Genetic Algorithm Inspired Task Scheduling Optimization in Cloud Environment

Raveel Abdullah

Abstract— With the advancements in various fields of technology, complex and data-oriented problems require supercomputers for proper computation. Massive data that is gathered from various fields of sciences and engineering and it is increasing exponentially with every passing day. There is a dire need of economical solution for efficient processing of data. This is where cloud computing comes into picuture. Cloud computing plays a major role in providing services to individual people and companies with added benefits such as elastic computation, pay on the go, scalable and high-performance computation solutions. The performance factor in cloud environments is highly reliant on task scheduling. Load balancing is simply the distribution of incoming load of users' request onto the available computing machines. This paper aims to cover a brief overview of Cloud computing and Genetic algorithm and implementation of genetic algorithm for task scheduling pupose. The aim of this paper is to perform a comparative analysis of various task scheduling algorithms that are already in practice and are used in cloud computing. The paper also covers improved scheduling techniques that are inspired from Genetic Algorithm. A systematic comparative analysis of different task scheduling algorithms is also included. Lastly some concluding remarks and future work inspiration is described.

Index Terms— Cloud computing, Task scheduling, Genetic Algorithm

1 Introduction

TODAY, the fields of science and technology have made so many advancements and we are facing much complex and complicated challenges that require tremendous amount of computing resources. Another important factor for consideration is cost of computational resources. Not many organizations can afford to spend huge amounts for the purchase of supercomputers.

Cloud computing is an economical solution for high computational demands and today cloud computing plays a major role in providing services to individual people and companies with added benefits such as elastic computation machines, pay on the go, scalable and high-performance computations, etc.

In cloud computing, load balancing is extremely important for efficient utilization of resources. Load balancing is simply distribution of incoming load of users' request onto the available computing machines[1].

There are two main techniques through which load balancing can be achieved in cloud environment, these are as follows:

- Task scheduling: A common technique for load balancing is using task scheduling. For this purpose, multiple scheduling algorithms have been presented to ensure effective allocation of tasks on available resources.
- Virtual machine migration: Load balancing can also be performed by switching tasks from one over-burden virtual machine to some other idle virtual machine.

Now in order to meet these challenges, some concepts have been around like Cloud computing and Grid computing. The term cloud computing is not entirely some new concept rather it is some improved variant of relatively old concept of Grid computing.

The idea of Grid computing is that there are many different computer machines located in different regions, that follow homogenous or heterogenous system attributes and are connected with each other to solve a particular problem at hand[2]. The computers in grid environment can be located locally within a building or they can be placed in different locations geographically given that all are connected with each other via high-speed internet. Basic idea in grid computing is to use many general computers and combine their computational power in order to solve large complex problems.

The rest of the paper is organized as follows: Section 2 discusses some basic concepts of cloud computing and the need of task scheduling. In section 3, some basic concepts of Genetic Algorithm are explained along with the Literature survey of various papers published in similar domain. Later section deals with a comparative study of various task scheduling algorithms that are used in Cloud environment. Lastly, section 4 deals with concluding remarks and future work inspirations.

2 Task Scheduling in Cloud Environment

Task scheduling in cloud environments is major hindrance in optimum performance of compute engines. Cloud computing is responsible for catering a large number of user requests with different processing needs therefore it is quite difficult to cope up with the ever-changing needs of customers[3].

Therefore, some of the algorithms are presented here that are commonly used for resolving this issue of task scheduling.

Task scheduling is a major issue in both cloud computing and grid computing and the algorithms presented for this purpose

- *The author is with Riphah International College Sargodha 40100, Pakistan. E-mail: raveelabdullah@gmail.com*

work exactly the same. Therefore, in the later section, we will discuss various scheduling techniques and solutions with respect to both computing paradigms.

A grid consists of several computers that are distributed in different regions geographically and each system running a different operating system, but all these systems are connected with each other via internet and thus forming a grid. There is a need to efficiently utilize each computing resource among grid in order to maintain a system workflow.
It is very important that each resource in cloud is accessible along with its attributes (computational power).

General steps involved for problem solving in cloud environment are as follow[4]:

1. Identifying resources that are available.
2. Gathering resource details and selecting a suitable task scheduler.
3. Successful completion of the job.

