Cone Model in Resource Provisioning for Service-Oriented Architecture System: An Effective Network Management to the Internet of Things

Robin S. Bhadoria¹, Hatim G. Zaini², Md. Manzar Nezami³, Md. Danish Equba⁴, Elfatih Elmubarak Mustafa⁵, and Sherif S. M. Ghoneim⁶, Senior Member IEEE

¹Department of Computer Engineering and Applications, GLA University, Mathura, 281406 India
²Computer Engineering Department, College of Computer and Information Technology, Taif University, Al Huwaya, P.O. Box 11099, Taif 21944, Saudi Arabia
³Department of Electronics and Communication Engineering, GLA University, Mathura, 281406 India
⁴Department of Electrical Engineering, Galgotias College of Engineering and Technology, Greater Noida, 201306, India
⁵Department of Computer Science, College of Science and Arts Balqarn, University of Bisha, P.O. Box 60, Bisha 61985, Saudi Arabia
⁶Department of Electrical Engineering, College of Engineering, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia

Corresponding author: Sherif S. M. Ghoneim (e-mail: s.ghoneim@tu.edu.sa).

This work was supported by Taif University Researchers Supporting Project TURSP- 2020/345, Taif University, Taif, Saudi Arabia

ABSTRACT The rapid growth of technologies like the Internet of Things (IoT) has thrown newer challenges to resource management, making it one of the crucial tasks in dynamic application scenarios of IoT. Service-Oriented Architecture (SOA) plays an important role in this respect by proffering reference architecture towards better positioning of infrastructural software and hardware components, mostly comprehending from a service-oriented perspective. In this direction, we propose Cone Model that deals with the performance and handling issues with SOA system. Service integration is all important feature of SOA that supports communication and interaction between multiple service components (Service Containers). In addition to discussing the related work for critical resource allocation algorithms, this paper proposes Cone Model for assigning an available set of resources to requested services, explained through mathematical expressions and algorithms of the model. These algorithms justify that how different services from various devices in IoT can be provisioned on the adoption of this proposed cone model. It is illustrated through the estimated utilization of resources by the services. The proposed model aligns with the applications in multifarious domains, primarily in the IoT and in areas of resource sharing for cloud computing and massive data centers.

INDEX TERMS Cone model, Critical resource allocation, Internet of things, Service computing, Service-oriented architecture (SOA), SOA reference architecture

I. INTRODUCTION Recent years have seen a tremendous surge in domains like the Internet of Things (IoT). Apart from fast development in the hardware domain for technologies and platform like IoT, the need of software solutions to provide logical infrastructure to the system on one hand and proffer suitable device abstraction and efficiency is inevitable. The necessary infrastructure for the variety of services provided by the IoT system is crucial and Service-Oriented Architecture (SOA) plays an important role in this respect. SOA defines the architectural paradigm towards designing efficient system, that facilitate providing services, including support for reusability, scalability and flexibility in delivering end-to-end business solutions. It allows architectural developers to build robust guidelines and templates for better positioning of the architectural components in their infrastructural resources. SOA Reference Architecture (SOA-RA) does this task of offering better visibility to the system, rather answering “what does SOA systems actually contain” [1-2]. SOA for IoT is subtly different from its usual approach and focuses mostly on the reusability of software functionalities for different tasks [24]. Considering the model from a service-oriented perspective, the focus is on the design the interactions of the IoT system with the outside world, which
differs slightly from the traditional perspective of IoT. Introduction of SOA in IoT systems bolsters the security due to different levels of abstraction and interfaces on one hand and facilitates the communication between heterogeneous entities tied in the IoT system by virtualizing the communications and data distribution on the other hand. SOA-RA is well defined and explained by many software industry and consortiums like OpenGroup Inc., OASIS, IBM Inc. It consists of layered architecture for the design and implementation of SOA system [3]. Fig. 1 illustrates a five-layer structure for SOA-RA system.

**FIGURE 1.** SOA reference architecture (logical view)

It consists of five layers: 1) Operational system layer consists of key components like storage, processor units, etc. It is a position where the actual service provider resides. It is also responsible for running and deploying services over web [2, 3]. 2) Service components layer manages and organizes service components (service modules), to facilitate the services in business process applications. It also supports the implementation of a particular service that is governed by the enterprise. It binds the service contract between its user and service suppliers by enabling loose coupling. 3) **Service Layer** consists of all logical services that have been supported by SOA. It consists the service stack that may include interactive services, orchestration and business process services, data information service, application service, accessibility service, and service governance. This layer also defines the business-functionality support and capabilities for the services in SOA. 4) **Business Process layer** manages the interaction between services and business processes to retain the data and control flow. It also maintains the sequence of processes aligned with business goals. It acts as a central coordinator in aggregating loosely coupled services with SOA. It also helps in integrating web services in delivering end-to-end enterprise solutions. 5) **Consumer layer** is the position where a consumer resides. It can be a person, browser or automation, for interaction with service in SOA.

### A. CHALLENGES FOR RESOURCE ALLOCATION IN SOA SYSTEM

The major challenges for any SOA system with resource handling are linked to its service quality, data collection from heterogeneous sources, optimal methodology and algorithm for resource allocation and satisfy quality of service (QoS) requirements to synchronize the workflows [16]. These challenges are directly related to the limitation of available resources for large-scale service integration over the grid and cloud computational resource handling [31]. In this respect, the following issues stand important:

**Resource Scalability:** When the computation is performed for large-scale service integration with the pool of resources, it needs a dynamic, scalable distributed system that works together as a powerful virtual machine over the internet. All such processes need to find an optimal resource allocation technique that supports the adaptability of resources in an efficient and scalable manner.

**Dynamic Runtime Environment:** A system must be developed and designed in such a manner that operates on open and dynamic environment with adaptive resource allocation methodology [8]. This vision combines the concept of Software-as-a-Service (SaaS) and Resource-as-a-Service (RaaS) together on common platform that supports dynamic runtime environment like self-management, context-aware messaging and implements efficient resource allocation algorithm. SOA system needs to be responsive and adaptable to the change in situations that arrive with large-scale service integration. It may include the rate at which service request arrived, workflow synchronization, priorities for task (workload), QoS milieu, resource availability, and various relevant environmental features that support dynamic runtime over the internet.

**QoS Estimation:** A SOA system must be capable of efficiently minimizing the gap between resource allocation and workflows. Analyzing the QoS feature need to considered with the issues of with Service-Level Agreement (SLA), loosely coupled and compositional behavior of SOA system. Service monitoring is evaluated based on the hosting server, CPU time, network bandwidth and memory used. In practice, a service generates a service instance called service component that contains the actual service to deploy, would depending on workflow and services beneath. These components tangibly utilize the virtualized resource made available through the virtual machine (VM) concept. Overall, analysis of these components could produce a brief estimation of QoS in a dynamic runtime environment.

