IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 5, 2019, accepted October 22, 2019, date of publication October 28, 2019, date of current version November 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949863

Energy Management in Smart Sectors Using Fog Based Environment and Meta-Heuristic Algorithms

ZAHOOR ALI KHAN[1], (Senior Member, IEEE), AYESHA ANJUM BUTT[2], TURKI ALI ALGHAMDI[3], AISHA FATIMA[2], MARIAM AKBAR[2], MUHAMMAD RAMZAN[4,5], AND NADEEM JAVAID[2], (Senior Member, IEEE)

[1]Computer Information Science Division, Higher Colleges of Technology, Fujairah 4114, United Arab Emirates
[2]Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan
[3]Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah 11692, Saudi Arabia
[4]Department of Computer Science and IT, University of Sargodha, Sargodha 40100, Pakistan
[5]School of Systems and Technology, University of Management and Technology, Lahore 54000, Pakistan

Corresponding authors: Zahoor Ali Khan (zkhan1@hct.ac.ae) and Nadeem Javaid (nadeemjavaidqau@gmail.com)

ABSTRACT Smart Grid (SG) plays vital role in modern electricity grid. The data is increasing with the drastic increase in number of users. An efficient technology is required to handle this dramatic growth of data. Cloud computing is then used to store the data and to provide numerous services to the consumers. There are various cloud Data Centers (DC), which deal with the requests coming from consumers. However, there is a chance of delay due to the large geographical area between cloud and consumer. So, a concept of fog computing is presented to minimize the delay and to maximize the efficiency. However, the issue of load balancing is raising; as the number of consumers and services provided by fog grow. So, an enhanced mechanism is required to balance the load of fog. In this paper, a three-layered architecture comprising of cloud, fog and consumer layers is proposed. A meta-heuristic algorithm: Improved Particle Swarm Optimization with Levy Walk (IPSOLW) is proposed to balance the load of fog. Consumers send request to the fog servers, which then provide services. Further, cloud is deployed to save the records of all consumers and to provide the services to the consumers, if fog layer is failed. The proposed algorithm is then compared with existing algorithms: genetic algorithm, particle swarm optimization, binary PSO, cuckoo with levy walk and BAT. Further, service broker policies are used for efficient selection of DC. The service broker policies used in this paper are: closest data center, optimize response time, reconfigure dynamically with load and new advance service broker policy. Moreover, response time and processing time are minimized. The IPSOLW has outperformed to its counterpart algorithms with almost 4.89% better results.

INDEX TERMS Cloud computing, fog computing, smart grid, smart city, load balancing, server broker policies.

I. INTRODUCTION

In the modern era, the traditional grid is converted into Smart Grid (SG) by integrating Information with Communication Technology (ICT) with it. Further, Renewable Energy Sources (RESs) are used to reduce the usage of fossil fuels. SG provides the facility of bi-directional communication. Smart meters are used to monitor and manage the household energy consumption of the users, which minimizes the electricity bill [1]. If the users' demand is more than the generated energy, Control Energy Management System (CEMS) is used to provide energy [2]. The sole purpose of CEMS is to minimize the energy consumption and to maximize the revenue. Similarly, the community Photo Voltaic (PV) with non-cooperative Stackelberg game theory is introduced to manage the energy demand of the community consumer [3]. The solutions to fulfill the energy demand of the consumers through RES and CEMS is applicable only at the community level. Owing to the rapid increase of smart cities, smart societies, smart communities and Smart Sectors (SSs), the demand for energy is increased. Therefore, handling the consumer demand and request is also a challenging task.

The associate editor coordinating the review of this manuscript and approving it for publication was Christos Verikoukis.

So, cloud is introduced to store the drastically increasing data. Cloud Data Centers (DC) are deployed to provide numerous services. Cloud DC is a pool of Physical Machines (PMs) and there is a huge number of Virtual Machines (VMs) inside the PMs [4]. Therefore, to manage the load of PMs and to reduce the energy consumption of physical resources on a cloud, the game-based theory is applied in [5]. It is also helpful to predict future load. Another game based theory with Iterative Proximal Algorithm (IPA) is proposed to balance the consumer‘s request at the server level. The Nash equilibrium technique is used to find the minimum Response Time (RT) value [6]. However, there is a large geographical distance between cloud and consumers, which increases the delay. So, fog computing is introduced by Computer Information System COmpany (CISCO), which is the intermittent layer between cloud and consumer layer [7]. All these services are provided on the edge of the network which efficiently balances the load of a cloud. Further, an enhancement is still required to balance the load of fog servers.

Numerous techniques, mechanisms and algorithms are proposed in various current literature. In [12], to schedule the user request and minimize the energy consumption of physical resources, Particle Swarm Optimization (PSO), Binary PSO (BPSO) and BAT algorithms are proposed. The meta-heuristic algorithms give an efficient and reliable solution in both local and global search space. PSO Non-dominated Genetic Algorithm (PSONSGA) is proposed by calculating the sum of two metaheuristic algorithms. The aim of this algorithm is to solve the multi-objective optimization problem at fog level [11].

An efficient load balancing at fog layer helps to minimize the RT and Processing Time (PT) of DCs. The implemented bio-inspired algorithms take time to predict the future load and schedule the tasks on DCs according to incoming requests on the fog. If the size of a request is large then it is allocated to DC having a large number of VMs. Because of this reason, the optimal solution is given by nature-inspired algorithms. The minimum RT, PT also minimized the computing cost and maintenance cost.

This paper is the extension of [13]. Whereas, the objective of this paper is to balance the load and minimize the energy consumption of VMs and handle the user's request efficiently.

The rest of the paper is organized as follows: Section I-A provides a problem statement. Section II defines related work. The proposed system model is defined in Section III, Section IV defines the mathematical formulation of proposed work, the proposed technique is defined in Section V. Section VII defines the simulation results and discussion. The last Section VIII gives the conclusion of this paper. The Table 1 defines the abbreviations, Table 2 and Table 3 describes the Nomenclature used in this paper.

A. PROBLEM STATEMENT

Yaghmaee *et al.* proposed the multi-tier communication architecture for the transfer of energy from the cloud level to home-gateway level to overcome the issues of traditional grid [7]. The cloud, local fog nodes, and home gateway are the main components of their proposed architecture. The open automated demand response protocol is proposed at the home-gateway level, which helps the consumers to manage their energy consumption. However, no mechanism is proposed to balance the load of a cloud or fog.

Mishra *et al.* [12] used PSO, BPSO and BAT algorithms to minimize the energy consumption of PMs in fog computing environment. The proposed algorithmsoptimized the energy

TABLE 1. List of abbreviations.

Abbreviation	Full Form
ACO	Ant Colony Optimization
BPSO	Binary Particle Swarm Optimization
BCU	Building Control Unit
BW	Band Width
CDC	Closest Data Center
CEMS	Control Energy Management System
CLW	Cuckoo with Levy Walk
DAF	Dynamic After Filtering
DC	Data Center
DT	Data Transfer
EI	Energy Internet
EC	Edge Computing
ELBS	Energy Aware Load Balancing Scheduling
EPO	Elephant Herding Optimization
FC	Fog Computing
FECoIT	Fog/Edge Computing-based IoT
FCA	Fairness Cooperative Algorithm
GA	Genetic Algorithm
HABACO	Hybrid Artificial Bee Ant Colony Optimization
HEMC	Home Energy Management Controller
ICT	Information Communication Technology
IoT	Internet of Thing
IPSOLW	Improved Particle Swarm Optimization with Levy Walk
IPA	Iterative Proximal Algorithm
IaaS	Infrastructure as a Services
LEAD	Latency Aware Workload Offloading
LVMM	Layered VM Migration
MCC	Mobile Cloud Computing
MG	Micro Grid
NASBP	New Advance Service Broker Policy
ORT	Optimize Response Time
PMs	Physical Machines
PSO	Particle Swarm Optimization
PSO-SA	PSO Simulated Annealing
PT	Processing Time
RT	Response Time
PaaS	Platform as a Service
RDL	Reconfigure Dynamically with Load
SAA	Simulated Annealing Algorithm
SA	Smart Appliances
SG	Smart Grid
SH	Smart Home
SS	Smart Sector
SaaS	Software as a Service
VMs	Virtual Machines

TABLE 2. Nomenclature.

