A Novel Multi-Objective Optimization Based Evolutionary Algorithm for Optimize the Services of Internet of Everything

SHAILENDRA PRATAP SINGH1, Gaurav Dhimani2,3,4, (Senior Member, IEEE), Wattana Viriyasitavat6, (Senior Member, IEEE), AND SANDEEP KAUTISH7

1School of Computing Science and Engineering, Galgotias University, Greater Noida 203201, India
2Department of Electrical and Computer Engineering, Lebanese American University, Byblos 1701, Lebanon
3Centre for Research and Development, Department of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali 140413, India
4Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248002, India
5Department of Project Management, Universidad Internacional Iberoamericana, Campeche 24560, Mexico
6Business Information Technology Division, Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University, Bangkok 10330, Thailand
7Lord Buddha Education Foundation (LBEF) Campus, Kathmandu 44600, Nepal

Corresponding author: Sandeep Kautish (dr.skautish@gmail.com)

ABSTRACT In the new era, the Internet of Everything (IoE) provides distributed services like data, processes, people, and things, etc. The services to the connected IoE significantly increase the time of service, workload, energy consumption, and delay. These objectives conflict with each other. To address the issue, a novel multi-objective based evolutionary algorithm is proposed. In the proposed method, a new rapid mutation operator is incorporated with multi-objective differential evolution (MODE) to overcome the stagnation of the local optimum. The proposed method to maintain the diversity and enhance the convergence speed of the existing MODE algorithm is described. The proposed method provides more diversity and convergence speed for choosing better candidate solutions. The addition of the proposed method is evaluated with the application of IoE services. We have designed the two objective and three objective-based IoE services scenarios. Furthermore, the proposed method optimizes services like service cost, service delay, and the lifetime of sensors. It is interesting to observe that the proposed approach better performs the most recent state-of-the-art multiobjective evolutionary algorithms.

INDEX TERMS Adaptation, differential evolution, multi-objective evolutionary algorithms, Internet of Everything.
particularly in sensor nodes. In this research, we discuss methods for energy optimization in sensor objects based on multi-objective Differential evolution (DE) algorithms.

DE was developed by Storn and Price [1], [2]. It is one of the stochastic population-based optimization algorithms and has proven to be the most promising algorithm to capture the approximate solution of NP-Complete problems [3], [4], [5], [6]. In the multi-objective approach, the author [8] addresses numerous objectives where all the objectives conflict with each other. Therefore, the nature of the multi-objective optimization problem is different as compared to the single-objective optimization. The description of a multi-objective problem using the mathematical expression is as follows.

\[
\min / \max f(Q) = (f(q_1), f(q_2), \ldots , f_m(q_n))^T \\
G_i(q) \leq I = 1, 2, \ldots \ldots , k \\
q \in \omega
\]

where, \( Q = (q_1, q_2, \ldots , q_n)^T \) denoted the decision variable of vector, \( \omega \) is denote the decision space, and \( n \) denote the number of variable. Therefore, \( f(Q) \) is a function of multiobjective problems (MOPs), which contains \( m \) objective functions. The MOPs for two or more objective functions’ mathematical model the Pareto-based approaches produce superior optimum solutions for solving objective functions. This approach uses the ranking-based optimum solution finding nondominated sorting algorithm.

Although the authors provided a wide range of cutting-edge evolutionary algorithms, multi-objective evolutionary algorithms [7], [8], [9] are more prevalent, straightforward, and have a low convergence rate. The basic goal of the mutation operator in evolutionary algorithms is to keep the solutions diverse. The literature [3], [4], [5], [6], [7], [8] shows that the diversity provided by the existing mutation operator is insufficient. It is essential to include a better modified mutation operator that produces more varied solutions in order to address the problem of stagnation in multi-objective optimization.

**A. HIGHLIGHTS OF AUTHOR’s CONTRIBUTIONS**

Highlights of author’s contribution are as follows.

- A MODE-based non-dominated algorithm with a new rapid adaption-based mutation strategy is proposed to address the various problems such as stagnation in local optima, diversity maintenance, and convergence speed in contrast to the existing DE algorithms, which suffer from all these issues [1], [2], [3], [4], [5], [6], [7], [8], [9].
- In contrast to most of the existing strategies, which are based on non-dominating sorting and increase the complexity of multi-objective optimization algorithms [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], the suggested method is based on the Pareto strategy, which minimises the energy consumption, delay, and service load.
- This proposed approach, as an outcome, diversity is retained from the initial generations and maintained till the end.
- It applies this algorithm to the application of two objective and three objective-based IoE services scenarios framework.
- The proposed method optimizes services like service cost, service delay, and the lifetime of sensors.