Due to presence of different software and hardware in cloud environments, there arise a number of problems. Among many problems is scheduling of tasks. And the fact that task scheduling in cloud is NP-complete problem, yet we try to select the most appropriate technique in each scenario[5]. The basic purpose of this scheduling of task in grid environment is just to get efficient and effective throughput.
In simple terms we schedule task in grid environment to gain maximum throughput by completing the tasks as early as possible.

The problem in cloud environment is that we cannot simply just utilize the simple scheduling algorithms like Shortest Job First (SJF) or First Come First Serve (FCFS)[6].
One of the reasons is that every problem is different and every time a new problem arises, the availability of appropriate resources also varies each time. As mentioned earlier, task scheduling in NP-complete problem therefore the most appropriate way to implement any algorithm is using some heuristic information and technique in order to cater the issues arising in multiple scenarios[7].

Therefore, each time a new complex problem arrives, the first step is to discover all resources that are available at that time. Then in the light of available resources, most appropriate scheduling algorithm is selected for smooth execution of the problem. Various researches have been presented for the task scheduling purpose. Most of the algorithms use heuristic approaches for better adaptation to varying resources and other factors that continuously evolve in grid environments. Some of the most commonly used algorithms are Min-Min, Max-Min, Genetic algorithms and Particle swarm optimization.

Load balancing simply ensures that the given complex problem is distributed appropriately (according to resource) across all available resources[1]. In this way we can ensure efficient working and maximum system throughput.
Several techniques have been presented over the years to tackle the issue of load balancing and to avoid any possible interruption in job execution due to evolving attributes.

Now we will shed some light on various scheduling algorithms[1]. When we talk about load balancing algorithms, there are general two types:

- Static load balancing
- Dynamic load balancing

Now in static load balancing, some prior information of computing machines or VMs is required instead of current condition. This prior information of system indicating status is used to allocate tasks on to appropriate machines. In this method, resource allocation is confirmed initially at compile time.

On the contrary, dynamic load balancing does not require prior information or status of system and it updates system information on the runtime and it reallocate available resources in real time, therefore today every other application is executed using this type of technique[8].

Task scheduling algorithms are categorized into various types depending upon different factors.

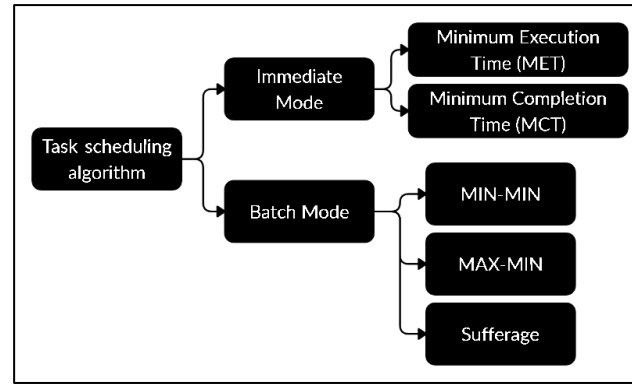

Fig. 1 Categorization of Task scheduling algorithms

3 GENETIC ALGORITHM

Genetic algorithm relies on the natural phenomenon of survival of the fittest principle, which is influenced by Darwin's natural theory of evolution. Genetic Algorithm was first introduced by a John Henry Holland back in seventies. This algorithm became popular in solving optimization problems and other large complex problems that required multiple objective requirements. Optimization problems works in way to produce better results than the present results. Therefore, optimization problems are iteratively trying to find better optimal results for a given problem[9][10].

Genetic algorithms are most popular for finding solutions to optimization problems as they can work efficiently and consider the entire solution space for finding best possible results.
Genetic algorithms are popular finding solutions with single as well as multiple objectives to given problems and due to this, genetic algorithms can be used for scheduling of tasks in cloud environments.