Symbol	Description
$\alpha*$	Size of Population
$\beta*$	Rate of Elitism
Ca	Fraction of Worst DC
δ	Number of Iterations
$DC*$	Defines the Number of Population including number of VMs
$\exp(-\gamma S)$	Best Direction
$f(x)$	Objective Function of CLW
(fx_i)	Current Local Best
(fpb_i)	Updated Local Best
f_gb	Global Best
$\gamma*$	Rate of Mutation
G_WT	Expenditure of Wind Turbine of MG
g_best	Global Best
ier, i	Each Iteration
Ir	Instruction of Consumer Request Length
it	Shows the Number of Iterations
$j*$	Random Selection of VMs among DC
k	Any Specific DC
KR	Current Status of the Transmitted Request
LCR_i	Length of Consumer Request
LT	Length of Consumer Instruction
L	Length
L_i	Random Best Solution
MG_cost	MG Cost
Max_it	Maximum Number of Iterations
$m^{j,iter}$	Probability of best VM
n_e	Crossover step
n_c	Mutation step
ω	Explain the Interia weight
$PRCost$	Physical Resources Cost
Pr_MG	Physical Resources Cost of MG
p_best	Best Position of the Particle
π	Instance
$pop1$	Population of Local Best
$pop2$	Population of Global Best
PVM	Population of VMs
RT_kj	RT of DC Including Fixed Number of VMs
$RecurringCost$	Recurring and Expected Life Cycle Cost
$r1$	Random p_best solution
$r2$	Random g_best solution
r_j	Random Probability
r^{ix}	Random Awareness Probability
SP_j	Processing Speed
$\sum$	Total Number (sum) of VMs
$\sum_{i=1}^{K}$	Aggregation of DCs
$\sum_{j=1}^{R}$	Aggregation of Consumers Requests
$Sig_{(i,j)}$	Sigmoid Function for Appliance Status
T	Describes the Tasks
T^{**}	24 Hours Time Slot
$T_transfer$	Total Transfer, All Time Included to Transfer the Data
$Total_latency$	Total Latency of the Network
$TotalCost_{sytem}$	Total Cost of the System Including PM and Hypervisor
$VM_n\epsilon VM$	set of VMs
VM_{TL}^{c}	Overall cost of VMs
V_Min	Minimum Velocity
V_Max	Maximum Velocity
$v(t)$	Velocity with respect to time
v_{i+1}	Updated Velocity
X_VM	Minimize the PT of VMs (defined in objective function of CLW)
${x_i}^{R*}$	Step Size According to Consumers Requests
${x_i}^{R*+1}$	LW is Performed According to Incoming Request
x_i	Position of the Vector
x_{i+1}	Position of the Vector in each Iteration
$X*$	Current Best
$x^{i,iter}$	Current Probability

TABLE 3. Specifications of fog.

Specifications of Fog	Range
Fogs	Fog 1, Fog 2, Fog 3
No. of VMs	25,50,75
Memory	512 Mb
BandWidth (BW)	1000 bits/s
Image Size	1000 bytes
VMM	Xen
Operating System (OS)	Linux
Architecture (Arch.)	X86

of physical computing resources; however, it increases the delay, which degrades the performance of the whole system.

The efficiency of RT and PT are not considered simultaneously in the aforementioned literature. Therefore, to tackle the aforementioned issues, this work is proposed to devise an integrated model for cloud and fog based computing in SG (using nature-inspired algorithm). The aim of this work is to minimize RT of DCs and PT of VMs along with the cost.

B. CONTRIBUTIONS

The contributions of this work are described as:

- Three-layered architecture is proposed in SG environment: cloud layer, fog layer, and the consumer layer. The dynamic and static DCs are considered to check the adaptivity of the proposed work.
- An Improved PSO with Levy Walk (IPSOLW) is proposed to balance the consumers' requests at fog level.
- The cost of VMs, MGs and DC is also minimized.
- The consumers' request is balanced on the fog layer to minimize the RT and PT.
- Two scenarios are considered to check the performance of the proposed system.

II. RELATED WORK

In [16], Xu *et al.* proposed Dynamic Resource Allocation Method (DRAM) to balance the load of computing nodes in cloud and fog environment. The proposed method allocates static resources using dynamic scheduling. However, the computational time of cloud and fog is not considered. In [17], Luo *et al.* proposed multi fog architecture and threshold algorithm. The multi fog architecture is proposed to enhance the efficiency of the multi-cloud. The algorithm is proposed to utilize the fog nodes efficiently and to reduce the delay. Due to their efficient utilization, the RT and PT of fog nodes are improved. In [8], Liao *et al.* proposed pricing incentive Simulated Annealing Algorithm (SAA) and Dynamic Resource Allocation (DRA) with a vehicle service framework. Fog based architecture was proposed to overcomes the peculiarities of the cloud. The proposed architecture considers the issue of unbalanced computing resource demands, which improves the flexibility of the traditional cloud system and balances the internet of vehicles. However, the computational time of fog node is not considered which effects the performance of the whole system.

The RT of mobile users is minimized in order to upload their applications on geographical distributed clouds. Latency Aware Workload Offloading (LEAD) strategy is proposed in [18]. The aim of the proposed strategy is to decrease the average RT of the request when the load demand is sent to the cloud. The proposed LEAD algorithm is then compared with two other algorithms namely: location-aware proposed algorithm and remote location-aware algorithm. The performance of the proposed algorithm is also evaluated. On one side, the RT of mobile users is minimized. On the other hand, PT of this LEAD algorithm is also considered.

The game-based consolidation method for VMs is proposed in [5]. The aim of this algorithm is to minimize resource utilization and to balance a load of PMs on cloud DCs. Two steps are performed to achieve the object. Firstly, every measured value of resource load is tested by t-test. Secondly, all online PMs are grouped. The future load of these resources is predicted by the grey theory. Cloudsim is used to calculate the results of the proposed method. The resource utilization is minimized by avoiding unnecessary VM migration. In [19], the authors proposed a tailored optimization method. This optimization method is proposed with a fog based environment at the edge of the network, which is the extension of the cloud. The aim of the proposed optimization method is to handle the heterogeneous scale of the fog. Delay is reduced almost 90% with proposed method. The minimization of the NP-hard problem is also considered in this work. In contrary, placement of VMs become a considerable issue for the fog. The wide range of applications related to the Internet of Things (IoTs) are proposed in literature. Several works are done on different layers of the communication system. However, interoperability is still not considered. This issue is approached by Negash *et al.* [20]. The web virtual of things is deployed at the middle fog layer. The aim of this method is to evaluate performance, resource utilization, etc. The implementation results show efficient resource utilization with improved performance and interoperability. However, information on mobile data is not stored properly.

Energy Internet plays an important role to utilize the RES efficiently and intelligently. The authors in [21] discussed the efficient forecasting and optimum utilization of energy. A hierarchical integration architecture is designed for energy Internet. Further, they proposed an energy forecasting and enforcement learning scheme is proposed. The aim of the proposed work is to minimize the energy cost based on heuristic learning and to predict the load of energy. For this purpose, they used cloud and fog based architecture and results are evaluated on the basis of E-matrix and RL approach. The proposed work efficiently predicts the load and minimizes the energy cost. Still, the performance and PT of the proposed schemes is not discussed.

The Mobile Cloud Computing (MCC) is known as an energy-efficient approach which uploads the tasks on cloud resources. The problem which is discussed in [22] is how to load applications on cloud computing in offloading mode. The agent-based MCC framework is proposed to enable the requests on cloud resources. The aim of the proposed work is to increase energy savings among multiple users. The Dynamic After Filtering (DAF) algorithm is proposed to solve the optimization problem of the proposed framework which increases the delay. Fog computing is then used to solve the aforementioned problem.

In literature, fog is used to extend the services of cloud. Additionally, it is used in the industrial side to enhance the scalability and to achieve better performance of the services. Tseng *et al.* [24] tackles the problem of compensation between scalability and operational cost. To overcome these issues, hypervisor with integrated virtualization fog platform is introduced and named as fuzzy-based time auto-scaling. The aim of the proposed work is to provide the solution of auto-scaling and low cost. The open-source Unix Bench is used to evaluate the performance of the proposed platform. After evaluation, they achieved accurate auto-scaling and less operational cost. However, the error rate and the delay increase.

Rehmani *et al.* [24] discussed the rapid changes IoT and ICT bring in Smart Homes (SHs), societies and industries. In this work, they concentrated on smart health care using a fog based environment. They proposed geographically distributed sensor nodes between fog and cloud to resolve the energy efficiency, reliability and scalability issues. They also implemented an IoT based health monitoring system which enhances the efficiency and reliability of the proposed system. However, the cost is not discussed.

The fog and edge-based architecture are proposed in [25] which is known as Fog/Edge Computing-based IoT (FECoIT). The proposed architecture with a cyber-physical system is proposed to make the intelligent transportation system of SG, SHs, and societies. The cyber-physical transportation system controls the flow of requests from SHs to edge and fog. Netlog and Cooja are used for simulations. The fog provides services at the edge of the network with less latency. However, fog provides limited services.

Kadhim and Seno [26] proposed resource management strategy and load balancing algorithm. The aim of presenting these two load balancing algorithms is to optimize the utilization of fog servers. The desire of scheduled strategy is to prevent the transference of tasks to the cloud. The authors of this work performed the simulations using OMNET++. After simulating the techniques, they calculated the percentage of resource utilization, bandwidth usage, and meeting deadline. Moreover, RT is also calculated. The latency is decreased and request transfer rate to the cloud is also minimized. On the other hand, RT and PT are interrelated with each other. However, PT is not calculated. The Energy-Aware Load Balancing and Scheduling (ELBS) and PSO are proposed in [27] for optimal scheduling and load balancing using fog computing nodes. After performing the experiments using PSO, authors calculated operational time and load balancing performance. They achieved improvement in the life cycle and load balancing on fog computing nodes due to proposed ELBS and PSO. However, authors did not calculate the cost of computing nodes.