**B. ARTICLE ORGANIZATION**

The rest of this paper is organised as follows. In Sect. II, related work based on multi-objective evolutionary algorithms and the Internet of Things is presented. In Sect. III, the application of IoE related to the services model is provided. In Sec. IV, the proposed multi-objective differential evolution method and IoE services-based algorithm have been discussed. In Sect. V, the obtained outcome of the experimental results and analysis is presented. Finally, in Sect. VI, concludes the study and paves the way for future research direction.

**II. RELATED WORK**

In this section, a literature review on multi-objective evolutionary algorithms is explained in subsection 1, and IoT-based models and evolutionary algorithms in subsection 2.

1) LITERATURE REVIEW ON RECENT ADVANCES ON EVOLUTIONARY ALGORITHM

The evolutionary algorithm is intended to tackle the global optimization problem. It is clear from the literature that DE has a problem with stagnation (solutions produced can fall due to the short diversity of solutions generated in the local optimum). Various multi-objective differential evolutionary (MODE) algorithms and Pareto-based techniques have been proposed to address these challenges [3], [4], [5], [6], [7], [8], [9]. The authors [6] developed a novel Pareto-based differential evolution (PBDE) technique for multi-objective problems. The authors introduced a MODE-based method [7], which implements mutation and crossover based on the parent population. Non-dominating sorting was also employed to lessen the time complexity. The variable parameters were tuned using this algorithm, which was based on Pareto and the diversity of the most recent solution. In [8], the authors proposed the concept of non-dominating sorting for genetic algorithms called NSGA-II. However, this approach was found to be more time-consuming (due to complexity). This issue is addressed by the newly proposed NSGA-III algorithms [9]. These algorithms use a uniform distribution-based sub-population.

The authors proposed a technique to estimate the software cost [16] and utilized it to boost the diversity of NASA 93 projects by acting as a homeostatic factor [27], [28], [29]. This technique outperforms multi-objective-based software cost estimation techniques in terms of spacing, generation distance, and inverted generation distance. The writers [18], [19] proposed a hybrid method for various applications, and
it managed two separate operators. For many objective tasks, this operator increases the diversity of the hybrid method. This technique outperforms several previous multi-objective algorithms [30], [31].

2) LITERATURE REVIEW ON INTERNET OF THINGS
Researchers proposed bi-objective based optimization with energy for IoT services in [12]. The proposed method provides the service for the distributed system by introducing a novel concept of reasonable time to find the request and response. In addition, it adopts the concept of reducing service costs and service time. But, the accuracy of the information issue was not resolved when tested on the IoT service [32], [33], [34].

In [13], the authors proposed the QoS service for computing environments. In this paper, providing services for scalable bandwidth allocation, availability, and reliability was suggested, which is known as a service environment. However, still, the actual resource-constrained issue was not resolved by researchers. In [14], the authors present a survey of different IoT scenarios from recent years. They also suggested selection criteria for the areas like health, agriculture, weather forecasting, etc. to solve the problem. In [15], the authors proposed dynamic resource management (DRM) for the IoT. This paper discusses DRM in the Internet of Things for real-world applications. Researchers proposed a biologically inspired approach with resource allocation for wireless network-based feature perception in [16]. This method provides the means for achieving the effect of the IoT framework by introducing a novel concept of reasonable allocation for network resources. In [17], researchers have proposed an efficient data scheduling scheme for the smart city. This approach reduces the waiting time and also reduces the packets for the IoT framework. In the experimental analysis, it was observed that the proposed method for emergency services in the smart city was viable. In [19], the author proposed a dynamic multi-objective-based Ant colony algorithm for railway rescheduling problems [35], [36]. This paper solved the multiobjective problems using a novel Ant colony algorithm for the real application of the railway. For reducing test time, the author of [21] suggested an AI-based test scheduling solution. This method extended the Ant colony algorithm for system-on-chip scheduling issues [22], [23]. Using a cutting-edge Ant colony method, this paper’s scheduling problem of minimizing test time was resolved [24].

III. APPLICATION OF INTERNET OF EVERYTHING
In this section, we first describe the proposed layered stack of the Internet of Everything model and the next Internet of Everything service framework [37], [38].

A. LAYERED STACK OF INTERNET OF EVERYTHING MODEL
This section presents the layered stack of modern IOE, and it is shown in Figure 1. This architecture is proposed by considering the quality of service (QoS) management of different components rather than physical communication between the devices. In modern applications, multiple service requests by an individual or independent system are generated and need to be responded to automatically with minimal interaction from a human. The top layer of this architecture is responsible for providing smart coordination and collaboration among the set of applications and services [39], [40], [41], [42].

The IOE management layer is responsible for adequately mapping, analyzing, and processing service requests before execution. Thus, this layer performs object virtualization, service request management, creation, and implementation. The QoS is primarily maintained by applying different approaches like optimization, observation, translation, coordination, etc. The cloud/middle layer is used to store the data generated from different sources. This layer performs data management, cleaning, transformation, and analysis to provide efficient services. The perception and communication layer is responsible for establishing the network and physical communication among the devices.