**Optimal Resource Algorithm:** A SOA system must handle resource allocation at runtime to support the QoS requirements. Thus, an algorithm must be devised to manage multiple workflows with dynamic resource assignments. Such an algorithm should utilize the existing resources to
their maximum and generate maximum throughput. This algorithm should also be adopted by service providers to facilitate its clients for resource allocation. Concurrently, multiple service components can utilize these existing set of resources to federate optimal resource allocation methodology [16].

**Usage Costing/Pricing of Allocated Resources:** While in the utilization of resources, costing/pricing would be considered according to pay-per-usage strategy for available resources. There have been several models that support resource renting policy which must be part of SLA at time of contract signing by client. The duration of service would be count on the basis of use and generate invoice against it. An SOA system must be capable of adapting the change in resources according to the need of the client over runtime [8]. The motivation for this paper is related to the efficient handling and provisioning of resources over the cloud and how it directly affects workload for IoT platform. There have been several cases observed where sharing of resources became critical issue for hosting various services over the cloud. The solution for such challenge can be done by adopting the propose “Cone Model” for assigning an available set of resources to requested services. It justifies that how different services from various devices of IoT can be provisioned on the adoption of this proposed cone model. It also aligns with the applications in multifarious domains for resource sharing in massive data centers.

The major contributions of this paper can be mentioned as per below:

- This paper discusses the role of SOA with its logical view reference architecture and also points out the challenges for resource allocation in SOA system.
- Also, presents the adapting changes for efficient resource provisioning according to need of client over runtime and different workloads with its extensive survey to Internet of Things (IoT).
- This paper also proposed the cone model that helps in efficient handling and provisioning of resources in computing over IoT. Also, justifies how many resources and its instances can be engaged by a particular service over SOA system.
- Derive the mathematical implication of cone model in resource provisioning and service accessibility with IoT adopted the system.
- Brief the algorithms for calculating throughput and formulating optimization methods to allocate critical resources in service-based system for IoT devices.

Section II briefs about the background and related work in resource allocation methodology for service-based system. This section also discusses different models and techniques used by several authors in critical resource allocation. Section III presents the proposed cone model in SOA system, which illustrates the relationship between service components and its associated resource (server instances).

Section IV provides the mathematical view for cone model along with the proposed algorithm for critical resource allocation. Section V demonstrates the results concurred using the proposed algorithm based on the cone-model for critical resource allocation and utilization based on SOA system. This paper is concluded with the facts of cone model in the implication of better handling and allocation of resources during runtime for service computing in SOA system and discusses the significance and relevance of this model in the bigger domain of IoT, cloud computing and data sharing.

**B. SOA with Federate and Communicate in Internet of Things**

With the recent advances in the domain of IoT, the focus is not just on the hardware design but also on the enhancement of performance from a software perspective [17]. IoT resource allocation problem is one of the most critical issues in this respect, and this paper presents a perspective of SOA for resource allocation between multiple services components. The problem of resource allocation spans from the small IoT devices to massive cloud-based systems as well. This work is aimed at providing a solution to the resource allocation problem in SOA. It explained the paradigm for SOA to build strong guidelines in better positioning of system components for critical resources on the one hand. These resources can dynamically be facilitated to achieve momentous economies of scale, illustrated in the SOA-RA. The SOA-RA sets up the strategy that helps in handling the existing set of resource. On the other hand, the proposed Cone Model in the SOA system helps to comprehend the core functionality in critical resource allocation [5]. This cone model is also applicable in different areas like space positioning, weather forecasting, rocket engine performance etc. This model discussed and commented on the accessibility of services through its components – namely “service containers” to server instances for resources. Efficient utilization of resources can easily be managed with this model. This paper also demonstrated some existing algorithms for the allocation of critical resources based on priority, presenting optimal throughput.

Service integration is an important aspect of SOA to federate and communicate multiple services on a common platform [11]. The probability related to resource allocation and utilization via multiple service components has been determined through mathematical models. This model also comments on issue related to a particular service that requires the specific number of resources to complete its execution. This service assessment can be done using this mathematical model. Estimating the number of resources also discussed the “expectation or expected value” concept for the SOA system. This gives an idea – up to what extent an organization should stock and hold resources of a
particular type and what could happen if we increase the stock units.

II. BACKGROUND AND LITERATURE WORK

In allocating and monitoring resources, workflow management has been a crucial issue for consideration. A system must be capable enough to handle multiple services. Quality of service (QoS) is a primary requirement to satisfy numerous application services over runtime due to loose-coupling, reusability, and dynamic behavior of SOA systems. With the cloud computing aspect, a market-based autonomic resource management system has been proposed [19], which provides an interface based on Service-Level Agreement (SLA) in managing and handling resource allocation between cloud service providers and clients. For the advancement in middleware technology, a sophisticated system has been designed and developed to manage architecture based on dynamic resource management [35]. At runtime, this optimizes an available set of resources with the current demands and status of particular resources. This system supports QoS even for overloaded time and produces a maximum throughput during its critical operation. A workflow-based resource broker for grid computing systems has been depicted in [10]. The principal job of these resource brokers is to check the availability of resources and synchronize workflow. This requirement helps the resource provider to administer multiple domains of services of the same kind. These resource providers also support interfaces that could be accessible through user’s credentials for an available set of resources in grid computing [9].

For web-services applications, a local and decentralized greedy approach has been devised for dynamic resource allocation in [19]. This proposed methodology involves a software agent that trades all the resources and network-related services between the service provider and its associated client. For cost estimation and pricing for the usage of resources in fog computing, an optimal methodology has been proposed for web service applications by [18]. With dynamic multiservice networks, this web-based application not only provides an optimal and well-organized resource allocation but also does pricing estimation to next-generation multiclass networks. In addition, it also formulates a nonlinear pricing model that provides a solution for network delay constraints.