The fog computing reduces the transmission delay and alleviates the congestion. Xiao and Krunz [28] focused on the energy-efficient design of fog computing to support low latency in internet applications. They also investigated RT of the end-users and power usage of a fog node. They noticed an existing tradeoff between optimization framework and power usage of the fog nodes. To resolve these issues in an efficient manner, fog platform is used to forward offloading requests, in which different fog nodes cooperate with each other. The sub gradient method with dual decomposition and the distributed Alternating Direction Method of Multipliers via Variable Splitting (ADMMVS) is proposed to reduce the RT with efficient energy consumption. However, the authors noticed a tradeoff between fog nodes of the forwarding offloading requests. In [29], the authors discussed cloud. The host of cloud servers expands themselves according to the requirement. The data is stored and processed simultaneously on the cloud. In the meantime, as the SDN based technology increases the requests on the cloud also increases rapidly, due to which the resource management becomes a challenging ask for the cloud. To solve this issue, Cloud-Pi with a low-cost testbed at Melbourne laboratory is introduced. The sole purpose of the work done is the efficient utilization of resources. VMs are allocated in an efficient manner. However, the proposed work is applicable on a small scale area.

In [30], the authors discussed that the main purpose of the node-based architecture is to provide services at minimum cost. To fulfill this aim, Tabu search method is introduced. The aim of this method is to minimize the computational cost and optimize the RT. The resources are allocated to the requests and linear programming is used to calculate the results. The usage of cloud and fog resources is maximized; on the other side, cost is reduced. For optimum utilization of resources in SHs, buildings, and societies, Edge Computing (EC) is proposed in [31]. The aim of the proposed method is to optimize the resource and transfer load from cloud to edge. The EC performs better in a cloud computing environment. EC and fog computing provide elasticity to the resources that provide distributed processing to overcome the drawback of central architecture. However, the implementation cost is very high.

The cloud and fog based environment with a Hybrid Artificial Bee Ant Colony Optimization (HABACO) load-balancing algorithm is proposed to improve the performance of the cloud. In this paper, authors worked on load minimization on cloud and fog processing nodes. Cloudsim is used to calculate the RT, PT and cost of physical resources. However, the utilization of physical resources increases in the current work [32]. In SG environment, the cloud and fog are introduced. The energy-efficient approach is proposed to manage the energy consumption of the fog environment. However, the computational time of fog and cloud is not monitored which may increase the chances of delay.

In [34], the concept of 5G-Home Energy Management Controller (HEMC) is introduced in cloud and fog based environment. The aim of this HEMC is to enhance the performance of fog by giving a quick response to the consumers' request with minimum delay. Cloud analyst is used for simulations. They optimized RT, PT, and cost for two different scenarios.

The authors in [35] minimized the RT and PT of cloud and fog computing nodes in SG. They introduced fog based system; further, PSO-Simulated Annealing (PSO-SA) and New Service Broker Policy (NSBP) are proposed. They optimized RT and PT of fog and cloud computing nodes. However, operational cost is high.

The proximal Jacobian ADMM is proposed to minimize the calculation cost of the cloud DCs. Authors reduced the gauge cost of the cloud nodes by deploying fog in their proposed architecture. Still, RT and PT of cloud and fog computing nodes are not calculated in their work [9]. Elephant Herding Optimization (EHO) with genetic, firefly, and BPSO are proposed to handle the load of SHs [36]. The optimization techniques are proposed to optimize the load of SHs and to minimize the electric bill cost. However, when there is a large number of homes or a large society, it is difficult for the SH controller to manage the load of all homes. So, the authors used smart meters for the bi-directional communication between consumers and fog.

The authors in [43] proposed an auction mechanism for demand response. This work is done to incentivize the cloud service providers. These providers operate the geographically distributed clouds. SGs submit bids to obtain demand response. The authors in [44] proposed an efficient mechanism to reduce the operational cost and minimizes the raising energy consumption. Further, fine grained differential method and precise power capping are presented to enhance the performance of cloud DCs.

A lot of work has been done to efficiently minimize the computational cost and energy consumption in DCs. The authors in [45] proposed an incentive mechanism and server sharing incentive mechanism. An incentive is given to tenants. This is done to motivate them to minimize the energy consumption. The authors in [46] proposed an incentive mechanism for cloudlets to minimize the energy consumption. Quality of services is also ensured. Further, the proposed bidding policy outperformed other techniques in literature. However, there is a chance of congestion which may increases the delay. In [47], the authors proposed an iteration-based algorithm to solve the aforementioned problem. This mechanism is done especially for mobile clouds while minimizing the computational cost along with energy consumption. Further, quality of services are also enhanced. However, an efficient mechanism is still required to handle cloud DCs as the number of users and the services are increasing drastically.

III. PROPOSED SYSTEM MODEL

In this paper, three layered architecture is proposed, i.e., cloud, fog and consumer layer. Cloud layer contains various cloud DC to provide services to the consumers and to store the data of all users permanently. However, it is far away from the consumer layer which increases the delay. So, CISCO introduced fog computing, which is an intermittent layer between cloud and consumers layer to solve the aforementioned problem. Consumer layer consists of cluster of Smart Sectors (SSs), buildings, Building Control Unit (BCU) and smart city.

The top most layer in the proposed system is cloud layer as shown in Fig. 1. Cloud provides numerous services to the consumers. Generally, these services are categorized into three types: first is Infrastructure as a Services (IaaS), which provides hardware services, e.g., Amazon EC2. Second is Platform as a Service (PaaS), which provides a platform to the consumers to run specific applications, e.g., Google Application Engine. Third is Software as a Service (SaaS) [37]. The service provider, utility, and wholesale market are connected with the cloud. The current rate of the electricity is taken from the wholesale market and then transmitted to the fog servers.

Fog layer is an intermittent between cloud and consumer layer to minimize the delay and provide the services at the edge of the network [7]. Further, it stores data for sometime and then send it to the cloud for permanent storage. It has data of all Micro Grids (MGs) and consumers along with their energy consumption. Consumers send request to the fog layer, it will first check the status, if the size of the request is less than the defined threshold, then the request is forwarded to the MG and it will provide energy to the SHs. Several load balancing algorithms are used to efficiently balance the load of fog servers. Further, service broker policies are used to device which DC will fulfill the requirement of consumers. The allocation of requests to DC depends on the size and load of the fog DC. If fog is unable to meet the requirement of user, it will send the request to the cloud. The detailed specifications of these three fog servers are defined in Table 3.

Then, there is a consumer layer in which smart city has three SSs and each SS has two clusters of buildings. There are 500 numbers of smart buildings in scenario 1 and 1000 number of smart buildings in scenario 2. The smart buildings are made up of SHs and Smart Appliances (SAs), which are controlled by BCU. Moreover, their energy consumption is also monitored by BCUs. Further, two MGs are placed near each SS to provide energy. The MG consists of renewable resources, i.e., wind turbines, solar panels, generators, and batteries. The additional or remaining energy of one MG can be shared with other cluster neighbor located in the same SS. The proposed model is graphically represented in 1. We proposed IPSOLW to efficiently balance the load of fog layer, which minimizes the delay.

IV. MATHEMATICAL FORMULATION

There are several consumers who send requests to top layers. The set of consumers are represented as $C = \{C_1, C_2, C_3, \ldots, C_n\}$. The set of requests sent by customers are $CR = \{Cr_1, Cr_2, Cr_3, \ldots, Cr_n\}$. The CR processed by the fog layer optimizes the power consumption of the consumer. The requests that are generated from the clusters of SSs have various sizes. In proposed scenario 1, the size of the request is fixed, i.e., 5000 and in scenario 2, the size of requests are generated randomly between the range of 1000-5000. When the consumers send requests to the fog for some specific services, the service broker policies allocate requests to the DCs according to their required demand.

$$T_{transfer} = Y/BW_{pressure}. \tag{1}$$

where,

$$BW_{pressure} = BW_{total}/KR. \tag{2}$$

BW_{total} is the total available bandwidth over the Internet. Y is the single request, and KR is the current status of the transmitted request. The delay which occurred during this procedure is known as transmission delay. Eq. 1 and 2 are taken from [37]. Transmission delay is calculated in Eq. 3:

$$Total_{transmissiondelay} = Total_{latency} + Total_{transfer}. \tag{3}$$

$Total_{latency}$ is the total latency of the network. $Total_{transfer}$ includes all the time needed to transfer data.