Genetic algorithm works in a series of steps and they can be

described briefly as:

a. *Initialization and population generation:*
Genetic Algorithm (GA) requires some initial population to start. In genetic algorithm there are chromosomes that are represented in binary. Each individual entity in population of GA is called chromosome which is further composed of genes. Now in order to use GA for task scheduling, we know that there are certain number of tasks and then there are some computing machines available. Therefore, we assign computing machines to genes and any particular scheduling technique will be referred as chromosome. Initially when the algorithm starts, due to random generation of population, the solution is not very effective or efficient. But after a certain number of iterations and multiple crossovers and mutations, the population evolves into better and more efficient and that is the whole point of using GA.

b. *Selection for mating (based on fitness function):*
In this step, chromosomes with most appropriate values are selected for further generations. The selection criteria can be determined using various techniques.

c. *Crossover:*
Crossovers are performed during new generation of population. There can be various types of crossovers that can be performed like single-point or two-point. Crossover involves exchange of genes among different chromosomes based on fitness values of individual chromosomes.

d. *Mutation:*
Mutation is performed along with crossover primarily to introduce variance in new generation.

e. *Re-iteration or Termination:*
In GA, each iteration ends with either best possible optimal solution, where entire search space has been covered, in that case the algorithm is terminated, else the algorithm uses this improved population and start all over again.

3.1 Literature Review

Cloud presents a pool of resources and like grid environments, the computational paradigm is mostly decentralized system with a vast number of computers that are interconnected to solve a complicated problem. This infrastructure facilitates communication and open access among resources in the heterogenous and decentralized environments. In grid environments, computers and servers execute autonomous functions based on their availability, capacity, efficiency and service specifications.

In order to make efficient use of computational resources in cloud, an optimum task scheduling algorithm is required. There are various task scheduling algorithms that have been presented over years.

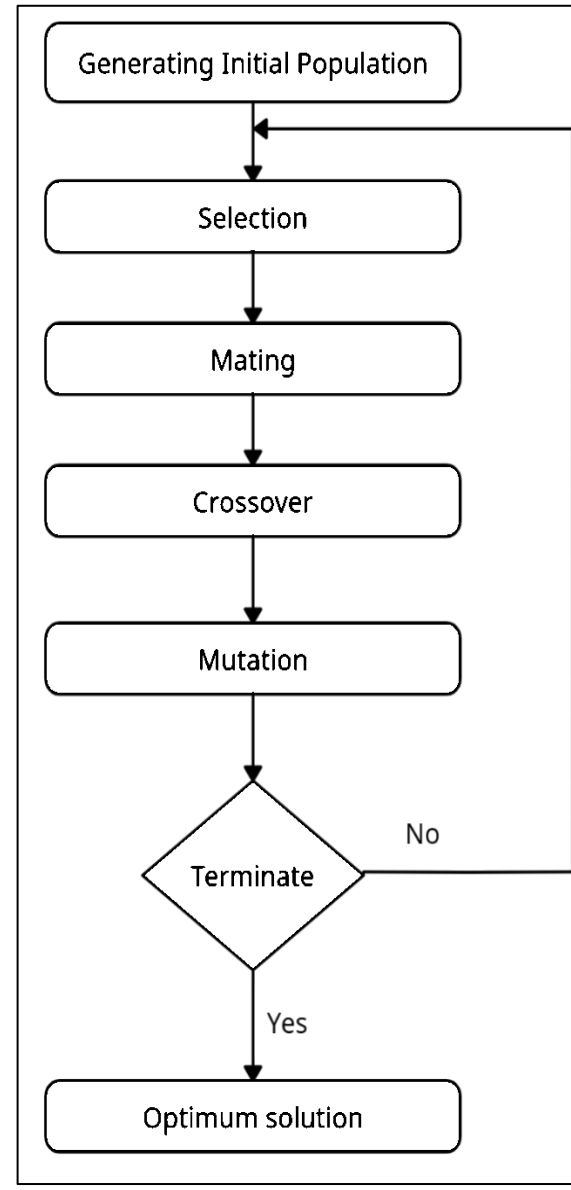

Fig. 2 Flow Chart of Genetic Algorithm

3.2 Related Work

A new algorithm is proposed in [11]. Methodology of their algorithm is that it works on basis of two traditional algorithms (MIN-MIN and MAX-MIN). Heuristic results are gathered and then some mathematical formulae like standard deviation are applied on group of jobs. Then on the basis of results, the decision is made for selection of MIN-MIN or MAX-MIN algorithm.
In this way, the proposed technique overcomes the shortcomings of both the algorithms by making the appropriate choice in any scenario. GRIDSim simulator is used for experiments, and the result shows that the performance of this algorithm is way better than conventional MIN-MIN OR MAX-MIN algorithms.
Merits: This proposed algorithm combines the benefits of both MIN-MIN and MAX-MIN algorithms and that makes the approach much better. In future, other heuristic techniques can be applied for comparison and finding optimal results.
De-Merits: The heuristic calculation approach discussed in this paper is entirely focused on of standard deviation. Another

drawback is that various operational costs (execution cost, communication cost) are not considered during calculations.