| Reference       | Resource Handling                          | Load Balancing                                      | Adopted Model                                      | Comment                                |
|-----------------|--------------------------------------------|------------------------------------------------------|-----------------------------------------------------|----------------------------------------|
| [17] Li et al., 2021 | Discuss Enterprise Architecture in IoT resource handling | No attention on Load balancing factor                 | Brief several technologies used for IoT resource allocation | No emphasis given on throughput for resource allocation |
| [23] Tsai, 2018 | Discusses the IoT resource allocation problem (IRAP) | Uses concepts of metaheuristic algorithm             | Search Economics for IoT Resource Allocation (SEIRA) | Limited for region based collected information |
| [29] Zhan et al., 2020 | Service-oriented IoT resources access and provisioning | Design & develop frame-based protocol stack for IoT system | IoT Context-Aware Environment | No focus made on throughput of maps resource operations |
| [35] Deng et al., 2020 | Use Markov decision process (MDP) for allocation scheme | Use edge servers for handling IoT devices             | Use reinforcement learning (RL) method              | No emphasis given on system states segregation |
| [17] Li et al., 2021 | Discuss Enterprise Architecture in IoT resource handling | No attention on Load balancing factor                 | Brief several technologies used for IoT resource allocation | No emphasis given on throughput for resource allocation |
| [23] Tsai, 2018 | Discusses the IoT resource allocation problem (IRAP) | Uses concepts of metaheuristic algorithm             | Search Economics for IoT Resource Allocation (SEIRA) | Limited for region based collected information |
| [29] Zhan et al., 2020 | Service-oriented IoT resources access and provisioning | Design & develop frame-based protocol stack for IoT system | IoT Context-Aware Environment | No focus made on throughput of maps resource operations |
| [35] Deng et al., 2020 | Use Markov decision process (MDP) for allocation scheme | Use edge servers for handling IoT devices             | Use reinforcement learning (RL) method              | No emphasis given on system states segregation |
| [17] Li et al., 2021 | Discuss Enterprise Architecture in IoT resource handling | No attention on Load balancing factor                 | Brief several technologies used for IoT resource allocation | No emphasis given on throughput for resource allocation |
| [23] Tsai, 2018 | Discusses the IoT resource allocation problem (IRAP) | Uses concepts of metaheuristic algorithm             | Search Economics for IoT Resource Allocation (SEIRA) | Limited for region based collected information |
For service-based systems like utility computing system, cloud computing system, and grid computing system that effectively use various application services simultaneously into a distributed environment, SOA system has been widely accepted as a distributed kind of system that handles and federates different services with different environmental support. A service that cannot decompose to further smaller components is called atomic service, and the rest of all other services are termed composite service. This supports the efficient organization of available resources for single (atomic) as well as group (composite) service(s), which has been done in [13]. This paper shows that each workflow is made up of multiple atomic services which is govern by its critical resources.

With the recent advances in the domain of IoT, the focus is not just on the hardware design but also on the enhancement of performance from a software perspective [17]. IoT resource allocation problem is one of the most critical issues in this respect, and this paper presents a perspective of SOA for resource allocation between multiple services components. The problem of resource allocation spans from the small IoT devices to massive cloud-based systems as well. This work is aimed at providing a solution to the resource allocation problem in SOA. It explained the paradigm for SOA to build strong guidelines in better positioning of system components for critical resources on the one hand. These resources can dynamically be facilitated to achieve momentous economies of scale, illustrated in the SOA-RA. The SOA-RA sets up the strategy that helps in handling the existing set of resource. On the other hand, the proposed Cone Model in the SOA system helps to comprehend the core functionality in critical resource allocation [5]. This cone model is also applicable in different areas like space positioning, weather forecasting, rocket engine performance etc. This model discussed and commented on the accessibility of services through its components – namely “service containers” to server instances for resources. Efficient utilization of resources can easily be managed with this model. This paper also demonstrated some existing algorithms for the allocation of critical resources based on priority, presenting optimal throughput.

Service integration is an important aspect of SOA to federate and communicate multiple services on a common platform [11]. The probability related to resource allocation and utilization via multiple service components has been determined through mathematical models. This model also comments on issue related to a particular service that requires the specific number of resources to complete its execution. This service assessment can be done using this mathematical model. Estimating the number of resources also discussed the “expectation or expected value” concept for the SOA system. This gives an idea – up to what extent an organization should stock and hold resources of a particular type and what could happen if we increase the stock units.

### III. PROPOSED CONE MODEL IN SERVICE COMPUTING FOR SERVICE ORIENTED ARCHITECTURE (SOA) SYSTEM

The cone model is derived from geometrical cone sharp. It justifies the one-to-many mapping relationship between multiple objects or vice-versa. It provides basic structure in forming resources utilization and support in establishing the connection between multiple service component in SOA systems. It also offers a path through which routing can be possible. Cone model has also been applied to multiple areas like space weather forecasting, in which this model improves accuracy of kinematical and geometrical properties that helps in identifying exact location. Work done in [4] used cone-based model with Coronal Mass Ejections (CME), in which several points tried to converge at the center, producing accurate results. This model is also adopted by [5] into Solar Energetic Particle (SEP) events analysis and forecasting. In Rocket Engine Performance, a small cone is formed into engine’s combustion chamber and gradually during the burning of propellant fuel. The implication of cone model in rocket engine depics that sharper the cone shape is, the larger would be thrust generated into the combustion chamber. Actually, the shape of cone is used in predicting the amount of thrust generated
and the performance of rocket engine [6]. With this fundamental understanding, this work employed the cone model for the SOA-based Enterprise Service Computing Architecture to predict the service resource estimation. Each service needs resources to process its business logic – either it could be memory, processor, or bandwidth [11].

This model could best explain the critical resources allocation and utilization, in which service requester (or simple service) is situated at the top of cone and the rest of the infrastructure resources at the bottom. This model can be best viewed in SOA Reference Architecture with layered approach, especially in the lower three layers, as shown in Fig. 2. Whenever any service is requested from the client side for a particular purpose, it needs a lot of resources to hold for its processing. This enterprise service architecture is an allusion from SOA Reference Architecture that had been initiated from several developers & designers from different software industries [33]. Each service is incorporated by one or more service components, which is a ‘service container’. These containers are the entities carrying data and associated information, which is plug-in and float with the service bus. This service bus helps in service integration between multiple services. This bus is known as “Enterprise Service Bus (ESB)”.

ESB provides a common framework for developers to federate different environments on the common platform [36]. The ESB works in the Service Component Layer of SOA-RA (Fig. 1). The cone model provides a conceptual view for services and maps the resources held/required into a logical cone shape based on the relationship between a particular service and currently held resources by it. This also explains how services are being accessed and could be allotted with the available resources. Similarly, the reverse cone model provides a conceptual view of the available resources and resources occupied by a set of services [37]. Such methodology could be best fitted into cloud-based computing systems.

For example, there are two services, namely service ‘A’ and service ‘B’. Each service is decomposed into its respective components. Here, Service ‘A’ has two components i.e., service component 1 and 2. Similarly, Service ‘B’ also has two components i.e., service component 3 and 4. These service components have actually engaged these resources in the operational system layer as shown in figure 2 and 4. This specifies the engagement of existing resource by service components and finally by its supervisory or parent service. The engagement of resources by service and its components creates a geometrical shape cone. It illustrates how many resources and instances can be engaged by a particular service. It also rectifies the possibility of deadlock occurrence for resources as it shows the details regarding available set of resources and its associated services. It also provides the path through which routing can also possible for message handling through service bus (which is job of ESB).

