Comprehensive Analysis of Resource Allocation and Service Placement in Fog and Cloud Computing

A.S. Gowri[1], P.Shanthi Bala[2], Immanuel Zion Ramdinthara[3]
Department of Computer Science, School of Engineering and Technology
Pondicherry University, India

***Abstract*—The voluminous data produced and consumed by digitalization, need resources that offer compute, storage, and communication facility. To withstand such demands, Cloud and Fog computing architectures are the viable solutions, due to their utility kind and accessibility nature. The success of any computing architecture depends on how efficiently its resources are allocated to the service requests. Among the existing survey articles on Cloud and Fog, issues like scalability and time-critical requirements of the Internet of Things (IoT) are rarely focused on. The proliferation of IoT leads to energy crises too. The proposed survey is aimed to build a Resource Allocation and Service Placement (RASP) strategy that addresses these issues. The survey recommends techniques like Reinforcement Learning (RL) and Energy Efficient Computing (EEC) in Fog and Cloud to escalate the efficacy of RASP. While RL meets the time-critical requirements of IoT with high scalability, EEC empowers RASP by saving cost and energy. As most of the early works are carried out using reactive policy, it paves the way to build RASP solutions using alternate policies. The findings of the survey help the researchers, to focus their attention on the research gaps and devise a robust RASP strategy in Fog and Cloud environment.**

Keywords*—Cloud; fog; reinforcement learning; energy-efficient computing; resource allocation; service placement

I. Introduction

Digitalization has revolutionized anything as a service (XaaS) on pay per usage basis [1]. With the increase in smart handheld devices, online business, transportation, health care, education, and food court which were once a commodity, are delivered as a service at the doorstep of the individual. These digital services produce and consume a variety of voluminous data, at a rapid speed that needs to be stored for big data analytics [2]. Consequently, enterprises depend on cloud Data Centers (DC) to store, process, and manage their data [3], [4].

A large number of commercial Cloud Service Providers (CSP) deliver compute, storage, and communication resources in the form of Infrastructure as a Service (IaaS) [5]. Estimating the required amount of IaaS resources and assigning the service (tasks) for execution is termed as Resource Allocation and Service Placement (RASP) [6][7]. Service is defined as the actual software instance that executes a task. The terms service and tasks are often used interchangeably [8].

A RASP framework abides by the Service Level Agreement (SLA) made between consumer and service provider [9][10]. SLA is the mutual agreement cum negotiation made between the service consumer and the CSP. Providing guaranteed resources to the consumers/applications on time aggravates many challenges. Inaccurate estimation of available resources, wrong forecast of workload, incorrect prediction of required resources, deadline violation, uncontrolled energy consumption, unexpected failures of hardware/software, SLA Violation (SLAV) are some of the other problems encountered by a RASP framework. Hence, a robust RASP that benefits the consumer and the service provider in terms of their requirements and revenue is needed.

Resources are allocated to the requesting services by either of the three policies viz., Reactive, Predictive, or Hybrid. In reactive policy, the initial allocation of resources is subject to change, only after the system enters an undesirable state. The reactive policy follows a predefined set of rules for scaling the resources. On the other hand, the predictive (also known as a proactive) policy, anticipates the forthcoming disruptions in advance, and updates the resources, well before the system enters the undesirable state [11]. It forecasts the workload and scales the resources in advance to meet future needs. The hybrid approach is an amalgamation of both the reactive and proactive policy [5].

Each policy bears its own cost in satisfying the SLA. The choice of policy purely depends upon the application and the RASP strategy adopted. Out of the works considered from the period 2011 to 2020, Table I shows that most of the works were carried out in the reactive policy, which opens the research gap in other policies to model RASP.

A. Significance of RASP in Cloud Computing

Cloud computing is an Information Technology service model that provides on-demand computing resources over the Internet independent of device and location [12]. The need for online services has made the enterprises move their data and applications to the DC, from where they are provisioned as services to the end-user. With the proliferation of IoT, communications, among smart devices are made possible through the cloud-assisted IoT, called a Cloud of Things (CoT) [13]. Consequently, RA in the cloud has become inevitable to serve IoT requests.

B. Significance of RASP in Fog Computing

Despite its huge processing capacity, the cloud suffers latency problems when it comes to delay-sensitive IoT applications. By the time the data are sent to the cloud for processing, the necessity to act on it might be gone, which costs lives. Hence, a computing model like Fog, which delivers services of the Cloud near the edge network is a better choice for time-sensitive applications.

TABLE I. POLICY DISTRIBUTION OF RASP WORKS

	Reactive	Proactive	Hybrid	Total
Cloud Computing	7	5		12
Reinforcement Learning	2	5	1	8
Energy-Efficient Computing	6	4	-	10
Fog Computing	17	1	-	18
Total	32	15	1	48
Percentage	67%	31%	2%	

Fog Computing (FC) is a computing paradigm where a huge number of ubiquitous, decentralized, heterogeneous, geo-distributed devices provide computation, storage, and communication facility at the edge of the local network from where the devices/objects generate and consume data [13]. It accelerates awareness cum response to events by eliminating RTT (Round Trip Time) to the cloud and avoids failures during peak period. As such, not all requests are serviced in Fog. Some of the delay-tolerant applications that involve huge computation are processed in the cloud [14] [15]. In fact, Fog complements Cloud to realize its potential with IoT applications.

C. Relevance of Reinforcement Learning (RL) in RASP

The design and implementation of RASP for the growing scale of IoT, require intelligence that is far beyond the capacity of the case-driven programming style [16]. Such programs depend on predefined rules which is hard to change instantaneously for the stochastic needs of IoT [17]. A robust RASP requires an approach like Reinforcement Learning which learns the environment (requirement and availability of resources) and maps the appropriate action on the fly.

RL is an Artificial Intelligence-based technique that automatically learns to make decisions under a dynamic environment without prior domain knowledge[18]. When service providers suffer to handle the complexity of stochastic requests in real-time, RL-assisted RASP, delivers better service in both Cloud and Fog.

D. Energy-Efficient Computing (EEC) in RASP for Green Environment

The rapid growth of DC has become the highest consumer of power that leads to the dissipation of Green House Gas (GHG) [5]. Compute and non-compute resources incur abundant energy waste [17],[19]. Measures taken to control the speed of processors, frequency/voltage, and switch-off/sleep modes, are not sufficient to reduce the effect of GHG emission [20]. Hence, an EEC-based RASP that enables sustainability of the Green Environment with minimal operational expenses is required.

An illustration of the coordinating computing models is shown in Fig. 1. It portraits the association of the Edge-Fog-Cloud computing paradigm in association with the application requests. The Fog Controller embeds the Reinforcement Learning and Energy-Efficient Computing components to achieve an efficient RASP system.

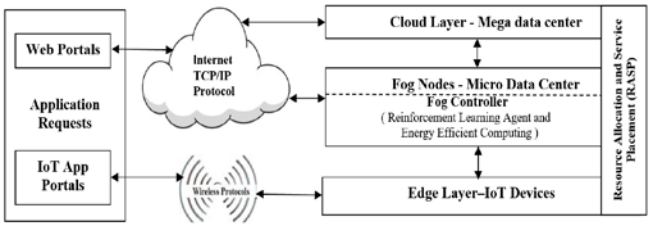

Fig. 1. Fog-Cloud Framework.

The rest of the paper is organized as follows. Section 2 reviews the existing literature works in RASP. Section 3 analyzes the RASP works in cloud datacenters. Section 4 discusses the approaches made in RASP using RL techniques while Section 5 presents EEC-based RASP. Section 6 discusses the efficacy of FC in addressing IoT applications and elaborates on the existing Fog based RASP works. Then the proposed survey concludes with a discussion on identified research gaps that could be useful to the research and development community in the future.

II. OVERVIEW OF EXISTING SURVEYS

This section analyzes the existing survey papers of RA, in Cloud Computing, RL, EEC, IoT, and Fog Computing. Resource provisioning and application management often exclude issues like unpredictable workload, poor utilization of resources, and unexpected Hardware (HW)-Software (SW) failures. The brownout paradigm that addresses such issues by enabling/disabling the optional parts of the application was presented in [21].

In [22] the author reviewed energy efficiency in four dimensions: (i) Virtual Machine (VM) placement, (ii) VM migration, (iii) Server consolidation, and (iv) Dynamic Voltage Frequency Scaling (DVFS). In [23] the author explored energy management techniques at the HW level, Resource Management (RM) level, and application level. While Static Power Management (SPM) technique was used at the HW level, Dynamic Power Management (DPM) was tackled at the RM level. Green Computing with renewable energy was recommended at the application level.

Maximization of resource utilization and minimizing the cost were the main goal of Resource Allocation (RA) in the IoT environment [24]. Scarce processing-storage capacity, low battery level, less bandwidth, and, poor implementation of resource management protocol were shortlisted as limitations of IoT. Lightweight container-based virtualization was suggested to process and store IoT applications. Though Cloud supports IoT, Fog computing resolves the time-sensitive-issues of IoT more diligently.