Figure 2 represents the proposed IoE service framework. Further, The proposed framework is used to provide services from smart health sector, which is shown in figure 3(a) and figure 3(b). Figure 3(a), IoT services enable sensors, storage devices, IoT devices, and users to interact with each other. This information is used by a huge number of sensing devices that communicate information through the Internet connection or cloud, as shown in figure 3(b). Therefore, the proposed algorithm applies to the IoT service for the optimization of its energy consumption, workload, delay, and service cost. IoT is also a multiobjective problem where we have to optimize various constraints like IoT services. In the next subsection, we will discuss the IoT service model.

B. IoE SERVICE FRAMEWORK MODEL
We have designed the service framework according to the request generated in the IoE service framework graph $G = (U, V)$, and represented five-tuple, represented as $U$, which
is Eq.4 as follows [4]:

\[ U = (U_{ID}, \ U_{Type}, \ U_{Wload}, \ U_{SD}) \] (4)

where \( U_{ID} \) denotes the identification of the service request. The \( U_{Type} \) denotes many different types of objects, like sensing devices that initiate the request. \( U_{loc} \) denotes the geographic coordinate location (150, 150) of the service request. And \( U_{Wload} \) denotes workload, and \( U_{SD} \) denotes the service data, which is gathering information from different sensors. A response message designated as Y and its expression Eq.5 are produced when a service request is approved by the service provider.

\[ V = (V_{ID}, \ V_{Type}, \ K, \ AVL, \ V_{loc}) \] (5)

where \( V_{ID} \) denotes the service provider. \( V_{Type} \) denotes the type of service. \( K \) denotes the usage status of the service provider. \( L \) denotes the unit energy consumption. \( AVL \) denotes the serviceability of the different devices or objects. \( S_{loc} \) denotes the geographical coordinates (150, 150) of the service.

C. OBJECTIVES FUNCTION OF IoE SERVICES MODEL

In this section, there are four types of performance objectives that are generally used for the IoE, i.e. service cost, energy consumption (energy loss), workload (load), and information request and response \((U_{req}, V_{res})\) metrics. These metrics are frequently used to evaluate how well the suggested algorithm works. These metrics’ values are computed via simulation work, and the IoE framework model’s QoS details are given below:

1) SERVICE COST

The service coordinates \( G(U, V) \) of the information response change with \( t \) in this scenario. This framework represents the information request \( U_i \) and data response \( V_i \). Therefore, information transmitted between \( U_i \) and \( V_i \) is represented by Dist\((U_i, V_i)\) in figure 3. The first objective (IoE1) in terms of cost is calculated by using the following formula Eq.6:

\[ \text{Dist}(U_i, V_i) = \sqrt{(U_i - U_j)^2 + (V_i - V_j)^2} \] (6)

2) ENERGY CONSUMPTION(EC)

Energy consumption is represented in the second objective by EC \((U_i, V_i)\), where \( U_i \) denotes the data request and \( V_i \) denotes the data response. The EC \((U_i, V_i)\) for responding data \( V_i \) to various information requests \((U_i)\). Therefore, the second objective (IoE2) in terms of energy consumption (Loss) is calculated as follows Eq.7:

\[ EC(U_i, V_i) = \sum_{i=1}^{t} \sum_{j=1}^{t} U_iV_j \] (7)

3) DELAY

In the third objective, our goal is to minimize the workload (load) from base station to substation. The third objective (IoE3) in terms of load is calculated as follows Eq.8:

\[ \text{Delay} = \sum_{i=1}^{t} \text{Dist}(V_i, U_i)/U \] (8)

where, \( U_i \) denoted data request and \( V_i \) denoted the data respond. \( U_i \) denoted the different requests of the IoE services.

4) LOAD

In the fourth objective, in the IoE framework, our goal is to minimize the workload (load). The fourth objective (IoE4) in terms of the load is calculated as follows Eq.9:

\[ \text{Load} = \sum_{i=1}^{t} \sum_{j=1}^{t} \text{Dist}(U_i, V_i) + EC(X_i, V_i) \] (9)

where, \( U_i \) denoted data request and \( V_i \) denoted the data respond. Where, \( U_i \) denoted data request and \( V_i \) denoted the data respond. The sensors for respond data \( V_i \) to different request information \((U_i)\) on IoE.