![Logical view of cone model in SOA system](image1)

**FIGURE 2.** Logical view of cone model in SOA system

Service Layer

![Service A](image2)

Service B

Service Component Layer

![Service Component 1](image3)

Service Component 2

Service Component 3

Service Component 4

Operational System Layer

Processors/Servers

Routers

Storage

Transporters/Refactors

The concept of resource allocation can also be viewed as a particular access and assignment of the available instance of resources that could be provided from server side in SOA system [21]. Whenever a particular resource is being engaged by multiple services, then it creates an inverted cone sharp geometry in accessing that particular resource. This can be termed as INVERTED CONE MODEL, as shown in Fig. 4 and Fig. 5.

![Abstract view of cone model](image4)

**FIGURE 3.** Abstract view of cone model

The concept of resource allocation can also be viewed as a particular access and assignment of the available instance of resources that could be provided from server side in SOA system [21]. Whenever a particular resource is being engaged by multiple services, then it creates an inverted cone sharp geometry in accessing that particular resource. This can be termed as INVERTED CONE MODEL, as shown in Fig. 4 and Fig. 5.
The abstract view of the above explained inverted Cone Model has been inherited from work in [5-6] that justified “Identifying and characterizing the cone model”. Model concept can be portrayed as shown in Fig. 5. This shows that multiple services are using a particular resource instance at given time. This concept raises the issue of capacity of a particular resource with its efficiency to being accessed. The relationship between these two proposed models is based on multiple resources and published services from particular domain of application. It could be a point of discussion in enterprise service computing architecture [22].

IV. THE IMPLICATION OF THE CONE MODEL IN RESOURCE PROVISIONING FOR SOA SYSTEM

This paper presents the mathematical idea for allocation and sharing of critical resources by multiple services components in SOA system [31]. This mathematical expression illustrates how particular service component can access and utilize particular set of resources as depicted in cone model Figures 2 and 4.

This paper also provides the concept for handling and provisioning of resources over the cloud and how it is directly affects workload for IoT platform. The role of cone mode and inverted cone model can be used in solving the problem of critical issue in hosting various services over the cloud. The solution for such challenge can be done by adopting the propose “Cone Model” & “Inverted Cone Model” algorithms for assigning an available set of resources to requested services. It justifies that how different services from various devices of IoT can be provisioned on the adoption of this proposed cone model. It also aligns with the applications in multifarious domains for resource sharing in massive data centers.

To overwhelm the functionality of service computing, discrete random variable stands as a better option to point out probability aspect in resource allocation and utilization by multiple service components [14]. This helps to calculate the probability of obtaining critical resources by these service components. Such process continues until each service component obtain or utilize a complete set of resources to fulfill its entire execution. It also provides an aggregated probability for supervisory or parent service.

| Table 2 | Various Symbol justification |
|---------|-----------------------------|
| Symbol  | Justification               |
| N       | distinct resources          |
| R       | random variable             |
| $S_i$   | service components          |
| k       | number of successful or grant resources instances |
| P       | probability mass function   |

Assuming that there are ‘N’ distinct resources and that one service process need to obtains or utilizes for completion of its assigned task. It should independent of any previous selection from such allocation. A random variable \( R \), is such number from resource instances which is required to be collected until one obtains a complete set of available resource instances, where \( k \) is successful or grant instance [7].

It follows probability mass function to discrete and for each service component $S_{h,2,3,...,N}$ the cumulative distribution function could be expressed as:

$$ P \left( \bigcup_{i=1}^{N} S_i \right) $$
\[
= \sum_{i} P(S_i) - \sum_{i} \sum_{j<i} P(S_i S_j) + \ldots
\]
\[
\ldots + (-1)^{i+1} \sum_{j<i} \sum_{k<j} P(S_i S_j \ldots S_k) + \ldots
\]
\[
\ldots + (-1)^{N+1} \sum_{j<i} \sum_{k<j} \sum_{l<k} \ldots \sum_{z<l} P(S_i S_j \ldots S_k \ldots S_z)
\]
\[
(1)
\]
Now, \( S_i \) will invoke 'n' resources collectively and that can be with probability of
\[
\frac{(N-1)^n}{N}
\]
Therefore, \( S_i \) will consume/ utilize 'n' resources collectively and that can be with probability of
\[
P(S_i) = \left( \frac{N-1}{N} \right)^n
\]
Also, for service components \( S_i S_j \) will utilize avail set of resources and that can be with probability of
\[
P(S_i S_j) = \left( \frac{N-2}{N} \right)^n
\]
Same could be possible for,
\[
P(S_i S_j \ldots S_k) = \left( \frac{N-k}{N} \right)^n
\]
For \( N^m \) term,
For \( R > n \) After substituting above values in Equation 1, we obtained
\[
= \sum_{i} P\left( \bigcup_{j=1}^{N} S_j \right)
\]
\[
= N \left( \frac{N-1}{N} \right)^n - \frac{N}{2} \left( \frac{N-2}{N} \right)^n + \frac{N}{3} \left( \frac{N-3}{N} \right)^n - \ldots
\]
\[
\ldots + (-1)^N \left( \frac{N}{N-N-1} \right) \left( \frac{1}{N} \right)^n
\]
\[
= \sum_{i=0}^{N} \binom{N}{i} \left( \frac{N-i}{N} \right)^n (-1)^{i+1}
\]
\[
(2)
\]
Now, each service component \( S_i \) where \( i = 1, 2, \ldots, N \) can also integrate with other service components \[15\]. The above probability can help us to determine, if 'n' number of resources are being fixed and random variable 'R' for any given set of devices in IoT Platform. This random variable could provide – how each service can utilize or hold distinct type of resources in IoT devices for which available set of resources is supported by cumulative distribution function. Since one must utilize at least \( N \) resources to obtain a complete set that must follow:
\[
P\{ R > n \} = 1 \text{ if } n > N
\]
Therefore, from equation 2, we can obtain interesting possible set of resources:
\[
= \sum_{i=0}^{N} \binom{N}{i} \left( \frac{N-i}{N} \right)^n (-1)^{i+1} = 1
\]
This can be re-written as:
\[
= \sum_{i=0}^{N} \binom{N}{i} \left( \frac{N-i}{N} \right)^n (-1)^{i+1} = 0 \text{ for } 1 \leq n < N
\]
\[
\text{NOTE: By Binomial theorem, the probability sum is 1, i.e.}
\]
\[
= \sum_{i=0}^{N} \binom{N}{i} (p)^i (1-p)^{n-i} = [p+(1-p)]^n = 1
\]
\[
(4)
\]
If \( R \) is random variable with \( \left( n, p \right) \) as parameters whereas \( 0 < p < 1 \), then as 'i' goes from \( 0 \) to \( N \), \( \left( R = i \right) \), first increase monotonically and then decrease monotonically reaching its largest value in case where 'i' largest integer for \( i \leq (N+1)p \). The above preposition has been proved by several mathematicians.
Above mathematical expression could provide necessary information on – how particular service \( S_i \) can obtain complete set of resources/devices in IoT through its components \( S_i S_j \ldots S_k \). This brief about the probability of particular service can hold distinct and critical resources and how these resources could be allocated and utilized between multiple service components \[24\]. This gives concept of Cone Model that one service via its components could share its available set of resources collectively and that can be with probability of \( \left( N-i \right)^n \) and \( j = N-i \)