Application placement, resource scheduling, task offloading, and load balancing, were explored in [25]. distinguished Fog, from Multi-Access Edge Computing (MEC) and cloud, in terms of operation mode and application addressed [4]. In [15], the author identified the challenges faced by Fog computing to process context-aware applications of IoT. In [3], RA and task scheduling were considered as one of the key challenges in IoT. The survey suggested CloudSim, MATLAB, and iFogSim to implement RA in Cloud and Fog.

The author recommended container-enabled micro-services to resolve the resource limitation problem.

III. Analysis of RASP in Cloud Computing

Cloud is a ubiquitous technology that offers infrastructure, software, and platform as service on-demand with the least interaction and management effort of the service provider [26],[27]. Despite its control over the IaaS management, CSP lacks knowledge about the application hosted in their machines. VMs of different applications overlap on physical servers leading to catastrophic failure which is not recognized by the CSP instantly.

Deployment of multi-tier applications is yet another complexity, as the configuration of VMs in one tier differs from the other causing interoperability problems [28] [29]. This section analyzes the existing RASP works in Cloud. While certain works adapt their own architecture, others follow the specific algorithm for the existing RASP. Table II shows the distribution of existing RASP articles under various criteria.

A. Uncertainty in Resource Availability

Unexpected HW failures, SW faults like overflow conditions, malware, DoS (Denial of Service) attacks, and changes in the number of objectives during execution are some of the uncertain behavior projected in [30]. Power consumption cost and overestimation of resources hinder the profit of the CSP due to which certain objectives like deadline and make-span are ignored/altered while deliberating RASP. As HW/SW failure is unavoidable, the Neural Network based Dynamic Non-dominated Sorting Genetic Algorithm (NN-DNSGA-II) converges before the occurrence of the next failure. Change in the number of objectives at runtime is tackled by a generalized periodic change in the objective size.

B. Impact of SLA/QoS in RASP

The applications that are hosted in DCs expect the utmost performance in terms of low latency and high throughput within budget and specified deadline. These performance measures form the QoS requirements. The mutual negotiation between the consumer and the CSP for a guaranteed QoS results in SLA. With the growing number of IaaS providers, not only does it require expertise but is time-consuming for the clients to select an efficient CSP.

TABLE II. Classification of existing RASP papers in Cloud Computing

Paper ID	Reference	Author	Architecture based	Algorithm-based	RA Policy			Problem Addressed				
					Reactive	Proactive	Hybrid	Uncertainty	SLA/QoS	Slashdot	Elasticity	ASP viewpoint
C1	[30]	Ismayilov & Topcuoglu, 2020		✓		✓		✓				
C2	[9]	Soltani et al., 2018	✓		✓				✓			
C3	[10]	Singh & Viniotis, 2017		✓		✓			✓			
C4	[12]	Djebbar & Belalem, 2016		✓	✓				✓			
C5	[32]	Ashraf, 2016		✓		✓				✓		
C6	[29]	RahimiZadeh et al., 2015	✓		✓					✓		
C7	[28]	Kaur & Chana, 2014	✓			✓					✓	
C8	[33]	Agarwal & Jain, 2014		✓	✓					✓		
C9	[34]	Espadas et al., 2013	✓		✓							✓
C10	[35]	Casalicchio & Silvestri, 2013	✓		✓							✓
C11	[36]	Xu & Li, 2013	✓		✓							✓
C12	[31]	Islam et al., 2012	✓			✓						✓

The RA framework in [9] follows the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) in which the available IaaS resources are ranked by their similarity index concerning the application requirements. Then, the top IaaS resource was allocated to the corresponding application.

Lack of CSP's knowledge about the message arrival rate and length of the Enforcement Period (EP) were the problems encountered in satisfying SLA. To overcome the loss caused by SLAV, a RA mechanism that allows an additional EP to execute the unpredictable IoT traffic is recommended by [10]. Execution speed and deadline were considered as primary QoS constraints in [12].

C. Slashdot Prediction in RASP

Slashdot refers to the unpredictable flash crowd workload on the Internet at any instant of time [31]. A sudden traffic surge makes the RASP framework unstable. The Slashdot effect if not addressed properly, leads to a cascade of problems like unacceptable delay, long downtime, application unavailability, revenue reduction, and losing the customer in the worst case.

Conventional predictive policies turn failure as they forecast the expected workload traffic, only a few steps ahead during which the Slashdot effect remains invisible. The Long Short Term Memory Recurrent Neural Network (LSTM-RNN) technique that predicts the workload traffic/pattern a thousand steps ahead was implemented in[32]. Based on the prediction provided by LSTM-RNN, resource scaling was performed without compromising SLA.

The performance of Virtualized Multi-Tier Application (VMTA) for the unstable workload was analyzed using the queuing network in [29]. Apache, Tomcat, and MySQL servers were used for the front end, application, and database tiers, respectively [33]. A Generalized Priority Algorithm (GPA) for scheduling tasks in the cloud, consumed the least execution time when compared to the First Come First Serve method.

D. Need for Elasticity in RASP

Resource elasticity refers to the automatic acquisition and release of resources at runtime to fulfill the QoS requirements in response to the changing workload. Though the workload traffic is predicted in advance, RA without an elasticity component is a failure, as neither the resources are efficiently scaled nor is the QoS met [34]. The QoS aware resource elasticity framework for multi-tier application was modeled in [28]. The framework employed MT-PerfMod (Multitier Performance Module) to compute the overall response time and resource utilization, based on which, the MT-ResElas (Multi-Tier Resource Elasticity) module computed the SLAV rate. Whenever the response time and the resource utilization rate were violated, VMs were increased; otherwise, the number of VMs was reduced by half.

E. RASP on ASP (Application Service Provider) Point of View

The majority of RA is performed from the CSP point of view, which reduces the preference for ASP. The ASP is charged for the resources that were wasted due to underutilization. Hence, an ASP (tenant) centric RA for scaling the application was modeled in [31][34]. The knapsack problem approach was implemented to predict the minimum number of VMs required.

Though the maximum number of VMs required was estimated in advance, it keeps changing depending upon the number of active users who access the application. The problem arises when the ASP (consumer) is charged for the idle resources. An SLA-based RP mechanism in the ASP point of view was presented in [35]. A framework where clients and operators suggest their preference for RA policies was presented in [36]. The technique described the allocation of jobs to a machine, based on the stable matching algorithm. Tables IIIA and IIIB tabulate the observations of the RASP works in Cloud Computing.

TABLE III. A. ANALYSIS OF RASP WORKS IN CLOUD COMPUTING

Paper ID	Ref.	Problem addressed	Objective	Algorithm/Approach	Performance metrics addressed
C1	[30]	Unexpected Hardware-Software failure and change in the number of objectives at runtime	Formulate a scheduling strategy to minimize cost, energy and maximize resource utility for periodical workflow	Neural network-based dynamic non-dominating sorting genetic algorithm (NN-DNSGA-II)	Cost, energy, and resource utility through Non-dominated solutions (NS), Schott's spacing (SS), and Hyper Volume (HV)
C2	[9]	Time and cost difficulties in cloud service selection	To build an automatic cloud service selection framework that overcomes time and cost problem	Architecture based- Hybridization of case-based reasoning with Multi-criteria decision making (MCDM) and TOPSIS (Technique for order of preferences by Similarity to Ideal Solutions)	Recommended CSP, CSP's service type, memory storage, region, Price/Hr., OS
C3	[10]	Enforcement of IoT SLA in the cloud environment	Conformance of SLA within enforcement period	Server over-provisioning approach, policing, Weighted Round Robin (WRR) scheduling algorithm, rate-limiting mechanism to enforce SLA	Number of messages arrived/processed, SLA confirmation rate, number of servers, additional enforcement period used
C4	[12]	High data management in scientific application	Minimize response time	Space and Time-shared policy based on deadline, length of the task, the execution speed of VM, and VM tree method.	Total response time
C5	[32]	Inaccuracy in the prediction of workload violates SLA and increases the cost	Prediction of resource demand and auto-scale them instantaneously that minimizes cost irrespective of application traffic	Long short-term memory RNN with peephole connections with Mean Absolute Deviation (MAD) to set threshold	Response time, No. of VMs, No. of completed request with the deadline.
C6	[29]	Stochastic burst and non-burst workload	Propose an analytical model-based queuing network to estimate aggregated QoS metrics	Analytical model-based queuing network (M/G/1)	Response time, disk utilization, CPU utilization.
C7	[28]	The contradiction between QoS and elasticity of resources	Mapping of the QoS attribute with minimum SLA violations thus maximizing the overall profit	Architecture-based - QoS aware resource Elasticity framework for the multi-tier web application. Control Theoretic based scaling algorithm	Response time < 5 secs, Resource utilization > 80%
C8	[33]	Task scheduling	Minimize execution time	Generalized Priority algorithm (GPA) based on highest length cloudlet to highest MIPS VMs	Execution time
C9	[34]	To solve under-utilization and over utilization of resources in cloud applications	tenant-based isolation, tenant-based load balancing, tenant-based VM allocation	Architecture based	CPU Utilization, memory utilization, Throughput
C10	[35]	SLA based resource provisioning in cloud	Achieve SLA oriented resource provision irrespective of workload type	Queuing model M/G/1 and M/M/m with autonomic QoS aware resource provisioning	CPU utilization, response time, number of VMs required
C11	[36]	Tasks to occupy a minimum number of VMs to achieve server consolidation	Develop a unified framework for resource management in the cloud, where policies are decoupled.	Conventional Job-Machine stable matching problem	Execution time, no of VMs
C12	[31]	Resource Prediction and Provisioning	Build an adaptive RM for applications hosted in the cloud.	Neural Network and Linear Regression to satisfy upcoming demands	CPU Utilization for each technique