Nicholas et al. [12] proposed a novel strategy using multi-objective and Particle Swarm Optimization technique with the aim to resolve multiple conflicts during resource acquiring step and selection of appropriate resource according to task complexity. Their experimental results were promising showing decrease in overall waiting time for jobs and effective system throughput.

Seiven Leu et al. [13] presented a novel job scheduling system in grid computing. The method also tackles the issue of load balancing among available resources. In this paper an improved version Improved Hierarchy Load Balancing Algorithm (IHLBA) is presented. The modification process is that when groups of tasks arrive, their computing information is analyzed against available resources using certain threshold. If grouped tasks satisfy threshold then tasks are divided among resources else the grouped tasks are regrouped in a way that every group has equally complex task.
Job scheduling system is constantly updated from all available resource and this helps in better assignment of newer jobs to appropriate resources that ultimately reduces the job completion time.
Merits: The proposed method efficiently selects appropriate resources and assign tasks on best available options. The algorithm also works towards load balancing issue by distributing tasks for best suited resources.
Shortcomings/ Future work: There are certain shortcomings like some additional threshold measures can be added with load balancing algorithm and then it may work more efficiently under varying scenarios.

In [14], authors have proposed two different scheduling algorithms that are Dynamic Multilevel Hybrid Scheduling (DMHS) using square foot and another DMHS with the use of median. The proposed algorithm work in time quantum fashion for scheduling of jobs. Normally if we choose a small-time quantum value then it will cause excessive context switching. Alternatively, if we choose a large time quantum then it will simple start acting as FCFS leading to starvation of some of the tasks. Therefore, an optimal approach is introduced in proposed algorithms is selection of dynamic time quantum. Now time quantum is derived after performing some calculations but the end results are very promising. Basic idea behind these algorithms is to devise a method that can execute jobs in best possible manner and that implicates minimum waiting times and response times. Experimental results have shown that these algorithms significantly reduced waiting times, turnaround times and response times.
Merits: Presented approach is very scalable with increasing number of computing resources and jobs. Secondly it uses dynamic time quantum which helps in appropriate allocation of computers to available jobs. And lastly it tackles the issue of starvation because it always executes the longest job in appropriate fashion.
In this paper[15], authors have presented a scheduling strategy that uses a combination of two algorithms; MAX-MIN and Round robin algorithm. These two algorithms working together performs scheduling of tasks and increase system throughput. Basically, the proposed technique is that there is a selector algorithm that chooses either of the algorithms based on certain calculations (like prediction accuracy, available idle processors).
Initially Max-Min algorithm obtains information about tasks length and available computing resources. This information is used to calculate shortest completion time for each job. Then prioritization is performed with shortest job having maximum priority. After that a second priority list of tasks is developed using round robin.
Now the selector algorithm selects either of the tasks list based on accuracy of prediction. In this way both the scheduling algorithms works in dynamic fashion and the selector algorithm make decisions on updated information about unfinished tasks and available computing resources.
Experimental results are obtained after running multiple simulations and the results shows that presented technique works quite better than simple Max-Min or round robin algorithm.
Merits: The proposed technique efficiently chooses dynamically the better of the two algorithms based on changing information. (prediction, remaining tasks, available computing resources etc.)

Keerthike et al. in [16] presented a solution in which tasks move from overloaded resources to idle or underloaded resources. The algorithm use is Multi Criteria Scheduling Algorithm. The basic working of algorithm is that there is a central schedular system that contains information about all resources like resource availability, computational power of resource etc. When jobs are submitted to grid system, the central schedular allocates these jobs to appropriate resources on the basis of resource capability and availability.
Merits: The given strategy performs quite well and reduces significantly the execution time of tasks.

Future work: Some additional algorithms could be added for faster execution and also to reduce the communication delay occurring in grid network.

Fong et al. [17] presented a modified Genetic Algorithm. Their algorithm works in an incremental fashion and hence they named it adaptive incremental genetic algorithm.
Basically, the algorithm initially takes all the available pool of tasks and then make multiple groups. Genetic algorithm is applied to each group of tasks. Genetic operators are incrementally changed rather than random variations. With every iteration of genetic algorithm, the genetic operators are incremented. Experimental results proved their method to be much better and efficient as compared to simple min-min or max-min algorithms. The only problem that could arise is when tasks are not grouped appropriately, in-case of a few tasks in each group, the total number of groups will increase enormously.