\[
(3)
\]
A. EXPECTATION IN RESOURCE HANDLING AND SERVICE ACCESSIBILITY

This mathematical expression helps us to calculate estimated number of IoT resources/devices that could be stocked within enterprise organization for smooth functioning \[29, 32\]. This could be done with help of “Expectation or Expected Value” in probability \[7\].
Definition: Suppose \( X \) is discrete random variable with probability mass function \( p(x) \), then expected value or expectation of \( X \):

\[
E[X] = \sum_{x \in p(x)} x \cdot p(x)
\]

It could be defined as weighted average of period value for \( X \).

\[
p(0) = \frac{1}{2} = p(1)
\]

Then,

\[
E[X] = 0\left(\frac{1}{2}\right) + 1\left(\frac{1}{2}\right) = \frac{1}{2}
\]

We could formulate mathematical model for deriving estimated number of resources of particular type that yield highest performance probability. Here, number of resources units is random variable having probability mass function \( p(i) ; i \geq 0 \). This could also raise the demand of particular resource in purchase section for any organization in advanced. This help us to determine number of units of particular resource that could be needed in better functioning of SOA system in service accessibility and expected profit. Assume \( X \) denotes number of units of particular type of resource. If \( 'r' \) is available set of resources in stock, then profit in terms of performance efficiency, could be expressed as:

\[
p(s) = aX - (r-X)b; \quad \text{if } X \leq r
\]

\[
= ar; \quad \text{if } X > r
\]

Therefore, he expected profit may be,

\[
E[p(r)] = \sum_{i=0}^{r} [ai - (r-i)b]p(i) + \sum_{i=0}^{r} ra p(i)
\]

\[
= (a+b)\sum_{i=0}^{r} i p(i) - rb\sum_{i=0}^{r} p(i) + rb\left[1 - \sum_{i=0}^{r} p(i)\right]
\]

\[
= (a+b)\sum_{i=0}^{r} i p(i) - (a + b)r \sum_{i=0}^{r} p(i) + ra
\]

\[
= ra + (a+b)\sum_{i=0}^{r} (i-r) p(i)
\]

(5)

It could analysis the situation whenever we increase the stock \( 'r' \) in existing set of resources. This could also be done by replacing \( 'r' \) to \( (r+1) \). This help us to determine - what changes occurred into performance efficiency of overall SOA system [23]. We substitute \( r = r+1 \) in Equation 5, we obtain:

\[
E[p(r+1)] = a(r+1) + (a+b)\sum_{i=0}^{r} (i-r-1) p(i)
\]

Above equation would produce same aspect for \( i = 0 \) to \( r \).

\[
= a(r+1) + (a+b)\sum_{i=0}^{r} (i-r-1) p(i)
\]

Therefore,

\[
E[p(r+1)] - E[p(r)] = a - (a+b)\sum_{i=0}^{r} p(i)
\]

Thus, increase in existing set of resources by one would have better results as compared to current set of resources i.e. \( '(r+1)' \) would be Better than \( 'r' \) case.

**B. Therefore, Equation 6, we could derive,**

\[
\sum_{i=0}^{r} p(i) < \frac{a}{a+b}
\]

(7)

As seen from Equation 7; here, LHS is increasing in \( 'r' \), while RHS is constant [7]. So, this inequality would be satisfied for all values of \( r \leq r^* \), where \( r^* \) is higher value of \( 'r' \) satisfying above equation.

\[
E[p(0)] < \cdots < E[p(r')] < E[p(r+1)] > E[p(r+2)] > \cdots
\]

(8)

It follows that increase in existing set of resource \( (r' + 1) \) item, would generate maximize expected performance efficiency profit in SOA system.

**ALGORITHM 1**

**CALCULATING THROUGHPUT FOR WORK FLOW IN SERVICE BASED SYSTEMS**

\( S \) is complete set of service such that \( \{ s | s \text{ is atomic service}\} \)

For each \( s \in S \),

\( \alpha \) is service cost

\( R \) is rate of service request

\( A \) is % of allocated critical resources to \( s \)

Let \( P_s \) be throughput of \( s \)

If \( R \leq A/\alpha \) then \( P_s = R \)

Else \( P_s = A/\alpha \)

Throughput of workload = \( \text{Min}(s\in S \{P_s\}) \)

The above-mentioned algorithm (Algorithm 1) determines the throughput especially for atomic service which cannot be split further into sub-services and treated as single/isolated service [20]. These isolated services could form composite service that is dependable on these atomic/isolated services [26]. In this algorithm, when available set of critical resource are not bottleneck/exhaustive, then throughput \( P_s \) would be rate of service request. Otherwise, throughput \( P_s \) would be allocated resource over service cost.
The second algorithm presents the throughput relationship using Linear programming as done in [30]. This formulates the allocation of critical resources for server Sv. Here, server makes available the instance of resources sv ∈ Sv. These instances would be allocated to requested services based on priority Prsv for workflow W. This gives new throughput based on requirement TR that each instance sv could obtain with SRsv service request rate. For adding task, throughput TH must be less than SRsv; otherwise, it would create a problem of bottleneck for available resources to its critical point. To complete the overall workflow w ∈ W, each service s in sv used by w should be greater than calculated constant C. This C is determining from throughput TH based of service cost α as shown above.

ALGORITHM 2

FORMULATING A LINEAR PROGRAMMING TO ALLOCATE CRITICAL RESOURCES FOR SERVER IN SERVICE BASED SYSTEM

W is workflow in service-based system such that {w| w is workload}
Sv is available servers such that {sv| sv is server in service-based system}
For each w ∈ W
Let throughput of w be TH and Priority of w be Prw
Linear Programming Objective function
Max{ ∑ (TH × Prw)}
Let TR would be throughput-requirement of w
Let SRw is rate of service request
For Task check (TH ≤ SRw)
If (TR ≤ SRw)
add Task (TH ≥ TRw)
For each sv ∈ Sv
For each w ∈ W
If w is atomic service directed from Sv
Let C be ant constant, i.e. C=0
C = C + α × THw
ARsv is percentage (%) of available set of critical resources of sv
Add Task (C ≤ ARsv)

The third algorithm is for allocating critical resources optimally. This algorithm considered all given parameters that brief about workflow and server related issues in service-based system [27] [28]. This devises the optimal throughput for workflow which shows each service s ∈ S, must be completed by allocated Av available critical resources.