The number of VMs required to run the consumers' tasks are given as $VM = \{VM_1, VM_2, VM_3, \ldots, VM_n\}$. The V^* presents the set of VMs and the processing speed of VMs is calculated in Million Instructions Per Second (MIPS). The length of consumers' requests is denoted by $LCRi$. The SPj is the processing speed in MIPS. The processing speed is calculated in Eq. 4. The formula to calculate processing speed is also defined in [12].

$$SPj = \frac{LCRi}{SPj}. \tag{4}$$

The maximum time taken by VMs to complete the consumers' tasks is known as makespan. Eq. 5 shows the calculation of the makespan of consumer requests [12] and [35]. Where, CR_{rj} shows the total completion or total execution time of the consumer request.

$$MakeSpan(CR_{rj}). \tag{5}$$

The objective is to maximize the performance and minimize the RT,

$$Performance_k = 1/ExecutionTime_k. \tag{6}$$

Eq. 6 describes the execution time of DC.

Where, $1/ExecutionTime_k$ defines the RT of any specific DC and 1 represents the processing or execution time of that specific DC. This equation is defined in [49].

(CR_{rj}) defines the completion time of consumers' requests allocated to VMs. Eq. 7 defines the overall $RT_{k,j}$ of DCs on the basis of defined VMs. Eqs. 6-11 are inspired from [34], [35] and [36].

$$RT_{k,j} = \sum_k \varepsilon VMn(CR_{rj})/Makespan \times setofVMs \tag{7}$$

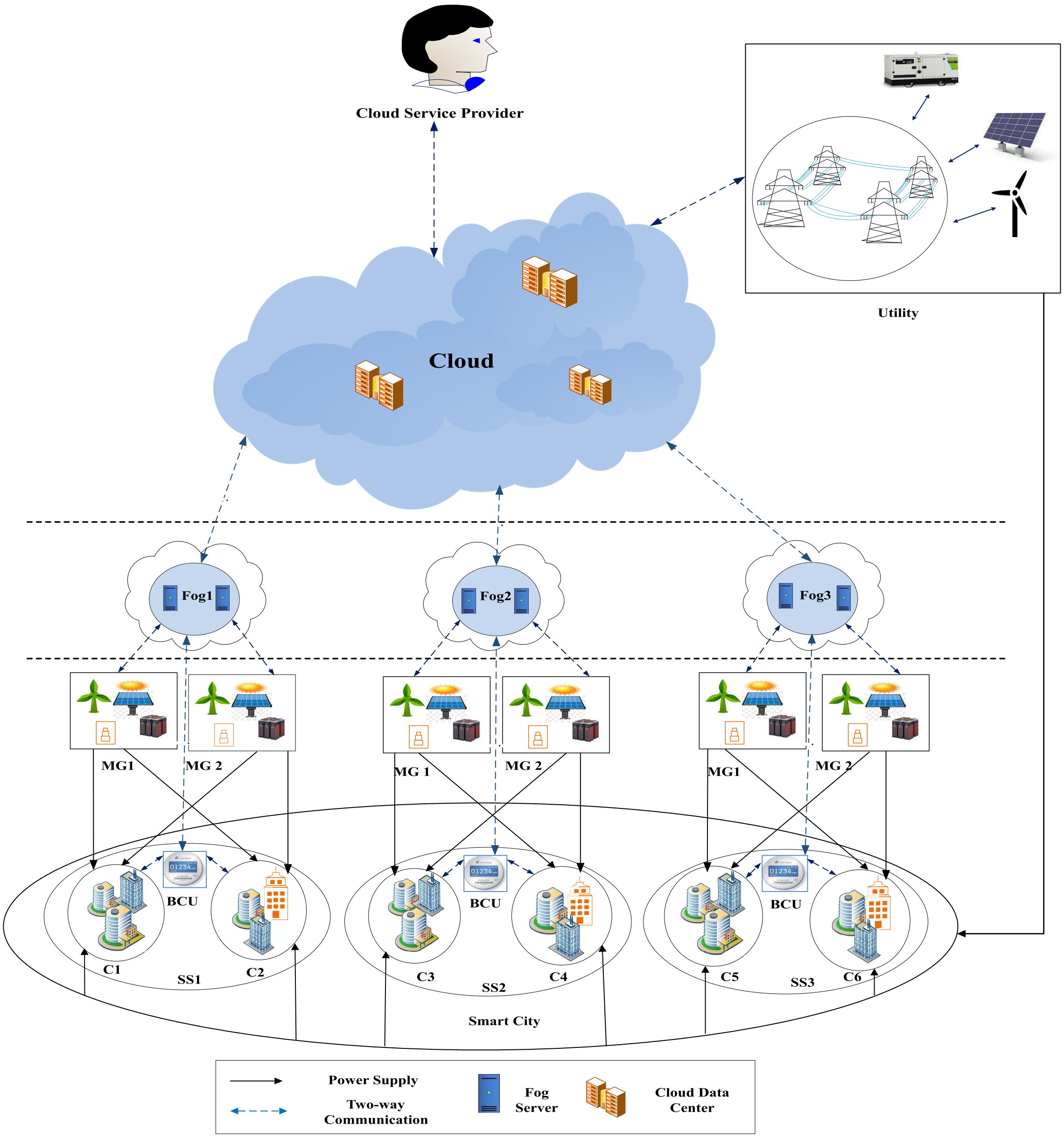


FIGURE 1. Three layered architecture.

The second objective is to minimize the PT of VMs,

$$PT = \sum_{i=1}^{K} \sum_{j=1}^{R} (PT_{K,R} \times \alpha_{K,R}). \quad (8)$$

The aggregation of $\sum_{i=1}^{K}$ DCs and consumer requests $\sum_{j=1}^{R}$ are shown in Eq. 8. It means that PT is calculated on the basis of the defined number of fogs and consumer requests that are considered in the two scenarios.

The estimated cost per VM is of two types, i.e., total fixed cost and recurring cost. These types of costs depend on the deployment of VMs. *TotalFixedCostofVM* describes the cost of deployed VMs. *Recurringcost* or expected life cycle cost depends on the cost of the physical resources that are used to meet the consumer's requirements. Eq. 9 defines the estimated cost for each VM.

$$EstimatedVMcost = TotalCostofVM + Recurringcost. \quad (9)$$

This equation defines the total cost of VMs. *PRcost* is the cost of the physical resources on which VMs are implemented. The implemented cost is the cost of VMs. The maintenance cost of each VM is also included in this cost. The software cost that is installed on the VM to run the applications and programs related to consumer request are also included in this cost. The number of requests in terms of instructions are given as *Ir* and the length of instructions given as *LT* are used to define the recurring cost in this work, $Ir \times LT$ are the number of instructions executed on the defined number of VMs per second and calculated in terms of MIPS.

The cost of a VM is calculated by the number of instructions executed in a given time. The Eq. 10 defines the number of instructions running on the VMs. The cost is calculated on the basis of instructions run on any VM.

$$VM^{c}_{TL} = (Ir \times costMIPS). \tag{10}$$

where, VM^{c}_{TL} defines the overall cost of VMs, c defines the cost, T describes the task and L is the length of the incoming consumer request. The overall cost of VMs is calculated in Eq. 11. This cost depends on the length of the tasks running on fixed number of VMs.

$$Cost_{VM} = \sum_{LT=i}^{Z} VM^{C}_{LT}. \tag{11}$$

The MG is deployed between the consumer and fog layer. MG supplies energy to the consumer and the whole energy of MG depends on RES. The cost of MG is calculated in Eq. 12 G_{SP} defines the overall expenses of Solar Panel (SP), G_{FC} represents the Fuel Cell (FC). The G_{PV} describes the PV cost. The Wind Turbine (WT) of MGs are described as G_{WT}.

$$MG_{Cost} = G_{SP} + G_{FC} + G_{PV} + G_{WT}. \tag{12}$$

The total cost of MG of the system is defined in Eq 13.

$$MG_{cost} = MG_{cost} \times Pr_{MG}. \tag{13}$$

Pr_{MG} defines the physical resources cost of MG. The Eq. 14, describes the total cost of the system. The VM cost defines the over all cost of the VMs having fixed size. The cost of MG is the overall cost of RES.

To calculate the cost of VMs, the cost of per VM server including PMs and hypervisor, etc., are all defined. These cost equations are taken from the [35].

$$TotalCost_{system} = Cost_{VM} + Cost_{MG} + Cost_{DT}. \tag{14}$$

These cost equations are defined and calculated in [35].

V. LOAD BALANCING ALGORITHM

When a huge number of requests come to the fog, it is necessary to make an efficient utilization of fog resources. There are a number of VMs inside PMs to execute the consumers' requests. The performance of these hardware resources depends on the size of VMs and capacity. The fog has a large number of DCs containing a large number of PMs. In this paper, user requests are allocated to the defined number of VMs. A mechanism is required to balance the load of rapidly increasing consumers. Various meta-heuristic algorithms are proposed to manage the consumers' request on the fog. In this paper, IPSOLW is proposed to balance the requests of the consumers. The main aim to implement a bio-inspired algorithm is that it gives the best optimal solution. Further, meta-heuristic techniques are not greedy and are problem independent. These techniques allow to explore the search space and give efficient solutions according to global and local search space. PSO and IPSOLW are explained below.