D. FORMULATION OF FITNESS FUNCTION OF IoE FRAMEWORK

In this section, the fitness functions of IoE services calculated by equations 6, 7, 8, and 9 are non-contradictory. As a result, using the sum of weighted approach as stated in Eq. 10, all of the objectives (IoE1, IoE2, IoE3, & IoE4) are turned into a single objective function.

\[ \text{Fitness} = f_{v_1} \times \text{IoE1} + f_{v_2} \times \text{IoE2} + f_{v_3} \times \text{IoE3} + f_{v_4} \times \text{IoE4} \] (10)
Here, values of \( f_1, f_2, f_3, \) & \( f_4 \) are the weights that are assigned to each of the objective function. The weight factor is an important performance parameter from one objective to another in multi-objective problems. A random weight value between 0 and 1 is added to every objective function in order to drive the convergence of the Pareto optimal solutions obtained by algorithms towards the true Pareto front solutions. This advantage of the optimum convergence of the Pareto front in local search space also provides sufficient diversity from the IoT framework. To compare the IOE framework model, the fitness function used the MOPSO, ABCO, and whale optimization algorithm (MOWOA).

E. FORMULATION OF PROPOSED MODEL BASED TWO OBJECTIVE AND THREE OBJECTIVE

This section we design the two objective based services functions using IOE parameters as shown in Eq. 11.

\[
\begin{align*}
  f_1(\text{min}) &= EC(U_i^j, V_j^i) = \sum_{i=1}^{t} \sum_{j=1}^{t} U_i V_j \\
  f_2(\text{min}) &= Delay = \sum_{i=1}^{t} \text{Dist}(V_i^j, U_j^i)/U \\
  f_3(\text{min}) &= \sum_{i=1}^{t} \sum_{j=1}^{t} \text{Dist}(U_i^j, V_j^i) + EC(X_i^j, V_j^i) \quad (12)
\end{align*}
\]

Further, we created three objective-based IoE-based service estimation models that compete with one another. Using the IoE-based service parameters as stated in Eq. 12, all objectives are transformed into multi-objective functions.

\[
\begin{align*}
  f_1(\text{min}) &= \sum_{i=1}^{t} \sum_{j=1}^{t} U_i V_j \\
  f_2(\text{min}) &= \sum_{i=1}^{t} \text{Dist}(V_i^j, U_j^i)/U \\
  f_3(\text{min}) &= \sum_{i=1}^{t} \sum_{j=1}^{t} \text{Dist}(U_i^j, V_j^i) + EC(X_i^j, V_j^i) \quad (12)
\end{align*}
\]

where, \( f_1(\text{min}) \) denoted the minimization problem of IoT2, which is the measurement of calculates the energy consumption for sensors from IoT service. \( f_2(\text{min}) \) denoted the minimization problem of IoT3, which is the measurement of calculates the load for IoT service from transmitting data, and \( f_3(\text{min}) \) denoted the minimization problem of IoT4, which is denoted the delay in between data request and data respond for IoT framework.

In the next section, the proposed approach is presented which the extension of MODE algorithms. This approach applied in the application of IoE.

IV. PROPOSED METHODOLOGY: RAPID ADAPTATION BASED MUTATION APPROACH USING MULTI-OBJECTIVE DE ALGORITHM

The new variants of evolutionary algorithm must be designed and executed to IoE based framework model in the real word application. For that, this paper outlined a new rapid adaption-based operators. This operator guides the better solution from the search space. The proposed operator helps to rapid the adaptation technique used to select best vector in the global search space. The rapid adaption vector plays an essential role in enhancing the convergence speed and maintaining diversity. This vector depends on the selection of the search area and the nature of adaptability of the current environment in the system. In this paper, the proposed methodology composed of various phases, namely, (1) Population initialization, (2) Non dominating sorting, (3) Rapid based mutation operator, (4) Crossover, (5) Selection, (6) New Population generation, and (7) Proposed algorithm applied the application of IoE. The detail description of all these phases as follows.

A. PHASE 1: POPULATION INITIALIZATION

A vector’s denotes a potential solution of the search space. The dimension of the vector is problem specific and value of each unit of dimension is chosen between the upper and lower bound values according to the problem. In the proposed work, ZDT and DTLZ series test functions are considered for the outcome analysis, therefore, the dimension of a variable is equal to two for bi-objective and three for the tri-objective problem. Let, \( V_i = \{X_{i,1}(t), X_{i,2}(t), X_{i,3}(t), X_{i,4}(t), ..., X_{i,D}(t)\} \) be the \( i^{th} \) vector of the solution vectors, where each component \( X_{i,d}(t) \) is initialized using Eq. 14. Then the vector can be represented as follows:

\[
V_i = \{X_{i,1}(t), X_{i,2}(t), X_{i,3}(t), X_{i,4}(t), ..., X_{i,D}(t)\} \quad (13)
\]

Initial population is generated randomly between upper lower and upper bound

\[
X_{i,d} = X_{i,d}^L + \text{rand}(\cdot) \times (X_{i,d}^U - X_{i,d}^L) \quad (14)
\]

where, \( X_{i,d}^U \) denotes the upper bound value of variable according to the problem and \( X_{i,d}^L \) denotes the lower bound value of variable according to the problem.