3.3 Comparative Study of Task Schedulers

Over the last few years, a large number of task scheduling algorithms have been proposed. Some of the common algorithms have been added to this study[7]

TABLE 1
COMPARATIVE STUDY OF TASK SCHEDULING ALGORITHMS

	Algorithm	Working	Merits	Demerits
1	Priority Scheduling Algorithm[6]	In this algorithm, some kind of priority list is created (on basis of any heuristic approach) then tasks are executed in the order of list.	Works efficiently in a way that each process executes according to defined priority	Can cause starvation for small tasks with low priority
2	First Come First Serve (FCFS)[6]	Works on basic principle that tasks are executed in the manner they arrive.	Efficient, quick and easy to implement	Can create convoy effect, can cause wastage of computing resources
3	Shortest Job First (SJF)[6]	This algorithm executes smaller jobs first.	It benefits smaller jobs	Can create starvation for larger jobs.
4	MIN-MIN[11]	This algorithm looks ahead of all the available jobs and then executes smaller jobs first	The smaller the task, the earlier it gets executed	Can cause starvation for longer jobs
5	MAX-MIN [11]	Contrary to MIN MIN, this algorithm executes larger tasks first and then executes smaller tasks	Larger complex task doesn't have to wait as they are executed first.	Can create a starvation situation for smaller tasks.
6	Genetic Algorithm[18]	This algorithm helps in finding best optimal solution.	Reduced completion time and efficient throughput	Local optimum problem, Time consumption in initial population generation
7	Multi objective Particle Swarm Optimization [1]	Stochastic algorithm always tries to find best possible solution	Efficient, minimize cost	May face local optimum problem,

TABLE 2
COMPARISON OF VARIOUS LOAD BALANCING TECHNIQUES

	Author (Paper)	Parameter	Technique	Tool
1	Manisha et al. [19]	Robustness, minimum execution time	Scheduling	CloudSim
2	S.C Sharma et al. [20]	Resource utilization, Scalable	Migration and Scheduling	Increased elasticity, Open stack
3	Rekha P M et al. [21]	Cost effective, Various requests	Migration and scheduling	Cloud Analyst
4	S.Sindhu et al. [1]	Reduced cost and minimum execution time	Scheduling	CloudSim

4 CONCLUSION AND FUTURE WORK

In this report, a brief overview is performed on scheduling of tasks in cloud environments. Different modules are discussed in paper showing how cloud computing is evolved from grid computing. An important issue of load balancing, which is a major hindrance in every cloud computing paradigm is discussed. A number of task scheduling algorithms are presented with their advantages and limitations.

In order to make efficient use of computational resources in grid, an optimum task scheduling algorithm is required. There are numerous algorithms that have been presented over years. Some of the algorithms that are highlighted in
this paper is Min-Min algorithm, Max-Min algorithm, Resource Awareness Scheduling Algorithm, Round Robin, First Come First Serve and Shortest Job First. The comparative study is performed and multiple categorizations are done on various scales.

This paper also highlights Genetic Algorithm, which is a stochastic optimization algorithm and the paper also discusses the way it is used in cloud environments for the purpose of scheduling of tasks.
Although Genetic Algorithm performs much better than traditional algorithms but there are certain limitations like premature convergence of solution, issue of local optimum, and time consumption of initial population generation.

For future work, studies can be extended by adding more heuristic based algorithms. Some more advanced algorithms can also be included along with QoS heuristics that can have significant improvement in terms efficient allocation of tasks among resources.

ACKNOWLEDGMENT

The author would like to thank the esteemed faculty members of Riphah International College, Sargodha that have contributed in this research study.

REFERENCES

[1] Sreelakshmi and S. Sindhu, "Multi-Objective PSO Based Task Scheduling-A Load Balancing Approach in Cloud," *Proc. 1st Int. Conf. Innov. Inf. Commun. Technol. ICIICT 2019*, pp. 1–5, 2019.

[2] I. Gandotra, P. Abrol, P. Gupta, R. Uppal, and S. Singh, "Cloud Computing Over Cluster , Grid Computing : a Comparative Analysis," *J. Grid Distrib. Comput.*, vol. 1, no. 1, pp. 1–4, 2011.