A. PSO

PSO mimics the behavior of flocking birds and school of fishes. However, it stucks in local optima and there is a problem of premature convergence in it. Following equation is used to find the position of a particle [12]:

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)} \tag{15}$$

where $X_i^{(t+1)}$ represents the position of ith particle at iteration t+1, $V_i^{(t+1)}$ is the velocity of ith particle at iteration t+1 and V_i is the velocity vector. Following equation is used to find the velocity of a particle [12]:

$$\begin{aligned} V_i^{(t+1)} = \omega \times V_i^{(t)} + c1 \times rand() \oplus (pbest_i - X_i^t) \\ + c2 \times rand() \oplus (gbest - X_i^t) \end{aligned} \tag{16}$$

where ω is inertia weight, c1 is cognitive weighting factor, c2 is a social weighting factor, rand () is a stochastic component of algorithm which is 0.5 and $\oplus$ shows the element by element multiplication.

B. IPSOLW

In this paper, different meta-heuristic algorithms are investigated or evaluated. An IPSOLW is proposed, which updates the velocity of a particle with LW. The purpose of this algorithm is to overcome the limitations of PSO, i.e., premature convergence of PSO and stucking in local optima. An IPSOLW takes long jumps toward global optimization to overcome the issues of PSO. Initially, in the search space, the particles are randomly distributed, similar to PSO. Then, LW is used to update the velocity of each particle. The position of a particle is calculated using Equation 17.

$$X_i^{t+1} = V_i^{t+1} \tag{17}$$

and

$$Levy_{walk}(X_i^{(t)}) = (X_i^{(t)}) + step \oplus random(size(X_i)) \tag{18}$$

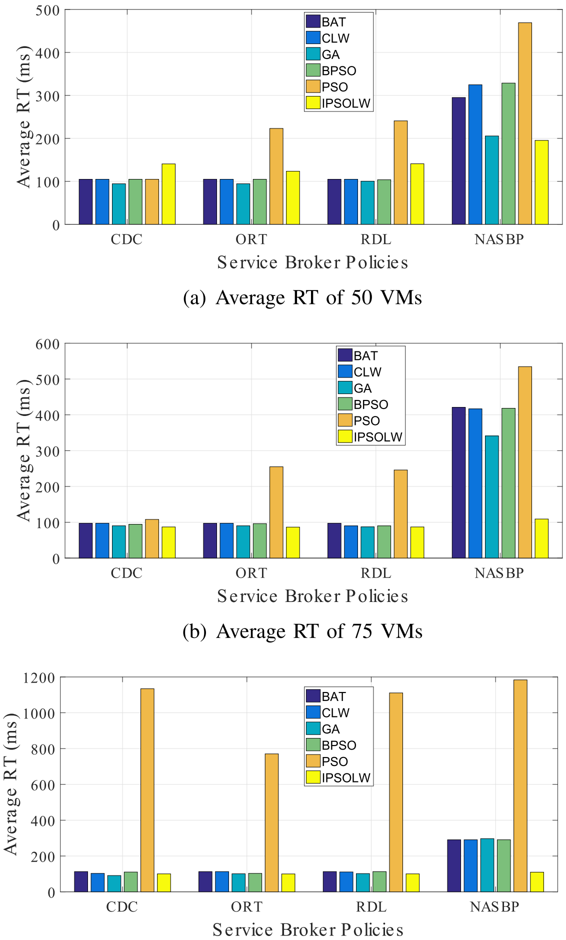
where

$$step = stepsize \oplus X_i^{(t)} \tag{19}$$

In this work, it is used to allocate the consumers' requests to the implemented resources and performs load balancing. The proposed algorithm exhibits the nature of both the swarm and the levy random behavior. The algorithm works similar to the nature of the swarm, i.e., particles search for best position. When the best position is found, it updates the local best position of particle. In the proposed algorithm, the velocity of particle swarm is updated with LW because of its premature convergence. It initializes VMs and fogs in the case of load balancing. Besides, the probability of fitness is calculated with respect to DC. Our environment is fog based, so the fogs act as a source of the best position. Therefore,

Algorithm 1 IPSOLW

Start
Initialize the position of particles randomly P inside the swarm;
Search for list of VMs and DC;
$j = DCs$;
$i = VMs$;
for t = 1:24 **do**
  Let Y is a random position of search space;
  Evaluate the position of the DC;
  Initialize the memory of each VM;
  Update the current position of the
  particle with current best position;
  While iter $\leq iter^{max}$
  Determine the $VMsize$;
  Calculate transmission delay using equation 3;
  Compute the RT using equation 7;
  Compute the PT using equation 8;
  for $i = 1 : DC$ **do**
    Randomly LW get a VM j to follow DC i;
    Define awareness probability;
    if $r^j \geq AP^{j,iter}$ **then**
      $x^{i,iter_1} = x^{i,iter} + r^i X(m^{j,iter} - x^{i,iter})$
    else
      $x^{i,iter+1} = Y*$;
    end if
  end for
  end while
  Check the feasibility of new fog;
  Evaluate the new position of the PVM;
end for



(a) Average RT of 50 VMs

(b) Average RT of 75 VMs

(c) Average RT of 100 VMs

FIGURE 2. Average response time.

the implemented load balancing algorithms help VMs to find the best feasible solution in random search space with the LW, it provides less optimization value in less time.

VI. SERVICE BROKER POLICIES

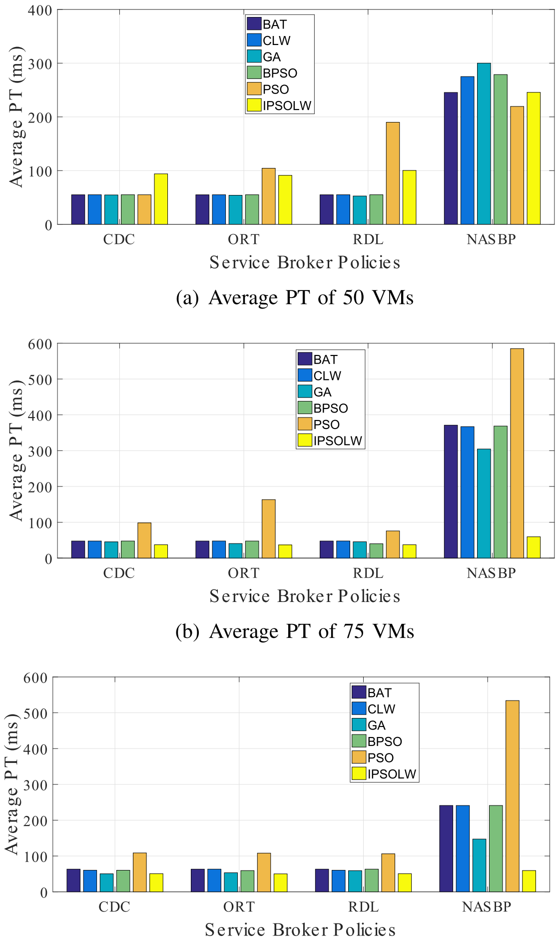
The services broker policies are used to select the suitable DC which provides the services to the consumers efficiently. The fog is selected on the basis of RT, PT, size of request, load on the DC, etc. The request is allocated to the DC according to the policy used. The CDC, ORT, and RDL are the service broker policies, which are defined in [37]. CDC selects the DC, which is closest to the user to minimize latency; the selection of DC in ORT depends on the RT; while RDL dynamically selects the DC, either considering distance or response time. Further, NASBP is described in [35]; which selects the potential fog from the same region by checking its history; and named as ORTPolicy.

VII. RESULTS AND SIMULATIONS

The simulations are done using "Cloud Analyst tool" for 24 hours [37]. This paper considers two different scenarios to evaluate meta-heuristic techniques and service broker policies. The sets of 50, 75 and 100 VMs are considered to run different applications. The BAT, CLW, PSO, BPSO, and GA are implemented and compared with IPSOLW algorithm. A smart city with three SSs is considered to evaluate the results of both scenarios. In the city, each SS has its own fog. These VMs are located on 5-20 number of PMs. There are two clusters in each sector. For implementation, the X86 architecture with Linux OS and Xen virtual machine manager is used. The speed of fog processors is 1000 MIPS. Further, two scenarios are considered to check the adaptivity of the proposed algorithm. In first scenario, the requests size per hour is static, i.e., 1000 requests while in second scenario, the size of requests per hour is dynamic, i.e., between 1000-5000. In the proposed scenarios, simulations are done using "CloudAnalyst" tool [37].

A. SCENARIO 1

In this scenario, the request of consumers from three cities is calculated and processed by fixed number of VMs, i.e., 50 VMs, 75 VMs and 100 VMs. Fixed size request is generated, i.e., 1000 requests per hour from consumer side. The static environment is considered in which all service broker policies along with service broker policies are evaluated. The RT depends on the distance of DCs in fog.