1) PHASE 2: RAPID BASED NON DOMINATING PARETO FRONT ALGORITHM

This phase, rapid adaption based selection of whole population of global search space. It is choose of 25% rapid adaption based solution vectors are arrange using the proposed non-dominating ranking based Pareto front algorithm. In NSGA-II [2], as the population size increases number of Pareto fronts increases that enhance the complexity. In proposed non-dominating algorithm, complexity is not affected as the population size increases. In this algorithm, \( \text{rank} \ 1 \) is assigned to Pareto front 1(PF1) and \( \text{rank} \ 2 \) is given to Pareto front 2(PF2) solution vectors.

Illustration of rapid based Ranking Methodology:

Rapid adaption-based strategy, which is to find the best optimum front. The best optimum means selecting the first rank from a circumstance area like the global search area. Therefore, its vector is a feasible solution to the search space. So, the algorithms have to find the best rank in the global search area, which is the best optimization solution. In this proposed method, Initially, candidate solution is sorted
Algorithm 1 Ranking Based Pareto Front Estimation

Data: Rapid Random Population Vectors with Objective Functions

Result: Rank1 and Rank2 Pareto Fronts

Initialization: Empty sets PF1 and PF2

Step 1: Sort all the solution Vectors \((X_1, X_2, \ldots, X_N)\) in decreasing order of their first objective function and create a sorted list

1.1 Chooses the 25% rapid based solution Vectors (RBS)

1.2 Remaining solution (RS) of the 75% from search space

Step 2: Add first element of RBS list to PF1

Step 3: For every RS list solution, compare solution from the solutions of RBS list

3.1 If any element of set PF1 dominate, Delete from the RBS list and add to PF2

3.2 If sorted list is non dominated to set PF1, Then update set \(PF1 = PF1 U PF2\).

3.3 If set PF1 becomes empty add immediate solution at immediate solution to PF2

Step 4: Select the first Pareto front according to the non dominance sorting criteria as mentioned in Algo. 1. Thereafter, \(Rank1_i\) is assigned to Pareto front 1 candidates as depicted in Eq. 15. Similarly, \(Rank2_i\) is given to remaining as shown in Eq. 16.

\[
rank 1_i = \sum_{j=1}^{n} \text{Solution}_j * 25\% \tag{15}
\]

where, \(rank 1_i\) denotes the ranking index of candidate solutions according to rapid solution of the 25% from search space, \(i = 1, 2, \ldots n\), which is defined the rapid adaption based vector for first ranking of the rapid adaption based candidate solutions \((\text{solution}_i)\).

\[
rank 2_i = \sum_{k=1}^{n} \text{Solution}_k * 75\% \tag{16}
\]

where, \(rank 2_i\), \(\forall 1 \leq i \leq n\) denotes the remaining solutions of 75% from search space. This solution help to the comparing the index of Pareto front 1 and Pareto front 2, in given the algorithm 1. If it is not solution feasible then delete the candidate solution. Remaining candidate solution is stored in the \((\text{solution}_i)\).

1) RAPID VECTOR BASED SOLUTIONS FROM PARETO FRONT 1

First vector is framed from the \(Rank1_i\) candidate solutions by considering the best 25% of the rapid member solution as shown in Eq. 17.

\[
RV_1 = rank 1_i * \sum_{i=1}^{n} \text{Rapid}_i * RA F1(0, 1) \tag{17}
\]

where \(RV_1\) denoted the rapid adaption vectors, \(rank 1_i\) denoted the first rank Pareto front of search space, which is select to best 25% feasible solutions. The \(\text{Rapid}_i\) denoted the first rank Pareto front solution from nondomination sorting algorithm. And, \(RAF1(0, 1)\) denoted the rapid adaption factor, which is provide the sufficient diversity as well as convergence rate from search space.

2) VECTOR BASED SOLUTIONS FROM PARETO FRONT 2

Next vector is estimated using the \(Rank2_i\) candidates solutions and considering the dynamic factor \(DF_2\) as shown in Eq. 18.

\[
RV_2 = rank 2_i * \sum_{i=1}^{n} \text{Rapid}_i * RA F1(0, 1) \tag{18}
\]

where \(RV_2\) denoted the rapid adaption vectors, \(rank 2_i\) denoted the second rank Pareto front of search space, which is select to best 75% feasible solutions. The \(\text{Rapid}_i\) denoted the first rank as well as second rank Pareto front solution from nondomination sorting algorithm. And, \(RAF1(0, 1)\) denoted the rapid adaption factor, which is provide the sufficient diversity as well as convergence rate from search space.