B. ANALYSIS OF RASP WORKS IN CLOUD COMPUTING

Paper ID	Ref.	Experiment	Evaluation	Workload	Limitations
C1	[30]	Real-time experiment with Amazon EC2	Evaluated with DNSGA-RI, DMOPSO, DNSGA-II-HM, DNSGA II-A, and DNSGA-II-B	100 to 1000 tasks from Pegasus workflow management that covers astronomy, physics, biology, geology, and bio-informatics dataset.	The work is compared with non-predictive algorithms.
C2	[9]	Test bed	Validated with a sample application that is to be deployed on one of the US regions	Service template of a sample application	Criteria for CSP selection, resource provision, task scheduling are problem-specific
C3	[10]	Discrete event simulator in C	Evaluated for a different rate of traffic request, change in capacity, enforcement period	Two million messages per tenant per month	Homogenous message size limited to 512 bytes
C4	[12]	CloudSim	Compared with time/space shared policy.	Simulated with 10-50 cloudlets (tasks)	The reactive policy cannot scale and tolerate dynamism
C5	[32]	CloudSim using deeplearnig4j open source	Compared with automatic scaling and conventional threshold-based scaling techniques.	NASA Clark net workload	Explanation required for computations of response time, number of the completed request.
C6	[29]	Test bed constructed with 2 servers, 6-VM/server	Evaluate the performance of VMTA (virtualized multi-tier applications) through cache hit ratio, request arrival rate.	Rubis and Wikipedia tiers under burst & non-burst workloads.	The trade-off between assignments of cores to domains, cache contention can be investigated.
C7	[28]	Amazon cloud watch (EC2 monitoring tool)	JMeter load tests-to measure response time & utilization, Amazon cloud watch - % of utilization	3-tier web applications	QRE (QoS aware Resource Elasticity) framework is considered a homogenous type of VMs only. Resource availability, fault tolerance can be measured.
C8	[33]	Cloud Sim	Compared with first come first serve, round-robin	web service generated workload traces	Cannot handle instantaneous demand of resources, leads to over-provisioning or under-provisioning.
C9	[34]	Test bed: eucalyptus cloud, Tomcat-based SaaS platform deployed over it.	t-test statistical analysis	Apache JMeter to create web service workloads to the Tomcat cluster	HPC and Online transactions, bandwidth, storage, and transfer data, need to experiment
C10	[35]	Amazon cloud watch (EC2 monitoring tool) with Mat lab graph generation	Partial ASP and limited ASP (Application service provider)	Wikibench- to generate workload from Wikipedia, Mediawiki for backend database	The reactive approach cannot address stochastic heterogeneous workload type
C11	[36]	1) Test bed-prototype implementation with a cluster of 20 dual-core machines and 2) Trace-driven simulation.	Correctness convergence, job-optimality of multistage deferred acceptance are proved through theorems & lemma	RICC (RIKEN Integrated Cluster of Clusters), explored for 200 tasks with 1000 VMs	VM migration can be included
C12	[31]	Amazon EC2 instances	Evaluated with MAPE (Mean absolute Percentage), PRED (25) (Prediction accuracy within 25%), RMSE (Root Mean Square Error)	TPC-W - interactive E-commerce application	Integration of prediction strategies with auto-scaling can enhance the effectiveness of the adaptive resource allocation in terms of performance and cost.

IV. Analysis of Reinforcement Learning Assisted RASP

The human-to-machine and machine-to-machine interaction-based IoT applications demand a technique that makes the optimal decision at high speed. The traditional rule-based programming approach does not withstand the stochastic requirements of IoT. Hence, a machine learning programming approach that observes and adapts to the environment is required. Such requirement leads to the choice of Reinforcement Learning (RL) which automatically learns to take decisions by trial and error method under a dynamic environment with prior domain knowledge. Fig. 2 depicts the basic structure of RL.

In RL based RASP, service request and the resource pool forms the environment. The values like the expected number of service requests and the amount of available resource observed at any instant of time form the state. At every time-step of interaction, the state values form the input to the agent from the environment. Action is the decision taken to place the service request in the appropriate resource. The agent chooses its action in such a way that the system achieves maximum resource utilization with minimum cost. For every action taken, the agent receives a suitable positive or negative reward as an incentive. By trial and error, the agent tries to maximize its reward by taking optimal decisions (actions) in the long run.

The agent is trained to take optimal action through either of the RL algorithms like Q-learning, SARSA, E-SARSA, or Deep RL. The choice of the RL algorithm depends on the type of problem encountered and the feasibility of implementation. This section analyzes the RL-assisted RASP works for the categories given in Table I.

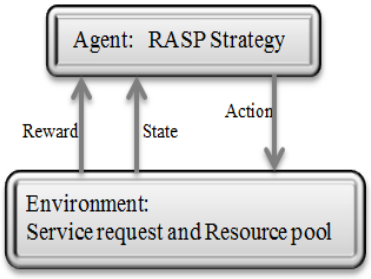

Fig. 2. RL Assisted RASP.

TABLE IV. Classification of RL Assisted RASP Works

Paper ID	Reference	Author	Architecture based	Algorithm-based	RA Policy			Problem Addressed			
					Reactive	Proactive	Hybrid	Fog-RAN	Job rejection & client retention	QoE	Auto Reconfiguration
R1	[37]	Nassar & Yilmaz, 2019		✓	✓			✓			
R2	[40]	Gai & Qiu, 2018	✓			✓				✓	
R3	[6]	Cheng et al., 2018	✓			✓			✓		
R4	[38]	Bahrpeyma et al., 2015	✓			✓			✓		
R5	[41]	Xiangping Bu et al., 2013	✓			✓					✓
R6	[42]	Xu et al., 2012	✓				✓				✓
R7	[43]	Dutreilh et al., 2011		✓	✓						✓
R8	[26]	Rao et al., 2011		✓		✓					✓

TABLE V. ANALYSIS OF REINFORCEMENT LEARNING ASSISTED RASP WORKS

Paper ID	Ref.	Problem addressed	Objective	Algorithm/Approach	Performance metrics addressed	Experiment	Evaluation	Workload	Limitations
R1	[37]	Intolerable delay due to sequential allocation of Fog resources for IoT	To achieve ultra-reliable low latency communication using Fog-RAN	RL methods - QL, SARSA, E-SARSA, MC	Resource utility and idle time	Test bed	RL based methods compared with a fixed threshold algorithm	19 scenarios of IoT applications	Sequential request arrival pattern alone considered. Effective only for static IoT environment.
R2	[40]	The contradiction between performance and time in providing QoE for IoT requests	To derive an optimal resource allocation strategy to achieve QoE	RL based mapping table algorithm and dynamic programming based RL enabled RA algorithm	QoE in terms of latency and energy	Java programming based simulated experiment	Performance time compared for varying number of tasks and compute nodes	Four data blocks	The number of IoT tasks assumed to be less than available compute nodes.
R3	[6]	Frequent change of QoS request pattern by clients	To allocate resources with minimum run-time, energy cost, and job rejection rate	Semi Markov Decision Process (SMDP) and Deep RL to solve the problem	Energy cost, run-time, job rejection rate.	Simulation experiments	With Greedy, Round Robin& FERPTS (Fast energy-aware resource provision & task scheduling)	200,000 user requests in 5000 servers in 10-100 clusters with the real workload from Google traces for 29 days.	Evaluated for batch application requests only. Real-time applications need to be considered.
R4	[38]	Client retention problem	Derive optimal policy that prevents job rejection and minimize energy consumption	Q-learning to deal with the uncertainty of problems, Neural NW-to predict future demand of a resource, Genetic Algorithm for the action selection mechanism	Job rejection rate (0%), wastage in resource usage (9.55%)	MATLAB tool box with 23 neurons in the hidden layer and Gaussian kernel function.	NN prediction for VM demand evaluated through normalized MSE (mean square error)	90 days' workload trace.	Cost overhead on different techniques. Response time and energy are not explained well.
R5	[41]	To overcome performance degradation of IoT applications	enable coordinated configuration of VMs and their hosted application dynamically	Hybridization of simplex and reinforcement learning	Throughput, response time	Test bed with 16 physical servers each with 100 VMs	Validated with 720 iterations, compared with Nelder Mead, Hill climbing strategy, ARMA controller strategy	Xen based virtualized environment with TPC-W & TPC-C benchmarks consisting of about 5000 clients	The model-free approach takes time to optimize configuration. Suitable only for applications that could be stable for a long time.
R6	[42]	Real-time auto reconfiguration of VM and the applications that run on it	To achieve SLA optimization on VM and application-specific parameter	RL algorithm with multilayer feed-forward back propagation neural network and polynomial regression to predict application parameter for reconfiguration	Response Time, Throughput, and resource utilization	Test bed with Xen virtualization platform	Tested homogenous &heterogeneous applications indifferent physical server	TPC-C, TPC-W, SPEC web workloads	RL suffers from scalability in model-based approach when there are insufficient prior cases
R7	[43]	Dynamic adaptation of resources	To achieve, 1). Effective allocation policy from the start of RL 2) prompt convergence to the optimal policy	Reinforcement learning-based autonomic resource allocation	Decision on convergence speed up measured	Test bed in Amazon EC2	With and without convergence speed-up applied for every 5000 observations	Ohio - a standard test bed for web services	To be tested for large scale application
R8	[26]	Resource provisioning	Minimize response time and number of VMs utilized	Architecture based iBalloon framework with the RL algorithm	Throughput, response time, number of VMs	Tested on2 clusters with16 & 22 machines each with 8 to 12cores	Compared with ARMA, Optimal strategy, Adaptive PI (Proportional Integral)	SPEC-Web (e-commerce), TPC-W (CPU intensive)	A discrete set of actions alone is considered. RL capacity management suffers poor initial