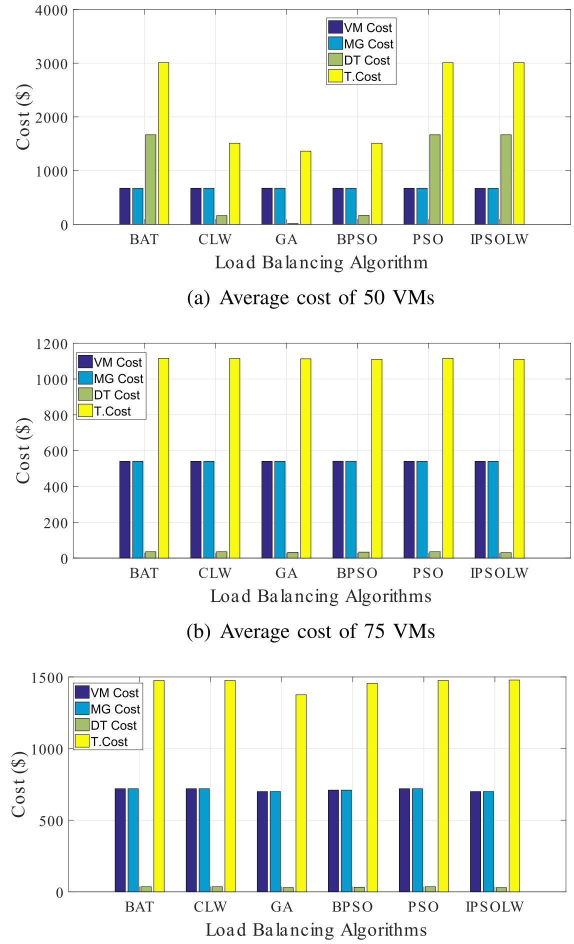


(a) Average PT of 50 VMs

(b) Average PT of 75 VMs

(c) Average PT of 100 VMs

FIGURE 3. Average processing time.



(a) Average cost of 50 VMs

(b) Average cost of 75 VMs

(c) Average cost of 100 VMs

FIGURE 4. Cost.

1) RESPONSE TIME

The time interval between sending a request and receiving a response from DC through the Internet is known as RT [37]. The proposed IPSOLW is compared with GA, BAT, CLW, PSO, BPSO and their RT is computed. Further, the RT of the load balancing algorithms is calculated with four service broker policies: CDC, ORT, RDL and NASBP. The RT is calculated in terms of Milli Seconds (ms).

In Fig. 2(a), the RT of 50 VMs is shown for scenario one. The RT of proposed IPSOLW with three different broker policies, i.e., CDC, ORT and RDL is better than the existing algorithms. However, the RT of IPSOLW with NASBP is not better, i.e., it takes 195.31 ms to respond because the requests are being assigned to those DCs, which are located near and are having a large number of requests. The VMs take time to respond to consumer request, which increases the RT. The RT of 75 VMs is calculated and graphically represented in Fig. 2(b) for fixed number of request size. The overall RT of IPSOLW is minimum because of LW. LW takes long jumps toward the optimality. IPSOLW achieved 4.86% better RT as compared with existing techniques. The RT of 100 VMs for 5000 requests is shown in 2(c). The proposed IPSOLW outperformed the counterparts. It achieved 6.07% better RT with CDC service broker policy. However, the overall RT of the PSO is not good because of its premature convergence.

2) PROCESSING TIME

The PT depends on the time taken by VMs to process the request [37]. The PT also depends on the size of the requests. The RT and PT are interrelated. If the VMs take large time to process the consumers' request the DCs will take more time to give a response to consumers' requests [35]. The hardware resources also effect PT of VMs. The larger the size of hardware in DC is, more optimal the PT will be. The PT of 50, 75 and 100 VMs is shown in Fig. [3(a)-3(c)], respectively. The PT of proposed IPSOLW with NASBP achieved 15.70%, 2.89% and 4.89% better results respectively with 50, 75 and 100 VMs.

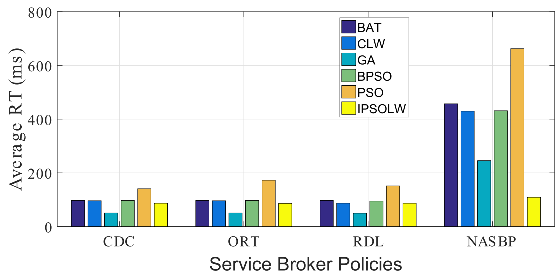
3) COST

The cost for using different load balancing and meta-heuristic algorithms are calculated for scenario 1. The cost of defined SSs on the basis of 5000 fixed requests are given in Table 4. The total cost of these algorithms is the sum of VM, MG, and

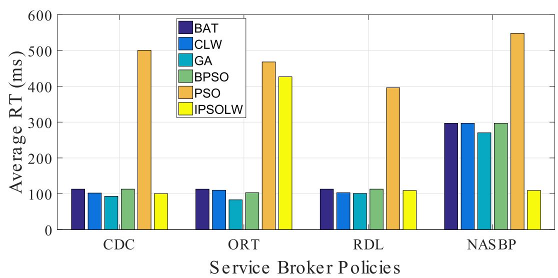
TABLE 4. Cost of scenario 1.

Fogs	VM Cost ($)	MG Cost ($)	DT Cost ($)	Total Cost ($)
Fog 1	540.02	540.02	35.36	1115.4
Fog 2	541.00	539.05	35.30	1115.39
Fog 3	543.94	547.28	35.33	1125.55

(a) Average RT of 50 VMs



(b) Average RT of 75 VMs



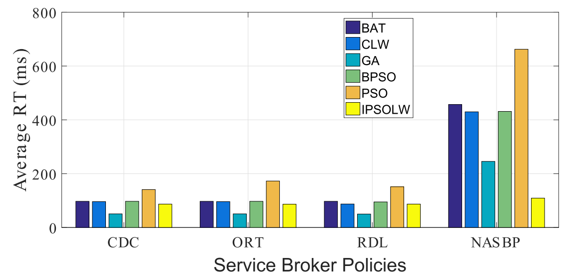
(c) Average RT of 100 VMs

FIGURE 5. Average response time.

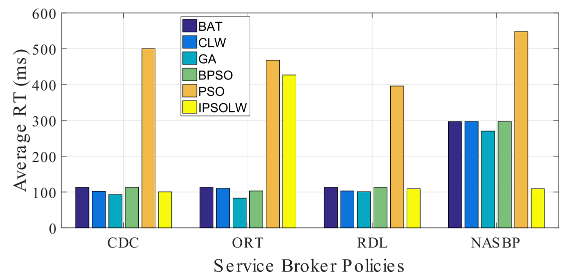
DT cost. The physical resources of DCs effect the VM cost. The VM, MG, and DT cost of all proposed and implemented balancing algorithms are mostly the same in the case of 50, 75 and 100 VMs because of the fixed number of VMs. The cost of MG is also the same in each policy because the MG assigned to every SS has the same number of RESs. The cost is calculated in terms of dollar $.

The Fog 1, Fog 2 and Fog 3 defines the number of fogs assigned to the SSs. The VM cost is almost same for all fogs because of the defined number of VMs. This VM cost consists of size, usage and recurring cost of VM. MG cost includes: cost of G_SP, G_FC, G_PV and G_WT. The cost of MG is also the same because the resources assigned to SSs are also same. The DT of the defined scenario is also same because of

(a) Average PT of 50 VMs



(b) Average PT of 75 VMs



(c) Average PT of 100 VMs

FIGURE 6. Average processing time.

fixed request size. Fig. [4(a)-4(c) show the cost for 50, 75 and 100 VMs, respectively.

B. SCENARIO 2

In the second scenario, static environment is considered in which the request size varies between 1000-5000. However, the number of VMs is same, i.e., 50, 75 and 100. The specifications of implemented hardware are same as mentioned in scenario 1.

1) RESPONSE TIME

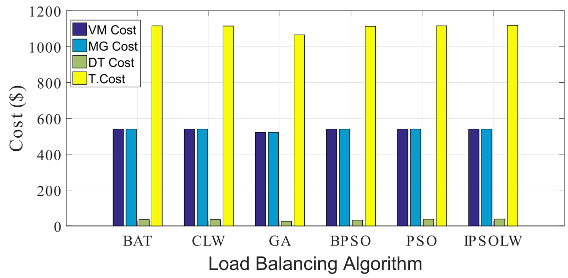
The RT is calculated for dynamically generated requests from the clusters of SSs. This scenario is considered to verify the adaptivity of the proposed algorithm for dynamically allocated requests. The RT of IPSOLW for NASBP achieved 4.61%, 4.61% and 6.00% better results with 50, 75 and 100 VMs, respectively. The RT for scenario two is shown in Figs. 5(a)-5(c).

2) PROCESSING TIME

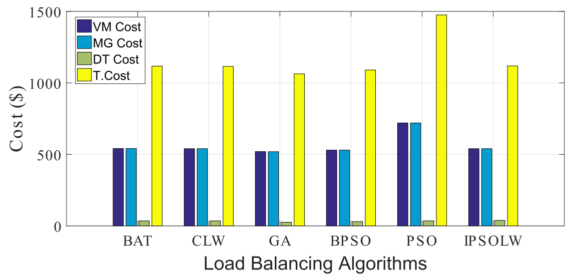
The PT of 50, 75 and 100 VMs is graphically shown in Fig. 6(a)-6(c). The proposed algorithm outperformed the existing techniques. It achieved almost 5.21% better results.