3) FORMATION OF DONOR VECTOR

The idea of environmental optimization is taken into consideration to generate an environment optimization based mutation scheme of DE / BEST / 1. Basic mutation operator is shown in Eq. 19. This process is composed of two steps, both of them are defined as follows.

\[
\hat{y}_i,G = \bar{a}_{\text{best},G} + \delta_1 \cdot (\bar{a}_{r_1,G} - \bar{a}_{r_2,G}) \tag{19}
\]

where \(\hat{y}_i,G\) denoted the donor vector of state of the art algorithm, \(\bar{a}_{\text{best},G}\) denoted the best vector of search environment. \(\delta_1\) \(\hat{a}_{r_1,G}\) and \(\hat{a}_{r_2,G}\) denoted the target vector of given candidate solution. \(\delta_1\) denoted the mutant factor lie between \([0,2]\).

a: STEP 1

In this step, first ranking based rapid vector’s \(RV_i,G\) is generated as shown in Eq. 20.

\[
\bar{R}V_i,G = \bar{a}_{\text{best},G} + \delta_1 \cdot (\bar{R}V 1_{r_1,G} - \bar{R}V 1_{r_2,G}) \tag{20}
\]

b: STEP 2

If better solution is not obtained using Eq. 20 then Eq. 21 will be used for mutation operator.

\[
\check{R}V_i,G = \bar{a}_{\text{best},G} + \delta_1 \cdot (\check{R}V 2_{r_1,G} - \check{R}V 2_{r_2,G}) \tag{21}
\]

B. PHASE 3: RAPID ADAPTION BASED MUTATION OPERATOR

In this phase, donor vector is generated by considering the rapid adaption environment or feasible solution. The detail description is given as follows.
where, \( \vec{RV}_{i,G} \), \( 1 \leq i \leq n \) denotes the ranking based mutant vector, \( n \) is population size, \( G \) denote the generation, and \( r \) denote the index of vector’s. \( \hat{a}_{best} \) denotes the best vector of current population. \( \vec{RV}_{1,G} \) and \( \vec{RV}_{2,G} \) will be generated rapid adaption based vectors of global search. Further, it is applied the crossover rate in the next subsection.

C. PHASE 4: CROSSOVER

This section, It is applied the crossover rate random(0, 1). These random values help to improve the convergence rate of the search space. For creating the trail vector, donor vector obtained using ranking based mutation operator is mixed with the target vector. Further, it is applied the selection operator in the next subsection.

**Algorithm 2** Proposed Multi-Objective IoE Service Model

**Input:**
(i) IoE based services (i): Data Request
(ii) IoE based services (j): Data Response
(iii) X axis: Candidate solutions range according to available service
(iv) Y axis: Candidate solutions range according to available service
(v) \( t \): Number of weighted vectors for search space
(vi) \( N \) Number of candidate solutions
(vii) \( NP \): Rapid population according to IoE service

**Output:** Write here the result

1. **Step1** \( fun_{obj} \), \( obj = 1, 2, \ldots, n \). Multi-objective problem with \( obj \) is a objective
2. **Step2** Search_space of D
3. **Step3** Number of generation according to fitness functions
4. **Step3.1** Generate the initial Population as mentioned in phase 1
5. **Step3.2** Apply the non-dominated ranking based algorithm and generate their Pareto front number with assign ranks to the solutions (Please Refer to Phase 2)
6. **Step4** While(\( t = \text{Max} \))
7. **Step4.1** Evaluate the fitness value of each \( f_i \), Vector
8. **Step4.2** Evaluate the two objective based IoE
9. **Step4.3** Evaluate the three objective based IoE
10. **Step4.4** Evaluate new vector using propose mutation operator as mentioned in above section
11. **Step4.5** Apply the selection process
12. **Step4.6** go to step 3 until convergence reach
13. **Step5** End while

D. PHASE 5: SELECTION

Selection operators generally use the concept of survival of the fittest. This concept applies to select the best optimum value. But if it does not achieve optimal value, then this operator selects the original vectors. This process is explained in algorithm 2. We have applied these phases to the IoE. In the next section, we have discussed the multiobjective optimization-based IoE.

**TABLE 1. Control parameters of proposed algorithm.**

| Sr. No. | Parameter                  | Type          |
|--------|----------------------------|---------------|
| 1      | Population Size            | 100           |
| 2      | Dynamic factor1 \( [DF1] \) | random [0.01-0.1] |
| 3      | Scale factor \( \alpha \)   | [0.1-2]       |
| 4      | Crossover Rate \( [CR] \)  | [0.1-1]       |
| 5      | Dimension \( [D] \)        | [3]           |
| 6      | Function Evaluations       | PBS           |
| 7      | Number of Generations      | 100,200, and 500 |
| 8      | Search Space               | [5,5]         |
| 9      | Number of Sensors          | 100           |
| 10     | Search Space of IoE Service| [150,150]     |
| 11     | Dynamic factor2 \( [DF2] \) | random [0, 1] |