A. RL based RASP for F-RAN (Fog-Radio Access Network)

Fifth-generation wireless communication is an emerging solution to the expectations of ultra-low latency, minimized energy consumption, and high throughput [37]. Cloud-based Radio Access Network (C-RAN) used base stations, remote radio heads as resources to process IoT applications. But, the unlimited IoT traffic imposes a heavy burden, turning the C-RAN less efficient for IoT applications. Employing RL assisted Fog nodes in the front-haul alleviated the cloud's burden, and elevated Fog-RAN (F-RAN) as a promising solution to tackle time-critical applications of IoT [39]. RL-enabled RASP in F-RAN has the advantage of local processing and distributed storage capability at the vicinity of the end-user resulting in high resource utilization [37].

B. Job Rejection Rate and Customer Retention in RL based RASP

Enterprises look for CSPs to host their applications for online business. A CSP is chosen based on the service quality they provide. But, in the CSP viewpoint, a job is rejected under certain circumstances: (i) If the job cannot be completed within the deadline even after using a large number of resources, (ii) if the estimated resource capacity is greater than the available resource capacity (iii) frequent change of requirements from the client-side. The increase in DCs has driven competition among the CSPs to attract and retain customers. Though the CSPs advertise a low price, consumers do not prefer them due to the diminished QoS they offer. [38] Hence, to avoid customer loss, CSPs adopt an optimal resource provisioning policy like RL-DRP (Reinforcement Learning based Dynamic Resource Provision).

C. Quality of Experience (QoE) in RL based RASP

In [40], the author addressed the issues of RA and achieved QoE through Smart Content-Centric Services for IoT applications (SCCS-IoT). The algorithm employs RL based Mapping Table (RLMT) to update/maintain the cost mapping table. Each IoT task is an n-tuple to represent m number of costs (energy, latency, bandwidth, execution time). The allocation path and the quality level represented the state of the environment. Each update that was carried out on the table represented the action. The sequence of costs formed the feedback. The updated cost mapping table forms the input to the second algorithm called, RL-based RA (RLRA) that generated a policy to obtain an optimal RA for the incoming tasks.

D. Auto Reconfiguration of VMs in RL Assisted RASP

Large-scale application deployment demands adaptive techniques like RL-based RASP that dynamically configure/reconfigure the VMs and the application requirements, as needed. The RL-based framework called CoTuner synchronizes the configuration of VMs and the applications hosted in it [41]. VMs and applications in the cloud were auto-reconfigured at an optimal range to improve the resource utility and application performance in [42]. Dynamic resource configuration through RL was suggested in [43]. The delayed learning process of RL was overcome by a value-function that converged the optimal learning policy at a fast rate.

A self-adaptive learning agent called iBalloon handled the dynamic capacity management of each VM in [26]. iBalloon was based on RL in which utilization of the CPU, memory, and I/O are considered as the state of the environment. The action to be taken was of the form (no-operation, scale-up, scale-down) on the VM's resources. The Decision Maker (DM) module computed the required resource capacity. The Host Agent module monitored and reconfigured VM's resources. Any deviation from the SLA was reported back to the DM that updated the capacity management. The observations of the existing works on RL-assisted RASP are tabulated in Table V.

V. Analysis of Energy Efficient Computing (EEC) Assisted RASP

With the proliferation of DCs, the CAGR (Cumulative Annual Growth Rate) of carbon emission is expected to cross 11% worldwide, which is a serious threat to be handled immediately [5]. Hence, an EEC-based RASP that minimizes energy consumption and carbon emission is required [44]. The EEC-assisted RASP is classified as thermal aware and power-aware energy management as shown in Fig. 3. In general, thermal aware energy depends on the number of resources involved rather than the temperature density of those resources. As power is directly proportionate to the temperature density of the resources, the proposed survey focuses on power-aware energy management [23].

Energy management through Load balancing tackles the overload and underload aspects of resources, only after the tasks are scheduled. Whereas, RA approach handles energy management by predicting the power consumption in advance and optimizes the resource utilization [22]. This section discusses the works related to EEC-assisted RASP under various criteria as shown in Table VI.

A. Minimization of Energy Cost and Latency

Energy consumption and latency reduction in Fog computing were implemented in[16]. In the health care case study, the Medium Access Control (MAC) scheduler allocated the available time slots in Time Slotted Channel Hopping (TSCH) frame to the requesting sensors, by an equally spaced method. Cloudlet (an interface node between the mobile device and cloud server) assisted with Dynamic Energy Cost Minimization (DECM) technique was adopted to reduce the energy cost in [19]. Whenever applications are invoked through mobile, the DECM finds the cloudlets that reside near to the CSP. Then, the mobile request is forwarded to the recommended cloudlet.

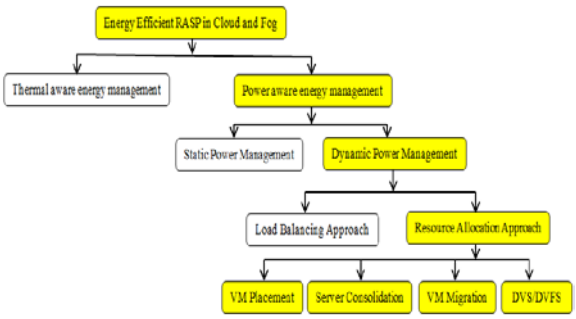

Fig. 3. Taxonomy of EEC.

TABLE VI. EEC ASSISTED RASP WORKS

Paper ID	Reference	Author	Architecture based	Algorithm-based	RA Policy			Problem Addressed			
					Reactive	Proactive	Hybrid	Energy cost and Latency Minimization	DCiE & PUE	VM Migration / Server Consolidation	DVS / DVFS
E1	[16]	La et al., 2019	✓		✓			✓			
E2	[18]	Thein et al., 2018	✓		✓				✓		
E3	[17]	Duan et al., 2017	✓			✓				✓	
E4	[45]	Shelar et al., 2017		✓	✓					✓	
E5	[19]	Gai et al., 2016		✓		✓		✓			
E6	[49]	Wu et al., 2014	✓		✓						✓
E7	[48]	Fargo et al., 2014	✓			✓					✓
E8	[20]	Basmadjian et al., 2012		✓	✓						✓
E9	[44]	Beloglazov et al., 2012		✓	✓					✓	
E10	[49]	Zhang et al., 2012	✓			✓					✓

B. Energy Conservation through PUE and DCiE

Power Usage Effectiveness (PUE) and DC infrastructure Efficiency (DCiE) were referred to in the RA framework to compute the power consumption of a DC in [18]. PUE is the ratio of the power consumed by IT equipment to the power consumed by the total IT facility. But, DCiE is inversely proportional to PUE [23]. The framework senses the state (number of physical hosts) of the DC and takes actions (allocate or not allocate), respectively.

C. Energy Conservation based on VM Placement, Migration, and Server Consolidation

VM migration is the process of transferring the process of the selected VMs from one host to another, to avoid overutilization or under-utilization issues [22]. VM migration enables server consolidation by utilizing only the optimal number of servers thereby shutting down the unused servers in [17]. To reduce energy consumption, the Modified Best Fit Decreasing (MBFD) algorithm[44], arranged VMs in decreasing order of CPU utilization and allocated them to the highest power-efficient host. The algorithm aiCloud optimized the total power consumption, by switching the idle and underutilized physical machines to a power-saving state or offline state (hibernate/ sleep/standby) in [45].

D. DVS/DVFS based Energy-Efficient Computing

The growth in the number of DCs has become a huge consumer of power. Scaling down the frequency/voltage, at the level of processor, memory, HDD, and NIC were the techniques employed to save power consumption in general. DVFS scaling that controls the frequency and voltage to maintain optimal performance was employed in [46]. The architecture specified the minimum and maximum frequency to run a job as one of the requirements, based on which the DVFS was programmed.