ALGORITHM 3

OPTIMAL CRITICAL RESOURCE ALLOCATION ALGORITHM OF SERVER IN SERVICE BASED SYSTEM

In finding the optimal throughput for overall workflow in service-based system
Let OTw be optimal throughput of overall workflow w ∈ W
For each s ∈ S
Let Av be available critical resource allotted to s, Av = 0
For each workflow w should be obtained s
Av = Av + α × OTw

The core algorithm is inherited from the work conducted in Ref [5] and Ref [6], that justified “Identifying and characterizing the cone model”. In this work, problem of modeling solar energetic particle (SEP) events is important to both space weather research and forecasting. Our proposed Cone Model/Inverted Cone Model also resolve the issue of service and resource provisioning for cloud based IoT Platform with Calculating the throughput for workflow in service-based systems (Algo 1) and Allocate critical resources - Optimal (Algo 2 & Algo 3).

B. PROPOSED ALGORITHM FOR CRITICAL RESOURCE ALLOCATION IN SOA SYSTEM

Based on mathematical view and previous work done in [25, 34], allocating available resource could be done with the help of this proposed cone model. This model not only determines the overload state of any SOA system, but also rectifies the provision of available critical resources. Allocation of critical resource algorithm using the cone mode is provided below:

ALGORITHM 4

ALLOCATION OF CRITICAL RESOURCE USING CONE MODEL IN SOA SYSTEM

W is workflow in SOA system such that {w| w is workload}
Sv is available servers such that {sv| sv is server instances in SOA system}
A, is available set of critical resources and O, is obtained set of critical resources by s ∈ S
Let $SR_w$ rate of service request
$\alpha$ is incurring service cost to make available resources
For each instance of server $sv \in SV$
For adding new task, check $(SR_w \leq A_s)$
If $w$ has already occupied $sv$ instances of server
Then $A_s = A_s - O_s$ and increment the associated flag for that service $s$
Repeat until each $w \in W$
If $(O_s \geq A_s)$
Then, % of available critical resources gets overloaded
Else
Critical resources could be allotted to $s$ for complete its execution
For each atomic service $s$ in $sv$ used by $w$
Rise value of flag per each allocation of instance of server $sv$ to $s \in S$
Build formation of Cone with this flag indication $F_c$
If $(F_c \leq A_s)$
Then, SOA system would be efficient enough to handle service request $SR_w$
Else
SOA system may be overloaded at server side

The above proposed algorithm would be best to handle available critical resources in SOA system. With use of flag for each service $s \in S$, specifies the obtained set of resource $O_s$ and available set of resources $A_s$. If percentage of available set of resources get drained out, i.e. $(O_s \geq A_s)$ then new task assignment must be in waited queue/postponed for some time. This shows that if flag counter for each atomic service $s$ went up to a permissible limit which is bottleneck condition for particular resource, flag must be set to its critical point, which is critical resource valuation to those particular resources. This condition must be check $(F_c \leq A_s)$ for each time allotment of instance of server resources to $s \in S$.

V. RESULTS AND ANALYSIS
This section comprises experimental study of services with its associated resources that are engaged by multiple services components. In this respect, cloud computing system is best to experiment resources utilization and sharing policies with client-server methodology [25]. This involves data centers (DC) in different region of world and user base (UB). DC contains all infrastructural resources like number of processors, memory, storage space, available bandwidth and virtualization policy. UB consists of fields like number of requester/clients, peak hours etc. DC contains Linux-based OS and “Xen” as Virtual Machine Model, which helps to establish virtualization in resources for being shared.

This involves the auto-loading and load balancing between multiple DC. The corresponding plots (as Fig.6 – Fig.11) justifies the increase in the demand from specific UB on different DCs could be standard by adopting the cone model.
This experimental setup involves seven datacenters (#DC) and five user bases (#UB). We setup 3 DCs at Asia region, single DC at each continent. We experimented with a variable number of requests in a range of 10,000 - 1,00,000 for each UB. This simulation is run for 10 hours.

Fig. 6, 7 & 8 shows almost same number of requests per hour as it placed in Asia region and all these DCs have similar number of resources. Maximum number of requests per hour is 40,000 in all three DCs. These requests have generated by its corresponding UBs in that region, leading to these DCs having similar loading metrics. The increase in the demand from specific UB on different DCs could be standard by adopting the cone model. The shape of cone in above figure justifies the requirement that each resource had been handled with proposed algorithms discussed in sub-section 4 (B).

Fig. 9 & 10 demonstrate the loading for data centers #DC04 and #DC07 respectively that are placed in a region, where number of users is high, but a smaller number of resources. In this scenario, DC is highly loaded and as a consequence, has a large number of requests per hour (reaching a maximum of 60,000 requests per hour).

Fig. 11 illustrates the loading scenario for DC#06, which is place at a region, where users are less and have higher number of resources. This would generate a smaller number of requests per hour (reaching a maximum of 30,000 requests per hour). This is just reverse of previous case that explained in Fig. 9 & 10.

The above explained result shows the compatibility of cone model on resource allocation and utilization perspectives. As number of user request increases, the load on data centers also got increased and that in turn would also raise the situation critical resources allocation as discussed in algorithm 2 and 3. It brings out important aspect of service integration along with critical resource allocation algorithm in SOA system. This would create logical cone shape in perception to handling and allocation of resources by multiple service components as depicted in Fig. 2.

Here, shape of cone, i.e., logical & virtualized, could brief about the throughput for SOA system and also discuss the allocation scheme with resources through multiple service components. Therefore, the changes in shape of cone depends on the rate of service requests. This relationship can also deduce as sharper the cone shape, larger would be its performance and on the other hand, larger the cone shape, lower its performance. Hence, if service requester rate is more, then it would be more requirement related to resource
allocation. This allocation may result in adverse or better performance of system according to situations.

There have been certain limitations that can be considered while adopting this cone model for resource provisioning. While handling different type of resources from different data centers, there has been a situation which arises due to lack of availability or denial of services associated to resource instance allocation or assignment. This cone model has such limitation based on availability of resource instances.

VI. CONCLUSIONS

This paper discusses the role of SOA with its logical view reference architecture and presents the various challenges for resource allocation in SOA system especially for IoT platform. Also, focuses on the need of client over runtime and different workloads with its extensive survey to Internet of Things (IoT). This paper proposes the cone model that helps in efficient handling and provisioning of resources in computing over IoT. Also, deduce the mathematical implication of cone model in resource provisioning and service accessibility with IoT adopted system. It also discusses the algorithms for calculating throughput and formulating optimization way to allocate critical resources in service-based system for IoT devices.