E. PHASE 7: THE PROPOSED ALGORITHM APPLIED APPLICATION OF IoE

The proposed method applies this algorithm to the IoE application for the estimation of its data during the various sensors’ data requests and responses for optimization. We have to optimise various constraint functions like \( f_1, f_2, f_3 \), and \( f_4 \). The proposed method applies the multiobjective-based six scenarios and measurement of sensor capability for communication between different objects. Furthermore, as shown in Eq. 10, we design the fitness function of the proposed framework of the IoE service. This framework has multiple requests and responds with different objects, like six scenarios. These functions provide measurement of the energy consumption, delay, and service load from two objective and three objective-based scenarios. The pseudo-code of the proposed IoE based algorithm is shown in Algorithm 2. This algorithm produced optimal solutions for the objective functions like energy loss, load, and delay, and also generate the Pareto optimal solutions of the IoE service.

V. RESULT ANALYSIS AND DISCUSSIONS

A. ANALYSIS FOR SCENARIO OF THE IoE SERVICES

In this paper, we have taken six scenario of the IoE services for the testing of our proposed IoE based MODE algorithm. These functions, sometimes referred to as ZDT (two objectives) and DTLZ (three objectives), are based on multiobjective problems (scenarios) as shown in figure 4. The different services based on service request and service respond by sensors are the foundation of the multiobjective issues. Three scenarios of the IoE services, referred to as biobjectives functions, are used to test the suggested approach. Additionally, the suggested approach is tested using three IoE service scenarios, also referred to as triobjective benchmark issues. Standard algorithms like MOWOA [10], MOPSO [11], and ABCO [19] are compared to the experimental results provided by the suggested algorithm. Table 2 provides a description of the characteristics of the objective functions.

B. EXPERIMENTAL SETUP OF THE IoE

In the application IoE-based service network, the sensors are used to detect the data are collected by sensors and
transmitted to the platform for processing. In this experimental setup, we set an IoE framework (150 × 150) in figures 3, and 4 where 100 sensors are evenly distributed, that is, service requests. Also, we have used the 100 sensors that are regarded as service providers, and they are active sensors according to request and response data from the process, People, and things. We have selected the experimental area in a matrix of 10 by 10. In this experiment, the generate randomly service requests is 50, and it has chosen to be independent into six service strategies. This services strategy name are scenarios 1, 2, 3, 4, 5, and 6 apply to the availability of the sensors, shown in figure 4. Further, the proposed method generates the solution is composed of real value and array encoded, which determine the bit of the sensors. The sensors have represented the dimensions of candidate solutions for IoE service framework.

C. COMPARISON OF FITNESS FUNCTION FOR THE PROPOSED ALGORITHM WITH STATE OF THE ART EVOLUTIONARY ALGORITHMS

The proposed method is used to improve the performance of the fitness functions as shown in Eq. 10. The proposed method

![Figure 4](image1.png)

**FIGURE 4.** Six Scenario of the IoE framework for distributed sensors.

![Figure 2](image2.png)

**TABLE 2.** Scenario of the IoE services for multiobjective functions.

| Sr. No. | Scenarios of IoE Services | IoE services Problems | IoE Domain Variable of Problems |
|--------|---------------------------|-----------------------|---------------------------------|
| 1      | Scenario 1 based on two objective function | Convex feature | ZDT1 domain variable lie between [0, 1] |
| 2      | Scenario 2 based on two objective function | Nonconvex feature | ZDT2 domain variable lie between [0, 1] |
| 3      | Scenario 3 based on two objective function | Convex disconnected feature | ZDT3 domain variable lie between [0, 1] |
| 4      | Scenario 4 based on three objective function | Linear feature | DTLZ1 domain variable lie between [0, 1] |
| 5      | Scenario 5 based on three objective function | Concave feature | DTLZ2 domain variable lie between [0, 1] |
| 6      | Scenario 6 based on three objective function | Concave feature | DTLZ3 domain variable lie between [0, 1] |

![Figure 5](image3.png)

**FIGURE 5.** IoE framework: Load computation v/s number of generations.
provides sufficient diversity from the optimal local problems. Therefore, it uses rapid adaption factors to provide the conversion speed of the proposed algorithm. The performance result is shown in figures 8 and 9.

This proposed method is incorporated into the IoE service model with an Eq. 10 to generate the value. These values are mentioned in Table 6 according to the best, average, and worst fitness functions which checks the performance of different runs like 1, 5, 10, 15, 20, 25, and 30 on the 100 generations. Table 6 show that the performance of the proposed method is better than that of other standard optimization algorithms on six scenarios. Each scenarios represents the best, average, and worst case of the fitness cost of the IOE service model respectively. Table 6 shows that the performance of
FIGURE 9. Fitness cost of the three objectives based scenarios.

The proposed method is better than that of other standard optimization algorithms like ABCO, MOPSO, and MOWOA in terms of lowest fitness cost.