An Autonomic Workload and Resource Management framework (AWRM) that reduced power consumption by predicting the workload was employed in [47]. [20] [48] presented an energy-saving and carbon footprint reduction model where the processor frequency was reduced instantly, once it turned idle. The author proved that with the right combination of optimization policy and power prediction model, energy consumption was reduced by 20%. Table VII, tabulates the observations of the EEC-assisted RASP works.

TABLE VII. ANALYSIS OF ENERGY EFFICIENT COMPUTING ASSISTED RASP WORKS

Paper ID	Ref.	Problem addressed	Objective	Algorithm	Performance metrics addressed	Experiment	Evaluation	Workload	Limitations
E1	[16]	1. Time slot allocation to the sensor for transmission, 2. Device driven task offloading	To achieve energy efficiency and latency reduction in time-critical IoT applications	Mixed-integer programming based Semi-definite relaxation algorithm	Delay, power consumption, Radio duty cycle, Packet delivery ratio	Test bed-Open Mote CC2538 as fog node & border router with Raspberri-pi gateway	Compared with local processing, random assignment, all to cloud policy.	Data from ECG, Accelometer, Temperature and humidity reading	2 case studies-applications are limited to a single user.
E2	[18]	Inefficient resource allocation consuming more energy	To achieve energy efficiency and to avoid SLAV	RL + Fuzzy logic approach	DCiE, PUE, CPU utilization, SLAV	CloudSim	experimented on the traces of the planet lab virtualized environment	Planet lab virtualized research data sets of 10	CPU alone considered for energy
E3	[17]	Non-adaptability of scheduling algorithm to peak demands instantaneously	Reduce energy consumption with high computation capacity	Fractal Mathematics (FM), Improved Ant Colony algorithm (IAC).	Energy consumption, CPU utilization, VM migration count	CloudSim (FM for prediction model, IAC to optimize energy consumption)	Compared with the first fit, round-robin, minimum migration. FM calculated by Pearson correlation coefficient.	Web services from Google cluster.	SLA violation to be addressed.
E4	[45]	Power conservation and reduction of VM placement failure rate.	Minimize frequency of VM migration, minimize active servers in DCs	aiCloud algorithm - switches idle PMs to power-saving state(sleep/standby)	Power consumption in DC, VM placement failure rate.	Test bed	Compared with FF, Best Fit (BF), and random selection.	Web service (memory and CPU intensive applications)	SLA/QoS not considered.
E5	[19]	Insufficient bandwidth and device capacity of mobile phones in accessing cloud	Minimize cost and make-span with efficient resource utilization	Dynamic programming approach	Energy and Latency	Dynamic energy cost minimization Simulator (DECM)	Mathematical comparison of DECM with traditional cloud applications	Simulated requests workload from mobile phones	Cloudlets are assumed static.
E6	[47]	Energy consumption, SLA violation	Minimize power consumption and execution time	Priority Job Scheduling algorithm.	Execution time(secs), power (watts)	CloudSim	Compared with the applicable machine and HPC machine (Max/Min scheduling MMs - DVFS)	Simulated with low workload (8000-44000 MIPS), high workload (250000-450000MIPS)	Only 3 VMs considered for Power consumption
E7	[48]	Power management methodology without compromising SLA requirements.	To invoke exactly the required number of VMs and minimize power consumption	Rule-based data mining algorithm Jrip (Weka implementation of RIPPER algorithm)	Power consumption, Execution time.	Test bed configured with IBM blade server, Debian, Xen hypervisor.	Execution time & power consumption compared between static resource allocation and frequency schedule	Rubis benchmark (an auction model - emulating eBay transactions)	Correct VM configuration to be analyzed in the training phase.
E8	[20]	Energy conservation	Incentives for saving energy	Power consumption prediction model	Energy	Test bed	iLo (integrated Lights out), PCM(Power consumption model)	Steady task - constant resource, Spiky task-sudden increase of resource, Rippling task-mixed	Carbon emission to be considered
E9	[44]	Operational cost due to electricity consumption and huge carbon emission	Energy-efficient RA satisfying QoS and power usage.	Modified best fit decreasing and Minimization of migration algorithm	Power consumption, average SLA violation, No. of VM migration	CloudSim tool kit	Compared with policies, Non-power aware (NPA), DVFS, ST	Modeled web application of variable workload	Power consumption of non-compute devices missed.
E10	[49]	Virtualization challenges power consumption	To devise a power management model that reconfigures resources based on their utilization, workload, and power consumption.	Reinforcement learning algorithm	Power(watts), execution time.	Manual test bed consisting of 64 servers. Wattsup Pro digital -power meter to collect power consumption.	Compared with ARMA (Auto Regression Moving Average) and CAPM (cloud adaptive power management), Validated with NPB, IoZone	NPB – CPU/memory-intensive application, IoZone - I/O intensive	SLA to be considered

VI. Analysis of RASP in Fog Computing

The term Fog computing was proposed by cisco systems in 2012. Cisco defines Fog as a computing architecture that extends the capabilities of the cloud closer to the things that produce and act on data. The IoT devices that produce and consume data are located in the edge network. Fog computing resides as a middle layer between the edge network and the cloud as shown in Fig. 1. The proximity of fog nodes near the edge network guarantees minimum bandwidth and latency for time-critical applications. A well-defined RASP strategy in Fog layer helps IoT realize its potential. The Fog computing-based RASP works considered for the survey are categorized in Table VIII.

A. Profit-Cost Oriented QoS in RASP

The profit-centric service provider saved their cost by employing an optimized RA model that guaranteed less response time in [8]. An empirical approach that maximized Fog utilization and minimized cost was presented in [49]. A RA strategy that maximized the profit of both the resource provider and consumer was suggested in [50]. The contradiction between price and time in completing a task was resolved through Priced Timed Petri Nets (PTPN) in which the required resources were chosen from a group of pre-allocated resources.

Besides other requirements, the cost is a significant QoS metric for both the service provider and the user [51]. A Cost aware Fog RA for the medical cyber-physical system was presented in [51]. While the base transceiver station was employed as a fog node, the data transmission rate, delay, and service rate were the QoS metrics used to compute the total cost in allocating the resource.

B. RASP based on Resource Utilization Oriented QoS

Resource utilization is the allocation of available resources among the competing tasks within the budget as specified in the QoS. The price of a resource depends upon whether it is over-demanded or under-demanded. A market equilibrium framework that balanced the interests of both the service (buyer) and the Fog resource (goods) was employed in [52].

A two-sided matching game problem that stabilized the association of Fog and IoT to maximize resource utilization was presented in [53]. The higher resource utilization rate indicated its optimal consumption which in turn reduces the carbon emission. A proximal algorithm that assured utility-oriented RA and reduced carbon disposal was suggested in [54].

C. RASP based on Quality of Experience (QoE)

Quality of Experience (QoE) is the key factor to evaluate the service satisfaction of the end-user. QoE varies with the expectation of the end-user. While certain consumers are satisfied with minimal latency and bandwidth, others prefer saving the cost. An efficient RASP strategy that enhanced the QoE of mobile users was described in [55]. A RA model that enhanced the QoE of IoT users in terms of cost reduction through the game theory approach was implemented in [56].

D. RASP based on Bandwidth Oriented QoS

The geographical distance and insufficient bandwidth issues of the Cloud were overcome by the Fog enabled Cloud architecture called ROUTER (ResOUrce management TEchnique for smaRt homes) [57]. ROUTER ensured minimum bandwidth and response time through the Particle Swarm Optimization algorithm. The algorithm found the best resource for a job (particle) through fitness value (sum of weighted values of required energy, bandwidth, latency, and response time).

Bandwidth aware Component Deployment Problem (CDP) was presented in [58]. The backtrack search algorithm picked a compatible Fog node to deploy a component (IoT request). The compatibility was verified in terms of the HW-SW requirement, communication link, and bandwidth capacity. When the requirement matched, the component was deployed in the Fog node, otherwise, the search was repeated to find a compatible Fog node. The author implemented a preprocessing procedure to reduce the search time of the Fog node.

E. QoS of Latency, Round Trip Time (RTT), Delay and Response Time

As far as Industrial IoT is concerned, a minimal delay is the most expected QoS metric. Even Fog suffers the delay caused by the VM boot time. Hence, virtual containers that consumed less memory and instantiation time was suggested as Fog resource in [7]. The Gaussian Process Regression for Fog-Cloud Allocation (GPRFCA) was employed to decide, whether a request is to be processed in Fog or Cloud in [59]. A QoS-aware Fog Service Placement Problem (FSPP) that reduced execution cost and response time was recommended by [60].

F. Fog Radio Access Network (Fog-RAN)

The scarcity of Fog resources was overcome by employing the fronthaul devices of the cellular network as fog devices in [39]. A loosely coupled architecture for emerging 5G networks of Fog-RAN was recommended by [39]. The architecture encouraged the participation of more Fog nodes to lessen the burden of the fronthaul on cellular networks.

A RA scheme with the radio spectrum and Fog nodes as the resource was implemented through the student project matching algorithm in [61]. The service provider maintained the list of radio spectrum and Fog resource pair to which the request was matched as per the preference of the users. The base transceiver stations, Wi-Fi access points, and femtocell routers upgraded with CPU and memory capacity served as Fog nodes to deliver ultra-high-speed latency for IoT applications in [61].