For simulation purpose, this paper experiments the proposed algorithm based on Cone Model on different datacenter resources with variable number of service requests at variable rate. With an increase in the number of resources using its instances with particular DC and client request, there would be moderate growth in requests per hour using the proposed model (which is 32,000 as average service request rate in existing work). This shows system performance could be above than moderate. Similarly, situations where the number of resources within particular DC is large and generated client requests are less in number, system performance could be high with less resources allocation and utilization. Condition where, client requests are higher in number along with lesser number of resources, system performance is degradable with excess utilization of resources.

ACKNOWLEDGMENT

This work is supported by Taif University Researchers Supporting Project TURSP 2020/345, Taif University, Saudi Arabia.

REFERENCES

[1] G. Schmalzhauf, Resource Management in SOA Environments with Adaptive Computing Virtualization. SAP Community Network, 2008.

[2] N. Niknejad, W. Ismail, I. Ghani, B. Nazari, and M. Bahari, “Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation”, Information Systems, Vol. 91, 101491, 2020.

[3] A. Bamhdi, “Requirements capture and comparative analysis of open source versus proprietary service-oriented architecture”, Computer Standards & Interfaces, Vol. 74, 103468, 2021.

[4] H. Xie, L. Ofman, and G. Lawrence, “Cone model for halo CMEs: Application to space weather forecasting,” Journal of Geophysical Research, Vol. 109, 2004, doi.org/10.1029/2003JA010226

[5] J.G. Luhmann, S.A. Ledvina, D. Odstrcil, M.J. Owens, X. P. Zhao, Yang Liu, and Pete Riley, “Cone model-based SEP event calculations for applications to multipoint observations,” Advances in Space Research, Vol. 46, pp. 1-21, 2010.

[6] The National Aeronautics and Space Administration (NASA) USA, Model Rocket Engine Performance. Retrieved May 16, 2022 from http://www.grc.nasa.gov/WWW/k-12/rocket/rktengperf.html

[7] S. Ross, “A first course in probability,” Pearson Education, USA, 2002.

[8] X. Liu, S. Wang, Y. Ma, Y. Zhang, Q. Mei, and Y. Liu, “Operating Systems for Resource-adaptive Intelligent Software: Challenges and Opportunities,” ACM Transactions on Internet Technology, vol. 21, no. 2, pp. 1-19, 2021.

[9] O. Ascigil, A. Tasiopoulos, T. K. Phan, V. Sourlas, I. Psaras, and G. Pavlou, “Resource provisioning and allocation in function-as-a-service edge-clouds,” IEEE Trans. on Services Computing. https://doi.org/10.1109/TSC.2021.3052139, 2021.

[10] A. Y. Nikravesh, S. A. Ajila, and C. H. Lung, “An autonomic prediction suite for cloud resource provisioning,” Journal of Cloud Computing, vol. 6, no. 1, pp 1-20, 2017.

[11] N. D. Nguyen, L. Pihan, D. Park, S. Kim, and T. Kim, “Elastic fog: elastic resource provisioning in container-based fog computing,” IEEE Access, vol. 8, 183879-183890, 2020.

[12] F. H. Khoso, A. Lakan, A. A. Arain, M. A. Soomro, S. Z. Nizamani, and Ke. Kanwar, “A microservice-based system for industrial internet of things in fog-cloud assisted network,” Engineering, Technology & Applied Science Research, vol. 11, no. 2, pp. 7029-7032, 2021.

[13] X. Cao, G. Tang, D. Guo, Y. Li, and W. Zhang, “Edge federation: towards an integrated service provisioning model,” IEEE/ACM Transactions on Networking, vol. 28, no. 3, pp. 1116-1129, 2020.

[14] L. Xunyun, and R. Buyya, “Resource management and scheduling in distributed stream processing systems: a taxonomy, review, and future directions,” ACM Computing Surveys, vol. 53, no. 3, pp. 1-41, 2020.

[15] R. Singh, M. Singh, S. Garg, I. Perl, O. Kalyonova, and A. Penskoi, “Multi-criteria decision making based optimal virtual machine selection technique for smart cloud environment,” Journal of Ambient Intelligence and Smart Environments, vol. 13, no. 3, pp. 185-199, 2021.

[16] M. M. Badawy, Z. H. Ali, and H. A. Ali, “QoS provisioning framework for service-oriented internet of things (IoT),” Cluster Computing, vol. 23, no. 2, pp. 575-591, 2020.

[17] X. Li, and L. Xu, “A review of internet of things: resource allocation,” IEEE Internet of Things Journal, vol. 8, no. 11, pp. 8657 – 8666, 2021.

[18] J. Santos, T. Wauters, B. Volckaert, and F. D. Turck, “Towards end-to-end resource provisioning in fog Computing over low power wide area networks,” Journal of Network and Computer Applications, vol. 175, 102915, 2021.

[19] S. Venugopali, X. Chu, and R. Buyya, “A negotiation mechanism for advance resource reservation using the alternate offers protocol,” In Proc. of 16th International Workshop on Quality of Service (IWQoS), pp. 40-49, 2008.
[20] B. Wang, C. Wang, Y. Song, J. Cao, X. Cui, and L. Zhang, “A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds,” Cluster Computing, vol. 23, no. 4, pp. 2809-2834, 2020.

[21] E. Masoumeh, M. Gholabaei-Arani, and A. Shahidinejad, “Resource provisioning for IoT services in the fog computing environment: An autonomic approach,” Computer Communications vol. 161, pp. 109-131, 2020.

[22] S. S. Yau, and Ho G An, “Adaptive resource allocation for service-based systems,” International Journal of Software and Informatics, vol. 3, no. 4, pp. 483-499, 2009.

[23] C. Tsai, “SEIRA: an effective algorithm for IoT resource allocation problem,” Computer Communications, vol. 119, pp. 156-166, 2018

[24] M. Hammoudeh, G. Epiphaniou, S. Belguit, D. Unal, B. Adebisi, T. Baker and P. Watters, “A service-oriented approach for sensing in the Internet of Things: intelligent transportation systems and privacy use cases,” IEEE Sensors Journal, Vol. 21, No. 14, pp.15763-15761, 2020.

[25] F. Bowen, X. Zhu, D. Liu, J. Chen, W. Bao, and L. Liu, “Elastic resource provisioning using data clustering in cloud service platform,” IEEE Transactions on Services Computing, doi.org/10.1109/TSC.2020.3002755, 2020

[26] M. P. Yadav, N. Pal, and D. K. Yadav, “Resource provisioning for containerized applications,” Cluster Computing, pp. 1-22, https://doi.org/10.1007/s10586-021-03293-5, 2021

[27] A. Mijuskovic, A. Chiumento, R. Bemthuis, A. Aldea, and P. Havinga, “Resource management techniques for cloud/fog and edge computing: an evaluation framework and classification,” Sensors, vol.21, no.5, p.1832, 2021.