[3] D. Wu, "Cloud computing task scheduling policy based on improved particle swarm optimization," *Proc. - 2018 Int. Conf. Virtual Real. Intell. Syst. ICVRIS 2018*, pp. 99–101, 2018.

[4] Y. T. H. Hlaing and T. T. Yee, "Static Independent Task Scheduling on Virtualized Servers in Cloud Computing Environment," *2019 Int. Conf. Adv. Inf. Technol. ICAIT 2019*, pp. 55–59, 2019.

[5] Z. Zong, "An improvement of task scheduling algorithms for green cloud computing," *15th Int. Conf. Comput. Sci. Educ. ICCSE 2020*, no. Iccse, pp. 654–657, 2020.

[6] Y. Shi and S. Kemp, "A Scheduling approach for cloud resource management," pp. 131–136, 2020.

[7] B. Anushree and V. M. Arul Xavier, "Comparative Analysis of Latest Task Scheduling Techniques in Cloud Computing environment," *Proc. 2nd Int. Conf. Comput. Methodol. Commun. ICCMC 2018*, no. Iccmc, pp. 608–611, 2018.

[8] S. G. Domanal and G. R. M. Reddy, "Load Balancing in Cloud Environment Using a Novel Hybrid Scheduling Algorithm," *Proc. - 2015 IEEE Int. Conf. Cloud Comput. Emerg. Mark. CCEM 2015*, pp. 37–42, 2016.

[9] N. Bharot and S. Shukla, "A Review on Task Scheduling in Cloud Computing using parallel Genetic Algorithm," *2020 Int. Conf. Comput. Inf. Technol. ICCIT 2020*, pp. 53–56, 2020.

[10] M. Agarwal, "A Genetic Algorithm inspired task scheduling in Cloud Computing," pp. 364–367, 2016.

[11] P. M. Naghibzadeh and Etminani, "A Min-Min Max-Min Selective Algorihtm for Grid Task Scheduling."

[12] E. S. Alkayal and N. R. Jennings, "Swarm Optimization in Cloud Computing," *41st Conf. Local Comput. Networks Work.*, pp. 17–24, 2016.

[13] Y. Lee, S. Leu, and R. Chang, "Improving job scheduling algorithms in a grid environment," *Futur. Gener. Comput. Syst.*, vol. 27, no. 8, pp. 991–998, 2011.

[14] A. K. bin Mahmood, A. Oxley, and S. N. M. Shah, "Dynamic multilevel hybrid scheduling algorithms for grid computing." 2011.

[15] C. Liang and H. Chang, "An Adaptive Task Scheduling System for Grid Computing," pp. 1–6, 2006.

[16] P. Keerthike and N. Kashuri, "A Hybrid Scheduling Algorithm with Load Balancing for Computational Grid," *Int. J. Adv. Sci. Technol.*, vol. 58, pp. 13–28, 2013.

[17] K. Duan, S. Fong, S. W. I. Siu, W. Song, and S. S. U. Guan, "Adaptive incremental Genetic Algorithm for task scheduling in cloud environments," *Symmetry (Basel).*, vol. 10, no. 5, pp. 1–13, 2018.

[18] M. Agarwal and G. M. S. Srivastava, "A genetic algorithm inspired task scheduling in cloud computing," *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2016*, pp. 364–367, 2017.

[19] R. Agarwal, N. Baghel, and M. A. Khan, "Load Balancing in Cloud Computing using Mutation Based Particle Swarm Optimization," *2020 Int. Conf. Contemp. Comput. Appl. IC3A 2020*, no. 1, pp. 191–195, 2020.

[20] M. Kumar and S. C. Sharma, "Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment," *Comput. Electr. Eng.*, vol. 69, pp. 395–411, 2018.

[21] P. M. Rekha and M. Dakshayini, "Dynamic Cost-Load Aware Service Broker Load Balancing in Virtualization Environment," *Procedia Comput. Sci.*, vol. 132, pp. 744–751, 2018.

Raveel Abdullah received the Bachelor of Science in Computer Science from University of Sargodha, Pakistan, and he is currently working towards the Master of Science in Computer Science. His research interests are in the fields of Cloud Computing, Digital Image Processing and Artificial Neural Networks.