TABLE VIII. CLASSIFICATION OF RASP PAPERS IN FOG COMPUTING

Paper-ID	Reference	Author	Architecture based	Algorithm-based	Policy			Problem Addressed						
					Reactive	Proactive	Hybrid	Cost aware	Resource Utilization	QoE	Bandwidth	Latency	Fog-RAN	QoS-SLA
F1	[8]	Tran et al., 2019	✓		✓			✓						
F2	[52]	Nguyen et al., 2019		✓	✓				✓					
F3	[55]	Kim, 2019		✓	✓					✓				
F4	[57]	Gill et al., 2019	✓		✓						✓			
F5	[53]	Abedin et al., 2019	✓		✓				✓					
F6	[56]	Shah-Mansouri & Wong, 2018		✓	✓					✓				
F7	[59]	da Silva & Fonseca, 2018		✓		✓						✓		
F8	[7]	Yin et al., 2018		✓	✓							✓		
F9	[61]	Y. Gu et al., 2018		✓	✓								✓	
F10	[39]	Rahman et al., 2018		✓	✓								✓	
F11	[60]	Skarlat et al., 2017	✓		✓							✓		
F12	[49]	Mulla et al., 2017	✓		✓			✓						
F13	[58]	Brogi & Forti, 2017	✓		✓						✓			
F14	[62]	Sun & Zhang, 2017	✓		✓									✓
F15	[50]	Ni et al., 2017		✓	✓			✓						
F16	[51]	L.Gu et al., 2017		✓	✓			✓						
F17	[63]	Alsaffar et al., 2016	✓		✓									✓
F18	[54]	Do et al., 2015	✓		✓				✓					

G. QoS-SLA based RASP in Fog Computing

With scalability being a challenge to Fog, the author suggested sharing computing resources from mobile users as Fog nodes in [62]. Incentives were provided to the mobile owners who contribute to the resource pool. A Fog-Cloud federated IoT RASP architecture that optimized resource utilization and data distribution was presented in [63]. Table IXA and IXB tabulate the analysis of Fog based RASP works

VII. DISCUSSION AND CONCLUSION

A. Identified Research Gaps and Future Enhancements

The survey explores different strategies to solve the RASP problem under various domains viz., Cloud, Fog, RL, and EEC. In the effort to solve the RASP problem arises many sub problems. Resource scalability, over-provision/under-provision of resources, violation of cost, budget, and time constraints are some of the subproblems that need to be addressed while implementing an effective RASP system. Especially, in the case of IoT applications where the requirements are stochastic and delay-sensitive.

Most of the RASP works were carried out using reactive policy. Though reactive policy incurs less cost, its case-driven programming approach does not withstand the time-sensitive requirements of IoT applications. Hence, adapting machine learning-based proactive and hybrid policies gives an effective.

TABLE IX. A. ANALYSIS OF RASP WORKS IN FOG COMPUTING

Paper ID	Ref.	Problem addressed	Objective	Algorithm	Performance metrics addressed
F1	[8]	Optimization of IoT task placement on fog	Maximize task deployment in fog & minimize response time, energy consumption, and operational cost	Empirical approach	Latency, energy, network load, operational cost.
F2	[52]	Allocation of capacity limited fog nodes to competing requests with diverse preferences.	Maximize resource utilization of fog under budget constraint	Market equilibrium (ME) solution with service requests as buyers and fog resources as goods.	Resource utilization.
F3	[55]	Inefficient coordination among mobile devices and Fog Controller in allocating resources	Maximize QoE and resource utilization, minimize task failure rate.	2 phase Gaussian model-based BVG and NBS resource allocation	Task failure probability, QoE, resource utilization at Fog Access Point (FAP)
F4	[57]	Response time issue in Fog-Cloud federated resource allocation for smart home	Optimize performance parameters through a fitness function	Particle Swarm Optimization algorithm	Response time, Bandwidth, latency, energy consumption
F5	[53]	Limited bandwidth in fog network resource allocation	Maximize fog network resource utilization for IoT applications	Analytics hierarchy process (AHP) based QoS prioritization through two-sided matching game best fit	Resource utilization, throughput, bandwidth, efficiency, job-delay
F6	[56]	Pure Nash Equilibrium problem in RA for IoT applications	Maximize QoE, minimize energy and delay	Near-optimal RA algorithm to tackle Pure Nash Equilibrium	Computation delay, average QoE, Number of IoT users benefited
F7	[59]	IoT service placement in Fog/cloud	Minimize energy consumption, request blocking, and latency.	Gaussian process regression fog-cloud allocation (GPRFCA).	Energy consumption, request block ratio, and latency.
F8	[7]	Delay due to limited resource capacity of fog in real-time analysis of smart manufacturing	Maximize Fog utilization, minimize task delay	A heuristic algorithm-based fixed threshold (FT), dynamic threshold (DT) with fixed and reallocation quota.	Number of accepted tasks, delay, execution time
F9	[61]	Instability in the allocation of channel bandwidth and computational resource for IoT in Fog	Maximize user satisfaction in terms of cost performance subject to delay, transmission quality, and power control	Student Project matching algorithm combined with user-oriented cooperation (UOC)	Latency, Service provider's revenue, data size, delay
F10	[39]	Restricted fronthaul capacity and computing delay increases the latency	To achieve ultra-low latency and optimized transmission rate	Jointly distributed computing algorithm and distributed content clustering algorithm	Delay, number of users served in fog
F11	[60]	QoS violation and execution cost	Maximize fog resource utilization with response time less than the deadline	constraint based empirical algorithm	Fog Utility, response time, make span
F12	[49]	Fault tolerance, overflow/underflow problem in resource allocation	Maximize Fog utilization	Empirical approach	Response time, DC processing time, total cost (VM cost + data transfer cost)
F13	[58]	QoS aware IoT task placement	Minimum latency and maximum task placement.	Back tracking and heuristic search	Latency and bandwidth
F14	[62]	Integration of spare resources from end-users to fog resource pool	Maximize resource utilization and income of fog broker	crowd funding algorithm approach refining Nash equilibrium	Failure rate of SLA, Task Completion time
F15	[50]	Price cost and time cost issues involved in allocating resources to IoT task in fog	Maximize resource utilization, profit of fog service providers and satisfy QoS requirements	Priced Timed Petri Nets	Task completion cost, make span

F16	[51]	Cost hike due to the unstable and long delay communication link between the medical device and datacentre	Minimize the cost of communication, delay, processing, and deployment to ensure QoS	Mixed Integer Linear Programming (MILP) through joint optimization using 2- phase LP-based heuristic algorithm	Total cost (cost of uplink comm., deployment, processing)
F17	[63]	Assurance of SLA/QoS in IoT service placement and RA in the fog-cloud federation	Improve RA and Optimization of Big data distribution	Decision rules of Linear decision tree approach	Response time, number of VMs used, Number of SLA met
F18	[54]	Joint optimization of resource allocation and carbon footprint issue	Maximize Fog utility and minimize cost with reduced carbon emission	Alternative direction method of multipliers (ADMM) as the proximal algorithm	Fog Utility and carbon emission rate

B. ANALYSIS OF RASP WORKS IN FOG COMPUTING

Paper ID	Ref.	Experiment	Evaluation	Workload	Limitations
F1	[8]	iFogSim with 28 NW configurations for task placement in fog landscape. 2)Test bed to emulate Intelligent transport system	Validated with IBM CPLEX optimization solver results	Simulated data & 65 applications from the Intelligent Transport System (ITS) with 28 scenarios tested.	Applications with independent tasks alone are considered.
F2	[52]	Amazon EC2 instances test bed coded using MATLAB, CVX/MOSEK	evaluated with five allocation schemes GEG, EG, PROP, SWM, MM benchmarks	Data set	Maximum resource capacity of fog nodes not mentioned while max. resource demand used
F3	[55]	Test bed with 25 FAP and 100 mobile devices.	evaluated with SDFC, SSEC, CFIC scheme	Mobile device generated service request (data set)	Due to reactive policy scalability issue arises.
F4	[57]	CloudSim, iFogSim	Validated with IoT based Smart Home application (SHA)	Real time- Small scale smart home automation experiment case study	PSO do not address dynamic scalability
F5	[53]	Test bed with 50 IoT devices and 10 fog devices	Validated for stability, complexity and convergence	Enhanced Mobile Broadband (eMBB) services, Ultra Reliable Low Latency Communication (URLLC) services-delay & BER (Bit Error Rate) intensive	Performance measured only for specific services
F6	[56]	Numerical Experiment and Test bed simulation	QoE at equilibrium with price of anarchy compared with social optimal cost	Simulated mobile request data set	Number of user request and computing services considered constant
F7	[59]	iFogSim, GPR implemented with gptool of python	Fog only tasks compared with fog-cloud	Remote VM application and augmented reality application.	Mobility of accessing device not considered
F8	[7]	Test bed set up	Evaluated with fixed and dynamic threshold for varying resource quota	GNOME to simulate concurrent request	Scalability issue
F9	[61]	Test bed set up with 45 to 210 IoT users	SPA, Random resource allocation, Energy Consumption and delay performance (EDM)	IoT device requests	Reactive policy restricts scalability
F10	[39]	Simulated experiment	Compared with fixed power allocation scheme and random fog clustering scheme	20 requests from 5 users for 20 fog access points	The transmission delay between fog nodes considered negligible
F11	[60]	iFogSim	evaluated with IBM Cplex solver, compared with first fit baseline & pure cloud models	Motion, video, audio, temperature-based applications	The reactive policy does not scale and fails to address stochastic requirements