TABLE 5. Cost of scenario 2.

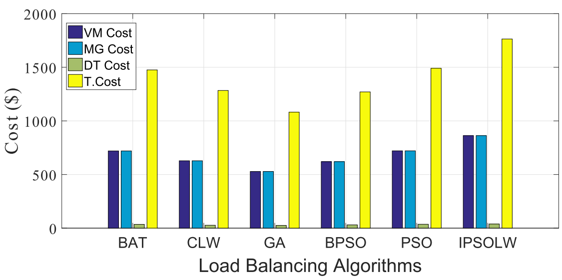
Fogs	VM Cost ($)	MG Cost ($)	DT Cost ($)	Total Cost ($)
Fog 1	540.00	540.00	33.36	1113.36
Fog 2	672.00	672.02	32.30	1376.32
Fog 3	720.00	720.01	29.36	1469.36



(a) Average cost of 50 VMs



(b) Average cost of 75 VMs



(c) Average cost of 100 VMs

FIGURE 7. Cost.

It is concluded that the PT of meta-heuristic algorithms decreases in a dynamic environment.

3) COST

The cost for the dynamic environment is calculated. The physical resources considered in this scenario is same like scenario 1. The tabular form of cost for scenario 2 in given in Table 5. In this case, the DT cost against each fog sector is different because different number of requests are assigned to the DCs. There is also a minor change between the VM and MG cost due to size and load. The Figs. [7(a)-7(c)] show the cost of load balancing algorithms against randomly generated requests.

The reason ''Cloud Analyst'' gives same results in both scenarios of cost is because (i) the cost per VM $/ Hr, (ii) the memory cost $ /sec., (iii) physical hardware units are already defined for DCs. There is a variation in DT cost and total cost because of their optimal nature.

We concluded that the proposed algorithm outperformed the counterparts because of following reasons: GA uses randomness for mutation and cross-over, which may skip the most appropriate elements. PSO, BAT and BPSO algorithms stuck in local optima. The hybrid of LW with PSO prevents premature convergence and accelerates global optimization. The simulation results also advocate the efficiency of proposed IPSOLW.

VIII. CONCLUSION AND FUTURE WORK

In this paper, a multilayer architecture for an efficient optimization of resources is presented. The proposed system model includes consumer layer, fog layer and cloud layer. The consumer layer communicates with cloud layer via the fog layer through IoT devices. Cloud then provides services to the consumers via fog layer. Actually, fog is an intermittent layer between cloud and consumer. The IPSOLW is proposed, which is the hybrid of PSO and LW. It is then compared with PSO, BAT, GA, BPSO and CLW algorithms. The RT and PT of the DCs is calculated on the basis of status relative to the average maximum and minimum consumption of the DC. The cloud layer takes the current electricity rates from the utilities and provides it to consumer layer through the fog, which helps the consumers to optimize their energy consumption. The energy is provided to the SH according to their demand. The proposed meta-heuristic algorithm outperformed the counterparts in both scenarios with NASBP. The other details related to proposed IPSOLW are discussed in section VII. The IPSOLW takes less time to transfer data from bottom to top and top to bottom layer. Overall IPSOLW performs better among all other algorithms. The results show that the delay and latency decrease, when the overall computation time of each policy is minimized.

REFERENCES

[1] J. M. Gonzalez, E. Pouresmaeil, C. A. Canizares, K. Bhattacharya, A. Mosaddegh, and B. V. Solanki, "Smart residential load simulator for energy management in smart grids," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1443–1452, Feb. 2019.

[2] V.-H. Bui, A. Hussain, and H.-M. Kim, "A multiagent-based hierarchical energy management strategy for multi-microgrids considering adjustable power and demand response," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 1323–1333, Mar. 2018.

[3] C. P. Mediwaththe, E. R. Stephens, D. B. Smith, and A. Mahanti, "Competitive energy trading framework for demand-side management in neighborhood area networks," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4313–4322, Sep. 2018.

[4] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang, and Q. Zheng, "Power-aware and performance-guaranteed virtual machine placement in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1385–1400, Jun. 2018.

[5] L. Guo, G. Hu, Y. Dong, Y. Luo, and Y. Zhu, "A game based consolidation method of virtual machines in cloud data centers with energy and load constraints," *IEEE Access*, vol. 6, pp. 4664–4676, 2018.

[6] C. Liu, K. Li, and K. Li, "A game approach to multi-servers load balancing with load-dependent server availability consideration," *IEEE Trans. Cloud Comput.*, to be published.

[7] M. H. Y. Moghaddam and A. Leon-Garcia, "A fog-based Internet of energy architecture for transactive energy management systems," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1055–1069, Apr. 2018.

[8] S. Liao, J. Li, J. Wu, W. Yang, and Z. Guan, "Fog-enabled vehicle as a service for computing geographical migration in smart cities," *IEEE Access*, vol. 7, pp. 8726–8736, 2019.

[9] L. Yu, T. Jiang, and Y. Zou, "Fog-assisted operational cost reduction for cloud data centers," *IEEE Access*, vol. 5, pp. 13578–13586, 2017.

[10] Y. Dong, S. Guo, J. Liu, and Y. Yang, "Energy-efficient fair cooperation fog computing in mobile edge networks for smart city," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7543–7554, Oct. 2019.

[11] D. Zhang, F. Haider, M. St-Hilaire, and C. Makaya, "Model and algorithms for the planning of fog computing networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3873–3884, Apr. 2019.

[12] S. K. Mishra, D. Puthal, J. J. P. C. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4497–4506, Oct. 2018.

[13] A. A. Butt, S. Khan, T. Ashfaq, S. Javaid, N. A. Sattar, and N. Javaid, "A cloud and fog based architecture for energy management of smart city by using meta-heuristic techniques," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1588–1593.

[14] X. Fu, J. Chen, S. Deng, J. Wang, and L. Zhang, "Layered virtual machine migration algorithm for network resource balancing in cloud computing," *Frontiers Comput. Sci.*, vol. 12, no. 1, pp. 75–85, 2018.

[15] Y. Guo, A. Stolyar, and A. Walid, "Online VM auto-scaling algorithms for application hosting in a cloud," *IEEE Trans. Cloud Comput.*, to be published.

[16] X. Xu, S. Fun, Q. Cai, W. Tian, W. Liu, W. Dou, X. Sun, and A. X. Liu, "Dynamic resource allocation for load balancing in fog environment," *Wireless Commun. Mobile Comput.*, vol. 2018, Apr. 2018, Art. no. 6421607.

[17] J. Luo, L. Yin, J. Hu, C. Wang, X. Liu, X. Fan, and H. Luo, "Container-based fog computing architecture and energy-balancing scheduling algorithm for energy IoT," *Future Gener. Comput. Syst.*, vol. 97, pp. 50–60, Aug. 2019.

[18] X. Sun and N. Ansari, "Latency aware workload offloading in the cloudlet network," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1481–1484, Jul. 2017.

[19] N. Verba, K.-M. Chao, J. Lewandowski, N. Shah, A. James, and F. Tian, "Modeling industry 4.0 based fog computing environments for application analysis and deployment," *Future Gener. Comput. Syst.*, vol. 91, pp. 48–60, Feb. 2019.

[20] B. Negash, T. Westerlund, and H. Tenhunen, "Towards an interoperable Internet of things through a Web of virtual things at the fog layer," *Future Gener. Comput. Syst.*, vol. 91, pp. 96–107, Feb. 2019.

[21] L. Du, L. Zhang, X. Tian, and J. Lei, "Efficient forecasting scheme and optimal delivery approach of energy for the energy Internet," *IEEE Access*, vol. 6, pp. 15026–15038, 2018.

[22] Z. Kuang, S. Guo, J. Liu, and Y. Yang, "A quick-response framework for multi-user computation offloading in mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 81, pp. 166–176, Apr. 2018.

[23] F.-H. Tseng, M.-S. Tsai, C.-W. Tseng, Y.-T. Yang, C.-C. Liu, and L.-D. Chou, "A lightweight autoscaling mechanism for fog computing in industrial applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4529–4537, Oct. 2018.

[24] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.

[25] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik, "Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4118–4149, Jun. 2019.

[26] A. J. Kadhim and S. A. H. Seno, "Maximizing the utilization of fog computing in Internet of vehicle using SDN," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 140–143, Jan. 2019.

[27] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, and C. Liu, "Fog computing for energy-aware load balancing and scheduling in smart factory," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4548–4556, Oct. 2018.

[28] Y. Xiao and M. Krunz, "Distributed optimization for energy-efficient fog computing in the tactile Internet," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2390–2400, Nov. 2018.

[29] A. N. Toosi, J. Son, and R. Buyya, "CLOUDS-Pi: A low-cost raspberry-pi based micro data center for software-defined cloud computing," *IEEE Cloud Comput.*, vol. 5, no. 5, pp. 81–91, Oct. 2018.