[28] S. Chalilla, F. Korte, J. Erbel, F. Zallia, J. Grabowski, and P. Merle, “Model-based cloud resource provisioning with TOSCA and OCCI,” Software and Systems Modeling, Vol. 20, pp. 1609-1631, 2021.

[29] Z. Liqing B. Cheng, and J. Chen, “Service-oriented IoT resources access and provisioning framework for IoT context-aware environment,” In Proc. of IEEE World Congress on Services, pp.245-251, 2020

[30] X. Li, and L. D. Xu, “A review of internet of things—resource allocation,” IEEE Internet of Things Journal, Vol.8, No. 11, pp.8657-66, 2020.

[31] V. Conti, C. Miitello, L. Rundo, and S. Vitabile, “A novel bio-inspired approach for high-performance management in service-oriented networks,” IEEE Transactions on Emerging Topics in Computing, https://doi.org/10.1109/TETC.2020.3018312, 2020

[32] S. Bansal, and D. Kumar, “IoT ecosystem: a survey on devices, gateways, operating systems, middleware and communication,” International Journal of Wireless Information Networks, Vol. 27, No.3 pp.340-364, 2020.

[33] A. K. Tripathi, S. Agrawal, and R. D. Gupta, “Cloud enabled SDI architecture: a review,” Earth Science Informatics, vol. 13, no. 2, pp. 211-231, 2020.

[34] Z. Zhou, S. Yu, W. Chen, and Xu Chen, “CE-IoT: cost-effective cloud-edge resource provisioning for heterogeneous IoT applications,” IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8660-8614, 2020.

[35] S. Deng, Z. Xiang, P. Zhao, J. Taheri, H. Gao, J. Yin, and A. Y. Zomaya, “Dynamical resource allocation in edge for trustable internet-of-things systems: a reinforcement learning method,” IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6103-6113, 2020.

[36] R. S. Bhadoria, N S. Chaudhari, and G. S. Tomar, “The performance metric for enterprise service bus (ESB) in SOA system: theoretical underpinnings and empirical illustrations for information processing, Information Systems,” vol. 65, pp. 158-171, 2017.

[37] R. Bavaresco, J. Barbosa, H. Vianna, P. Blütenbender, and L. Dias, “Design and evaluation of a context-aware model based on psychophysiology,” Computer Methods and Programs in Biomedicine, p. 189, 105299, 2020.

[38] S. Chauhan, M. Mittal, M. Woźniak, S. Gupta, and R. P. Prado, “A technology acceptance model-based analytics for online mobile games using machine learning techniques symmetry,” vol. 13 no. 8, p. 1545, 2021.

**BIographies**

Robin Singh Bhadoria (M’12) is currently working as Assistant Professor in the Department of Computer Engineering & Applications at GLA University, Mathura, India. He has been obtained his Ph.D. degree in the month of January 2018 from Discipline of Computer Science & Engineering at Indian Institute of Technology (IIT) Indore, Madhya Pradesh, India. He also completed his Master of Technology (M. Tech) and Bachelor of Engineering (B. Eng.) in Computer Science & Engineering from Rajiv Gandhi Technological University (RGTU) of Madhya Pradesh, India in 2011 and 2008 respectively. He has been awarded with Gold Medal in M. Tech degree program at state level. His current area of interests are: Big Data Analytics, Service-Oriented Architecture, Internet of Things and Wireless Sensor Network.

Hatem Ghazi Zaini received the bachelor's degree/Diploma degree in electrical and computer engineering from the Faculty of Engineering, Umm Al-Qura University, Mecca, Saudi Arabia, in 1993, and the master's and Ph.D. degrees from the Florida Institute of Technology, Melbourne, FL, USA, in 1998 and 2003, respectively. In 2005, he joined the Faculty of Applied Computer Sciences, Umm Al-Qura University, as an Assistant Professor. He became a Supervisor of Cisco Networking Academy, in 2007, and a member of the College Committee of Quality Control and Quality Assurance the College Higher Committee of Consultation at Technical College at Makkah, in 2008, and the Department of Computer Technology, Saudi Arabia, in 2009. From June 2011 to January 2013, he became an Assistant Professor and an Associate Professor with the Department of Electrical Engineering, Taif University, Saudi Arabia. His research interests include programming, computation algorithms, and real time control systems.

Md. Manzar Nezami was born in Samastipur, Bihar, India, in 1984. He received the B. Sc. Engineering degree in Electronics and Communication from Magadh University, Bodh Gaya, India, in 2009. He has been teaching in the Department of Electronics and Communication Engineering at Siwan Engineering and Technical Institute, J. P. University, Chhapra from 2009 to 2011. He received M.E and Ph.D. degree in Instrumentation and Control Engineering from National Institute of Technical Teachers’ Training and Research, Chandigarh and Jamia Millia Islamia (A Central University), New Delhi, India, in 2014 and 2020, respectively. Currently, he is working with the Department of Electronics and Communication Engineering at GLA University, Mathura, India. His area of research includes condition monitoring of transformers' insulation, thin film sensor fabrication with their signal conditioning circuits, and application of soft-computing techniques.

Md. Danish Equbal received his B. Tech degree in Electrical Engineering and M. Tech in Power System from Jamia Millia Islamia, New Delhi, India, in 2006 and 2009, respectively. He also earned a Ph.D. in Electrical Engineering from Jamia Millia Islamia in 2019. Currently, he is an Associate Professor in Electrical Engineering Department at Galgotias College of Engineering and Technology, Greater Noida, India. His area of research includes...
insulation condition monitoring using Dissolved Gas Analysis (DGA) of transformer oil and application of soft-computing techniques.

**Elfatih Elmubarak Mustafa** received his bachelor Bachelor’s degree from Sudan University of Science and Technology in 2003. He received Master’s degree from Al-Neelain University-College of Computer Science and Information Technology in 2008 and Ph.D. degree from Al-Neelain University-College of Graduate Studies in 2013. His major research area includes E-government’s applications, Artificial Intelligence and Big Data.

**Sherif S. M. Ghoneim** (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Faculty of Engineering at Shoubra, Zagazig University, Egypt, in 1994 and 2000, respectively, and the Ph.D. degree in electrical power and machines from the Faculty of Engineering, Cairo University, in 2008. Since 1996, he has been teaching at the Faculty of Industrial Education, Suez Canal University, Egypt. From 2005 to 2007, he was a Guest Researcher with the Institute of Energy Transport and Storage (ETS), University of Duisburg, Essen, Germany. After that, he joined the Department of Electrical Engineering, Faculty of Engineering, Taif University, as an Associate Professor. His research interests include grounding systems, dissolved gas analysis, breakdown in SF6 gas, and AI technique applications.