F12	[49]	Cloud analyst	Efficient resource allocation (ERA) compared with existing Optimize response time (ORT) and Reconfigure dynamically with load balancing (RDLB)	Simulated data	Cannot address stochastic requirements
F13	[58]	Fog torch prototype, a proof of concept java tool.	Evaluated for expected QoS profile in 50 fog nodes	Fire alarm IoT application offered by an insurance company to its customers.	A single application tested for task placement. Scalability problem.
F14	[62]	Test bed with 50 smart phones	Validated with Minimum Migration and MBFD (Modified Best Fit Decreasing)	Test data for pressure application generated by JMeter	The static approach does not support scalability
F15	[50]	Test bed set up with dawn-3000 parallel machine with ten Linux cluster to model fog computing environment	MFR (Mapping Fog Resource to user directory scheme) compared with MinMin and MaxMin algorithm	Random function generated service requests	As resources are mapped to user price, the waiting time for a resource, increases the delay of completing user tasks.
F16	[51]	Test bed set up of 300x300 network size with 80 users and 50 Base stations.	Total cost evaluated across several base stations and 2-phase LP compared with the greedy algorithm	The medical device-generated data traffic	VM deployment in the base station is application-specific
F17	[63]	CloudSim	Internally compared among shared and reserved allocation.	The workload of multimedia big data from fog-cloud broker to use smart devices	Static number of requests and data considered for the experiment
F18	[54]	Mathematical model	Convergence rate of proximal algorithm and ADMM	Video streaming request from Akamai- the world's largest content delivery network	Only Theoretical proof of mathematical model analyzed

Solution. In general, the existing works were from the service provider's point of view saving their cost. A RASP strategy that prioritizes consumer's profit, needs focus. Deployment of multi-tier and parallel applications in fog nodes is another issue that needs attention.

The unexpected network traffic and access rate of the hosted applications were not foreseen during SLA. This leads not only in the violation of QoS requirements but some catastrophic failures of resource access. Hence dynamic provision, to monitor and configure the resources automatically with intelligence is the need for such a situation. Research on autonomic computing that possesses self-management capability will enhance the RASP strategy.

The proliferation of IoT requires unlimited bandwidth. The huge number of heterogeneous geo-distributed devices involved in the fog layer that handles IoT consumes enormous energy. Instead of draining the available energy, fog nodes that work on solar and green energy should be brought into usage. Hence, a Fog-based RASP solution that supports green environmental sustainability needs focus.

The manufacturing units in Industry, nowadays depend on Fog services for instantaneous processing. But, the protocol interoperability problem between the assembling units and Fog devices causes a delay that is not tolerable in Industrial IoT. With fewer works carried out in this area, it remains yet another open challenge in Fog research.

Findings show that based on the delay constraint of the applications, the arriving requests are segregated among the Cloud and Fog for processing. But, the question arises how the decision is made when the delay constraint is not explicitly mentioned. One possible approach is that the Cloud/Fog center can be decided based on the application type. Service requests from critical health-care, disaster management, real-time chemical reactors, and Industrial IoT can be considered as emergent applications that need to be processed in the Fog layer.

Further, the efficiency of the RASP system can be escalated by clustering the fog nodes on application basis for processing. Instead of making all fog nodes available for processing, certain fog nodes can be employed for general purposes while the rest of the fog nodes can be reserved exclusively for emergent applications. Algorithms are to be devised that ensure maximum utilization of the fog nodes. The idle fog cluster can be employed either for the migrated emergent applications or for the local non-emergent applications during peak hours. As Fog computing is still in its infancy stage, standard protocols are yet to be explored.

VIII. CONCLUSION

The survey elaborates various RASP strategies in Fog and Cloud environments. The survey investigated the individual work from the viewpoint of, the problem defined, objective set, algorithm adopted, performance metrics addressed, experiment and evaluation tools employed, and the workloads used for testing. The tabulated information presents an exhaustive analysis of the individual work with their limitations projected as open challenges.

Although review articles exclusive to Cloud and Fog exists, the proposed survey explores the RASP problem, in Cloud and Fog for IoT applications. The survey stands unique to employ techniques like Reinforcement Learning (RL) and Energy Efficient Computing (EEC) to save cost and energy respectively. Sure enough, the survey will motivate the researchers to focus on the research gaps and helps them to conceive innovative RASP solutions in the Fog-Cloud federation.

REFERENCES

[1] Z.Alhara, F.Alvares, H.Bruneliere, J.Lejeune, C.Prud'Homme, and T.Ledoux, "CoMe4ACloud:Anend-to-end framework for autonomic Cloud systems," Future Gener. Comput. Syst.,vol. 86, pp. 339–354,Sep. 2018.

[2] M. R. Anawar, S. Wang, M. Azam Zia, A. K. Jadoon, U. Akram, and S. Raza, "Fog Computing: An Overview of Big IoT Data Analytics," Wirel. Commun. Mob. Comput., vol. 2018, pp. 1–22, 2018.

[3] R. K. Naha et al., "Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions," ArXiv180700976 Cs, Jul. 2018, Accessed: Oct. 29, 2019.

[4] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," IEEE Commun. Surv. Tutor., vol. 20, no. 1, pp. 416–464, 2018.

[5] A. Hameed et al., "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," Computing, vol. 98, no. 7, pp. 751–774, Jul. 2016.

[6] M. Cheng, J. Li, and S. Nazarian, "DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers," in 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), Jeju, Jan. 2018, pp. 129–134.

[7] L. Yin, J. Luo, and H. Luo, "Tasks Scheduling and Resource Allocation in Fog Computing Based on Containers for Smart Manufacturing," IEEE Trans. Ind. Inform., vol. 14, no.10,pp. 4712–4721, Oct. 2018.

[8] M.-Q. Tran, D. T. Nguyen, V. A. Le, D. H. Nguyen, and T. V. Pham, "Task Placement on Fog Computing Made Efficient for IoTApplication Provision,"Wirel. Commun.Mob.Comput.,vol.2019, pp.1–17,Jan. 2019.

[9] S. Soltani, P. Martin, and K. Elgazzar, "A hybrid approach to automatic IaaS service selection," J. Cloud Comput., vol. 7, no. 1, Dec. 2018.

[10] A. Singh and Y. Viniotis, "Resource allocation for IoT applications in cloud environments," in 2017 Inter Conf on Computing, Networking and Communications, Silicon Valley, CA, USA, Jan. 2017, pp. 719–723.

[11] T. Bhardwaj and S. C. Sharma, "Fuzzy logic-based elasticity controller for autonomic resource provisioning in parallel scientific applications: A cloud computing perspective," Comput. Electr. Eng., vol. 70, pp. 1049–1073, Aug. 2018.

[12] E.I.Djebbar and G.Belalem,"Tasks Scheduling and Resource Allocation for High Data Mgmt in ScientificCloud Computing Environment,"in Mobile, Secure, andProgrammable Networking, vol. 10026, S.Boumerdassi, É.Renault, and S.Bouzefrane, Eds.Cham: Springer Inter. Publishing, 2016, pp. 16–27.

[13] H. Atlam, R. Walters, and G. Wills, "Fog Computing and the Internet of Things: A Review," Big Data Cogn. Comput., vol. 2, no. 2, p. 10, Apr. 2018.

[14] A. S. Gowri and P. Shanthi Bala, (2020). Fog Resource Allocation Through Machine Learning Algorithm. In Goundar, S., Bhushan, S. B., & Rayani, P. K. (Ed.), Architecture and Security Issues in Fog Computing Applications (pp. 1-41) ch001. IGI Global.

[15] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," in Internet of Everything, B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Singapore: Springer Singapore, 2018, pp. 103–130.

[16] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. S. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," Digit. Commun. Netw., vol. 5, no. 1, pp. 3–9, Feb. 2019.

[17] H. Duan, C. Chen, G. Min, and Y. Wu, "Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems," Future Gener. Comput. Syst., vol. 74, pp. 142–150, Sep. 2017.

[18] T.Thein,M.M.Myo,S.Parvin, and A. Gawanmeh, "Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers," J. King Saud Univ.- Comput. Inf. Sci.,Nov. 2018.

[19] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," J. Netw. Comput. Appl., vol. 59, pp. 46–54, Jan. 2016.

[20] R. Basmadjian, H. Meer, R. Lent, and G. Giuliani, "Cloud computing and its interest in saving energy: the use case of a private cloud," J. Cloud Comput. Adv. Syst. Appl., vol. 1, no. 1, p. 5, 2012.

[21] M. Xu and R. Buyya, "Brownout Approach for Adaptive Management of Resources and Applications in Cloud Computing Systems: A Taxonomy and Future Directions," ACM Comput. Surv., vol. 52, no. 1, pp. 1–27, Jan. 2019.