[30] N. Téllez, M. Jimeno, A. Salazar, and E. Nino-Ruiz, "A tabu search method for load balancing in fog computing," *Int. J. Artif. Intell. Educ.*, vol. 16, no. 2, pp. 1–31, 2018.

[31] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.-T. Lin, "Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2017.

[32] S. Zahoor, S. Javaid, N. Javaid, M. Ashraf, F. Ishmanov, and M. K. Afzal, "Cloud–fog–based smart grid model for efficient resource management," *Sustainability*, vol. 10, no. 6, p. 2079, 2018.

[33] A. Khalid, S. Aslam, K. Aurangzeb, S. Haider, M. Ashraf, and N. Javaid, "An efficient energy management approach using fog-as-a-service for sharing economy in a smart grid," *Energies*, vol. 11, no. 12, p. 3500, 2018.

[34] R. Bakhsh, N. Javaid, I. Fatima, M. I. Khan, and K. A. Almejalli, "Towards efficient resource utilization exploiting collaboration between HPF and 5G enabled energy management controllers in smart homes," *Sustainability*, vol. 10, no. 10, pp. 1–24, 2018.

[35] R. Bukhsh, N. Javaid, Z. A. Khan, F. Ishmanov, M. K. Afzal, and Z. Wadud, "Towards fast response, reduced processing and balanced load in fog-based data-driven smart grid," *Energies*, vol. 11, no. 12, pp. 1–21, 2018.

[36] R. Bakhsh, N. Javaid, M. I. Khan, Z. A. Khan, and I. Usman, "Cost efficient hybrid techniques for DSM in smart homes," *Int. J. Ad Hoc Ubiquitous Comput.*, pp. 1–13, 2018.

[37] B. Wickremasinghe and B. Rajkumar, "CloudAnalyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments," School Comput. Inf. Syst., Univ. Melbourne, Melbourne, VIC, Australia, MEDC Project Rep. 22, 2009, pp. 433–659, no. 6.

[38] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. M. Leung, "Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1688–1699, Jun. 2018.

[39] *Techopedia*. Accessed: Jan. 2019. [Online]. Available: https://www.techopedia.com/definition/17137/genetic-algorithm

[40] S. H. H. Madni, M. S. A. Latiff, J. Ali, and S. M. Abdulhamid, "Multi-objective-oriented cuckoo search optimization-based resource scheduling algorithm for clouds," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3585–3602, 2019.

[41] J. P. B. Mapetu, Z. Chen, and L. Kong, "Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing," *Appl. Intell.*, vol. 49, no. 9, pp. 3308–3330, 2019.

[42] Y. Wang, P. Wang, J. Zhang, Z. Cui, X. Cai, W. Zhang, and J. Chen, "A novel bat algorithm with multiple strategies coupling for numerical optimization," *Mathematics*, vol. 7, no. 2, p. 135, 2019.

[43] Z. Zhou, F. Liu, S. Chen, and Z. Li, "A truthful and efficient incentive mechanism for demand response in green datacenters," *IEEE Trans. Parallel Distrib. Syst.*, to be published.

[44] S. Wu, Y. Chen, X. Wang, H. Jin, F. Liu, H. Chen, and C. Yan, "Precise power capping for latency-sensitive applications in datacenter," *IEEE Trans. Sustainable Comput.*, to be published.

[45] Y. Wang, F. Zhang, C. Chi, S. Ren, F. Liu, R. Wang, and Z. Liu, "A market-oriented incentive mechanism for emergency demand response in colocation data centers," *Sustain. Comput., Inform. Syst.*, vol. 22, pp. 13–25, Jun. 2019.

[46] S. Chen, L. Jiao, L. Wang, and F. Liu, "An online market mechanism for edge emergency demand response via cloudlet control," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr./May 2019, pp. 2566–2574.

[47] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2019, pp. 1459–1467.

[48] A. Bhati, M. Hansen, and C. M. Chan, "Energy conservation through smart homes in a smart city: A lesson for Singapore households," *Energy Policy*, vol. 104, pp. 230–239, May 2017.

[49] D. Patterson and J. Hennessy, "Computer abstraction and technology," in *Computer Organization and Design: The Hardware and Software Interface*. Amsterdam, The Netherlands: Elsevier, 2013, pp. 29–31.

ZAHOOR ALI KHAN (SM'15) holds different academic positions with Dalhousie and Saint Mary's Universities, Canada. He is currently the Division Chair of the Computer Information Science (CIS) Division and the Applied Media Division, Higher Colleges of Technology, United Arab Emirates. He has more than 19 years of research and development, academia, and project management experience in IT and engineering fields. He has multidisciplinary research skills on emerging wireless technologies. His current research interests include e-health pervasive wireless applications, the theoretical and practical applications of wireless sensor networks, smart grids, and the Internet of Things. His research outcomes include several journal articles, book chapters, and numerous conference proceedings, all peer-reviewed. The journal articles have appeared in prestigious and leading journals. Most of his conference articles have been published by IEEE Xplore, Springer, or Elsevier, and indexed by Scopus and Thomson Reuters' Conference Proceeding Citation Index. He is an Editorial Board Member of several prestigious journals. He also serves as a Regular Reviewer/Organizer of numerous reputed ISI indexed journals, the IEEE conferences, and workshops. He is a Senior Member of IAENG. Several conference articles have received the Best Paper Awards (BWCCA 2012, IEEE ITT 2017, and EIDWT-2019).

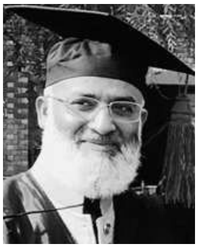
AYESHA ANJUM BUTT received the B.S. degree in computer science from UAAR-UIIT and the M.S. degree in computer science from the Department of Computer Science, COMSATS Institute of Information Technology, Islamabad, Pakistan, under the supervision of Dr. N. Javaid. Her current research interests include energy management in smart grid, cloud, and fog computing.

TURKI ALI ALGHAMDI received the B.Sc. degree in computer science from King Abdulaziz University, Jeddah, Saudi Arabia, in 2003, the M.Sc. degree in distributed systems and networks from the University of Hertfordshire, Hatfield, in 2006, and the Ph.D. degree in computer networks from the University of Bradford, Bradford, U.K., in 2010. He was the Vice Dean of Technical Affairs (IT Deanship) with Umm Al-Qura University, Makkah, Saudi Arabia, and the Dean of eLearning and IT with Taif University. He is currently an Associate Professor with the Computer Science Department, Faculty of Computer and Information Systems, Umm Al-Qura University. He holds the CDCDP and CDCMP certificates. He is passionate about developing the translational and collaborative interface between industry and academia. His research interests include wireless sensor networks, energy and QoS aware routing protocols, network security, IoT, and smart cities.

AISHA FATIMA received the M.S. degree in computer science from COMSATS University Islamabad (CUI), Islamabad, Pakistan. She is currently a Research Associate with the Communication over Sensors (ComSens) Research Group, COMSATS University Islamabad, under the supervision of Dr. N. Javaid. Her current research interests include cloud computing, wireless sensor networks, and big data.

MARIAM AKBAR received the M.Sc. degree from the Department of Physics, Quaid-i-Azam University, Islamabad, Pakistan, in 2001, the M.Phil. degree from the Department of Electronics, Quaid-i-Azam University, in 2004, and the Ph.D. degree from COMSATS University Islamabad, Islamabad, in 2016, under the supervision of Dr. N. Javaid, with the thesis entitled, On Network Lifetime Maximization in Wireless Sensor Networks With Sink Mobility. She is currently an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad. Her current research interests include underwater wireless sensor networks and energy management.

MUHAMMAD RAMZAN received the M.S. degree in computer science from the University of Management and Technology, Lahore, Pakistan, where he is currently pursuing the Ph.D. degree from the School of Systems and Technology. He was a Lecturer with the Virtual University of Pakistan. He is currently a Lecturer with the Department of Computer Science and IT, University of Sargodha, Sargodha, Pakistan. His current research interests include software engineering and digital image processing.



NADEEM JAVAID (S'08–M'11–SM'16) received the bachelor's degree in computer science and physics from Gomal University, Dera Ismail Khan, in 1995, the master's degree in electronics from Quaid-i-Azam University, Islamabad, Pakistan, in 1999, and the Ph.D. degree from the University of Paris-Est, France, in 2010. He is currently an Associate Professor and the Founding Director of the Communications over Sensors (ComSens) Research Lab, Department of Computer Science, COMSATS University Islamabad, Islamabad. He has supervised 16 Ph.D. and 112 master's theses. He has authored more than 850 articles in technical journals and international conferences. His current research interests include energy optimization in smart grid and in the IoT enabled wireless sensor networks, data science, and blockchain. He was a recipient of the Best University Teacher Award from the Higher Education Commission of Pakistan, in 2016, and the Research Productivity Award from the Pakistan Council for Science and Technology, in 2017. He is also an Associate Editor of IEEE Access and an Editor of the *International Journal of Space-Based and Situated Computing*.