[21] A. Rahman, Kiran Sultan, Sujata Das, Muhammad Aftab Khan, "Management of Resource Usage in Mobile Cloud Computing", International Journal of Pure and Applied Mathematics, Vol. 119 No.16, pp. 255-261, 2018.

[22] S. Dash, S. Biswas, D. Banerjee, A. Rahman, "Edge and Fog Computing in Healthcare – A Review", Scalable Computing 20(2):191-206, 2019.

[23] F. Alhaidari, A. Rahman, & R. Zagrouba, "Cloud of Things: architecture, applications and challenges." J Ambient Intell Human Comput (2020). https://doi.org/10.1007/s12652-020-02448-3.

[24] A. Rahman, "GRBF-NN based ambient aware realtime adaptive communication in DVB-S2." J Ambient Intell Human Comput (2020). https://doi.org/10.1007/s12652-020-02174-w.

[25] A. Rahman, S. Dash, & A.K. Luhach, "Dynamic MODCOD and power allocation in DVB-S2: a hybrid intelligent approach." Telecommun Syst, vol. 76, pp. 49–61, 2021. https://doi.org/10.1007/s11235-020-00700-x

[26] M.A. Khan, S. Abbas, A. Atta, A. Ditta, H. Alquhayz, M.F. Khan, A. Rahman, R.A. Naqvi, "Intelligent cloud based heart disease prediction system empowered with supervised machine learning," Computers, Materials & Continua, vol. 65, no.1, pp. 139–151, 2020.

[27] M. Ahmad, M.A. Qadir, A. Rahman et al., "Enhanced query processing over semantic cache for cloud based relational databases." J Ambient Intell Human Comput (2020). https://doi.org/10.1007/s12652-020-01943-x.

[28] A. Rahman, S. Dash, A.K. Luhach, N. Chilamkurti, S. Baek, Y. Nam, "A Neuro-Fuzzy Approach for User Behavior Classification and Prediction", Journal of Cloud Computing, 8(17), 2019.

[29] A. Rahman, "Memetic Computing based Numerical Solution to Troesch Problem", J. Intell. Fuzzy Syst, 37(1):1545-1554, 2019.

[30] A. Rahman, "Optimum Information Embedding in Digital Watermarking", J. Intell. Fuzzy Syst, 37(1):553-564, 2019.

[31] A. Rahman, K. Sultan, I. Naseer, R. Majeed, D. Musleh et.al., "Supervised Machine Learning-based Prediction of COVID-19," Computers, Materials & Continua, vol. 69, no.1, pp. 21-34, 2021. DOI: 10.32604/cmc.2021.013453.

[32] A. Dilawari, M.U.G. Khan, Y.D. Al-Otaibi, Z. Rehman, A. Rahman, Y. Nam, "Natural Language Description of Videos for Smart Surveillance," Appl. Sci., vol. 11, no. 9: 3730, 2021.

[33] S. M. Alotaibi, A. Rahman, M. I. Basheer and M. A. Khan, "Ensemble machine learning based identification of pediatric epilepsy," Computers, Materials & Continua, vol. 68, no.1, pp. 149–165, 2021.

[34] R. Zagrouba, A. AlAbdullatif, K. AlAjaji, N. Al-Serhani, F. Alhaidari, A. Almuhaideb, A. Rahman, "Authenblue: a new authentication protocol for the industrial internet of things," Computers, Materials & Continua, vol. 67, no.1, pp. 1103–1119, 2021.

[35] F. Alhaidari, S. H. Almotiri, M. A. Ghamdi, M. A. Khan, A. Rehman, S. Abbas, K.M. Khan, A. Rahman, "Intelligent software-defined network for cognitive routing optimization using deep extreme learning machine approach," Computers, Materials & Continua, vol. 67, no.1, pp. 1269–1285, 2021.

[36] N.A. Saqib, A.A. Salam, A. Rahman, S. Dash, "Reviewing risks and vulnerabilities in web 2.0 for matching security considerations in web 3.0," Journal of Discrete Mathematical Sciences and Cryptography, vol. 24 (3), pp. 1-17, 2021.

[37] A. Rahman, M.I.B. Ahmed, "Virtual Clinic: A CDSS Assisted Telemedicine Framework", Chapter 15, Telemedicine Technologies, 1st Edition. Elsevier, 2019.

[38] A. Rahman, K. Sultan, N. Aldhafferi, A. Alqahtani, M. Mahmud, "Reversible and Fragile Watermarking for Medical Images", Computational and Mathematical Methods in Medicine, June 2018.

[39] A. Rahman, K. Sultan, Naheir Aldhafferi, A. Alqahtani, D. Abdullah, M. Mahmud, "Robust and Fragile Watermarking for Medical Images: A Joint Venture of Coding and Chaos Theories", Journal of Healthcare Engineering, June 2018.

[40] M. Zaheer, I.M. Qureshi, A. Rahman, J. Alhiyafi, M.Z. Muzaffar, "Improved and Secure Differential LSB Embedding Steganography", Journal of Information Assurance & Security, vol. 11, pp. 170-178, 2018.

[41] M. Azam, A. Rahman, K. Sultan, S. Dash, S. Khan, M.A.A. Khan, "Automated Testcase Generation and Prioritization Using GA and FRBS" Advanced Informatics for Computing Research, Second International Conference, ICAICR 2018, 955, pp. 573-584, Shimla, India, July 14–15, 2018. Springer.

[42] A. Rahman, "Applications of Hybrid Intelligent Systems in Adaptive Communications", Modeling, Analysis and Applications of Nature-Inspired Metaheuristic Algorithms, Edition: 1st, Chapter: 10, pp. 183-217, Publisher: IGI Global, 2017.

[43] A. Rahman, Qureshi I.M., Malik A.N., Naseem M.T., "QoS and Rate Enhancement in DVB-S2 using Fuzzy Rule Base System", Journal of Intelligent & Fuzzy Systems (JIFS), vol. 30 (1), pp. 801-810, 2016.

[44] A. Rahman, I.M. Qureshi, A.N. Malik, M.T. Naseem, "Dynamic Resource allocation for OFDM Systems using DE and Fuzzy Rule Base System", Journal of Intelligent & Fuzzy Systems (JIFS), vol. 26 (4), pp. 2035-2046, 2014.

[45] A. Rahman, I.M. Qureshi, A.N. Malik, M.T. Naseem, "A Real Time Adaptive Resource Allocation Scheme for OFDM Systems using GRBF-Neural Networks and Fuzzy Rule Base System", International Arab Journal of Information Technology, vol. 11, no. 6, pp. 593-601, 2014.

Dr. Atta-ur-Rahman is currently working at department of CS, College of CS & IT (CCSIT), Imam Abdulrahman Bin Faisal University (IAU), Dammam, KSA, as Associate Professor. Dr. Atta has completed his BS degree in CS, MS and PhD degree in EE in years 2004, 2008 and 2012, respectively. He authored more than 100 publications, cumulative impact score is above 100. His research interests include; Adaptive Digital & Wireless Communication, DSP, Information & Coding Theory, AI and Applied Soft computing.

Noura AlDossary, Sarah AlQahtani, Haya AlUbaidan
are currently persuing their Masters degree at department of CS, CCSIT, Imam Abdulrahman Bin Faisal University, Dammam. Their research interests include data mining, Artificial Intelligence and Machine Learning and Big data.