[22] M. Hosseini Shirvani, A. M. Rahmani, and A. Sahafi, "A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter: Taxonomy and challenges," J. King Saud Univ. - Comput. Inf. Sci., Jul. 2018.

[23] M. Zakarya and L. Gillam, "Energy efficient computing, clusters, grids and clouds: A taxonomy and survey," Sustain. Comput. Inform. Syst., vol. 14, pp. 13–33, Jun. 2017.

[24] S. Zahoor and R. N. Mir, "Resource management in pervasive Internet of Things: A survey," J. King Saud Univ. - Comput. Inf. Sci., Sep. 2018.

[25] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian, "Resource Management Approaches in Fog Computing: a Comprehensive Review," J. Grid Comput., Sep. 2019.

[26] J. Rao, X. Bu, C.-Z. Xu, and K. Wang, "A Distributed Self-Learning Approach for Elastic Provisioning of Virtualized Cloud Resources," in IEEE 19th International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems,Singapore, Singapore, Jul. 2011, pp. 45–54.

[27] A. Keshavarzi, A. Toroghi Haghighat, and M. Bohlouli, "Adaptive Resource Management and Provisioning in the Cloud Computing: A Survey of Definitions, Standards and Research Roadmaps," KSII Trans. Internet Inf. Syst., vol. 11, no. 9, Sep. 2017.

[28] P. D. Kaur and I. Chana, "A resource elasticity framework for QoS-aware execution of cloud applications," Future Gener. Comput. Syst., vol. 37, pp. 14–25, Jul. 2014.

[29] K.RahimiZadeh,M.AnaLoui, P. Kabiri, and B. Javadi,"Performance modeling and analysis of virtualized multi-tier applications under dynamic workloads," J. Netw.Comput.Appl.,vol.56,pp.166–187, Oct. 2015.

[30] G.Ismayilov and H.R.Topcuoglu,"Neuralnetworkbased multi-objectiveevolutionary algorithm for dynamic workflow scheduling in cloud," Future Gener. Comput. Syst., vol. 102, pp. 307–322, Jan. 2020.

[31] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," Future Gener. Comput. Syst., vol. 28, no. 1, pp. 155–162, Jan. 2012.

[32] A. Ashraf, "Automatic Cloud Resource Scaling Algorithm based on Long Short-Term Memory Recurrent Neural Network," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 12, 2016.

[33] Dr. A. Agarwal and S. Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment," Int. J. Comput. Trends Technol., vol. 9, no. 7, pp. 344–349, Mar. 2014.

[34] J. Espadas, A. Molina, G. Jiménez, M. Molina, R. Ramírez, and D. Concha, "A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures," Future Gener. Comput. Syst., vol. 29, no. 1, pp. 273–286, Jan. 2013.

[35] E. Casalicchio and L. Silvestri, "Mechanisms for SLA provisioning in cloud-based service providers," Comput. Netw., vol. 57, no. 3, pp. 795–810, Feb. 2013.

[36] H. Xu and B. Li, "Anchor: A Versatile and Efficient Framework for Resource Management in the Cloud," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1066–1076, Jun. 2013.

[37] A. Nassar and Y. Yilmaz, "Reinforcement Learning for Adaptive Resource Allocation in Fog RAN for IoT With Heterogeneous Latency Requirements," IEEE Access, vol. 7, pp. 128014–128025, 2019.

[38] F.Bahrpeyma,H.Haghighi, and A.Zakerolhosseini, "An adaptive RL based approach for dynamic resource provisioning in Cloud virtualized data centers," Computing, vol. 97, no. 12, pp. 1209–1234, Dec. 2015.

[39] G. M. S. Rahman, M. Peng, K. Zhang, and S. Chen, "Radio Resource Allocation for Achieving Ultra-Low Latency in Fog Radio Access Networks," IEEE Access, vol. 6, pp. 17442–17454, 2018.

[40] K. Gai and M. Qiu, "Optimal resource allocation using reinforcement learning for IoT content-centric services," Appl. Soft Comput., vol. 70, pp. 12–21, Sep. 2018.

[41] Xiangping Bu, Jia Rao, and Cheng-Zhong Xu, "Coordinated Self-Configuration of Virtual Machines and Appliances Using a Model-Free Learning Approach," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 4, pp. 681–690, Apr. 2013.

[42] C.-Z. Xu, J. Rao, and X. Bu, "URL: A unified reinforcement learning approach for autonomic cloud management," J. Parallel Distrib. Comput., vol. 72, no. 2, pp. 95–105, Feb. 2012.

[43] X. Dutreilh, S. Kirgizov, O. Melekhova, J. Malenfant, and N. Rivierre, "Using Reinforcement Learning for Autonomic Resource Allocation in Clouds: Towards a Fully Automated Workflow," p. 8, 2011.

[44] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Gener. Comput. Syst., vol. 28, no. 5, pp. 755–768, May 2012.

[45] M. Shelar, S. Sane, V. Kharat, and R. Jadhav, "Autonomic and energy-aware resource allocation for efficient management of cloud data centre," in 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, Apr. 2017, pp. 1–8.

[46] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters," Future Gener. Comput. Syst., vol. 37, pp. 141–147, Jul. 2014.

[47] F. Fargo, C. Tunc, Y. Al-Nashif, A. Akoglu, and S. Hariri, "Autonomic Workload and Resources Management of Cloud Computing Services," in 2014 Inter. Conf. on Cloud and Autonomic Computing, United Kingdom, Sep. 2014, pp. 101–110.

[48] Z. Zhang, Q. Guan, and S. Fu, "An adaptive power management framework for autonomic resource configuration in cloud computing infrastructures," in 2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC), Austin, TX, USA, Dec. 2012, pp. 51–60.

[49] M. Mulla, M. Satabache, and N. Purohit, "An Efficient Architecture for Resource Provisioning in Fog Computing," Int. J. Sci. Res. IJSR, vol. 6, no. 1, pp. 2065–2069, Jan. 2017.

[50] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets," IEEE Internet Things J., vol. 4, no. 5, pp. 1216–1228, Oct. 2017.

[51] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System," IEEE Trans. Emerg. Top. Comput., vol. 5, no. 1, pp. 108–119, Jan. 2017.

[52] D. T. Nguyen, L. B. Le, and V. Bhargava, "A Market-Based Framework for Multi-Resource Allocation in Fog Computing," ArXiv180709756 Cs, Apr. 2019, Accessed: Oct. 29, 2019.

[53] S. F. Abedin, Md. G. R. Alam, S. M. A. Kazmi, N. H. Tran, D. Niyato, and C. S. Hong, "Resource Allocation for Ultra-Reliable and Enhanced Mobile Broadband IoT Applications in Fog Network," IEEE Trans. Commun., vol. 67, no. 1, pp. 489–502, Jan. 2019.

[54] C.T.Do,N.H.Tran, Chuan Pham, Md. G. R. Alam, Jae Hyeok Son, and C. S. Hong, "A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing," in Inter Conf. on Information Networking (ICOIN), Cambodia, Jan. 2015, pp. 324–329.

[55] S. Kim, "Novel Resource Allocation Algorithms for the Social Internet of Things Based Fog Computing Paradigm," Wirel. Commun. Mob. Comput., vol. 2019, pp. 1–11, Feb. 2019.

[56] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical Fog-Cloud Computing for IoT Systems: A Computation Offloading Game," IEEE Internet Things J., vol. 5, no. 4, pp. 3246–3257, Aug. 2018.

[57] S. S. Gill, P. Garraghan, and R. Buyya, "ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices," J. Syst. Softw., vol. 154, pp. 125–138, Aug. 2019.

[58] A. Brogi and S. Forti, "QoS-Aware Deployment of IoT Applications Through the Fog," IEEE Internet Things J., vol. 4, no. 5, pp. 1185–1192, Oct. 2017.

[59] R.A.C.daSilvaand N. L. S. da Fonseca, "Resource Allocation Mechanism for a Fog-Cloud Infrastructure," in 2018 IEEE Inter Conference on Communications (ICC), Kansas City, MO, May 2018, pp. 1–6.

[60] O.Skarlat,M. Nardelli,S.Schulte,and S.Dustdar,"Towards QoS-Aware Fog Service Placement," in IEEE 1st International Conf.rence on Fog and Edge Computing (ICFEC), Madrid, Spain, May 2017, pp. 89–96.

[61] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint Radio and Computational Resource Allocation in IoT Fog Computing," IEEE Trans. Veh. Technol., vol. 67, no. 8, pp. 7475–7484, Aug. 2018.

[62] Y. Sun and N. Zhang, "A resource-sharing model based on a repeated game in fog computing," Saudi J. Biol. Sci., vol. 24, no. 3, pp. 687–694, Mar. 2017.

[63] A. A. Alsaffar, H. P. Pham, C.-S. Hong, E.-N. Huh, and M. Aazam, "An Architecture of IoT Service Delegation and Resource Allocation Based on Collaboration between Fog and Cloud Computing," Mob. Inf. Syst., vol. 2016, pp. 1–15, 2016.