In Figures 8 and 9, the X-axis represents the number of generations, and the Y-axis represents the fitness cost of the IoE-service model for two and three objective scenarios, respectively. The fitness values for enhancing the system’s service by achieving pretty encouraging performance, which shows that the suggested methodology has high convergence speed in terms of the minimum fitness cost.

D. COMPARISON OF PARETO FRONT: THE PROPOSED METHOD WITH OTHER STATE OF THE ART ALGORITHMS

We have checked the six scenarios of different multiobjective problems like the ZDT and DTLZ series. This series has six benchmark functions with IoE service problems. These problems are solved by the proposed multiobjective based DE algorithm. Further, comparing analysis of the rate of convergence, which depicts how faster an IoE service model reaches the desired value, is calculated from the Table 2, ZDT and DTLZ series functions.

The convergence speed is used for optimum fronts to compare with other sophisticated algorithms of the proposed algorithms. This technique is used to find a rapid adaptation-based strategy, which is to find the best optimum front. The best optimum means selecting the first rank from a circumstance area like the global search area. Therefore, its vector is a feasible solution to the search space. So, the algorithms have to find the best rank in the global search area, which is the best optimization solution. Further, the trade-off between energy and delay for ZDT functions is shown in figures 6. From figure 6, with a balanced Pareto front between energy and delay, it is clear that the suggested technique produces good outcomes. Also, the trade-off between energy, delay, and load DTLZ functions is shown in figure 7, which reflects that the proposed algorithm (proposed Algo) has better results with well-spread Pareto fronts. Also, the proposed tuning operator minimises the energy rate, load, and delay.

FIGURE 10. Two objective based IoE framework of the lifetime of sensors.

E. RESULTS ANALYSIS FOR THE INTERNET OF EVERYTHING

This section, the proposed method incorporated to the IoE services for estimating like the energy consumption, service load, and delay. This framework is being used by IoE services, which are optimized the various constraints for multiobjective problem and tunes all the parameters as given in Tables 1, and 2. The proposed approach is evaluated on the IoE Service for comparing the Pareto Front on two objectives like ZDT and three objectives like the DTLZ series. Further, this approach is also applied to the IoE Service for calculating the data service cost, energy loss, load, and delay. The detailed description is given as follows:

1) IoE SERVICE COMPARISON OF THE ENERGY LOSS

This section analyses the energy loss caused by the IoE service, which is produced using reference Eq. 7. It is evident from Table 3 that the suggested method outperforms state-of-the-art algorithms ABCO, MOPSO, and MOWOA in terms of sensor lifetime. The suggested strategy offers the IoE service better diversity, a higher rate of convergence, and minimal energy loss.
TABLE 3. Comparison of energy for IoE service model.

| Scenarios/Algorithm | MOPSO  | MOWOA  | ABCO   | Proposed Algo |
|---------------------|--------|--------|--------|---------------|
| Scenario 1          | 0.0192362 | 0.0173126 | 0.0167355 | 0.0150042    |
| Scenario 2          | 0.0257418 | 0.0231671 | 0.0223949 | 0.0200781    |
| Scenario 3          | 0.0854360 | 0.0768924 | 0.0743293 | 0.0666400    |
| Scenario 4          | 0.1856301 | 0.1670671 | 0.1614982 | 0.1447915    |
| Scenario 5          | 0.2470584 | 0.2223528 | 0.2149408 | 0.1927055    |
| Scenario 6          | 0.2486313 | 0.2237681 | 0.2163097 | 0.1939323    |

TABLE 4. Comparison of load for IoE service model.

| Scenarios/Algorithm | MOPSO  | MOWOA  | ABCO   | Proposed Algo |
|---------------------|--------|--------|--------|---------------|
| Scenario 1          | 0.0451763 | 0.0406587 | 0.0393034 | 0.0352375    |
| Scenario 2          | 0.4189657 | 0.3770691 | 0.3645018 | 0.3267932    |
| Scenario 3          | 0.3058467 | 0.2752627 | 0.2660867 | 0.2385606    |
| Scenario 4          | 0.5390407 | 0.4851405 | 0.4689692 | 0.4204551    |
| Scenario 5          | 0.1957413 | 0.1761672 | 0.1702949 | 0.1526782    |
| Scenario 6          | 0.2458413 | 0.2212575 | 0.2138823 | 0.1917565    |

TABLE 5. Comparison of delay for IoE service model.

| Scenarios/Algorithm | MOPSO  | MOWOA  | ABCO   | Proposed Algo |
|---------------------|--------|--------|--------|---------------|
| Scenario 1          | 0.0258647 | 0.0232783 | 0.0225022 | 0.0201744    |
| Scenario 2          | 0.0745846 | 0.0671262 | 0.0648881 | 0.0581760    |
| Scenario 3          | 0.0524617 | 0.0472155 | 0.0456416 | 0.0409201    |
| Scenario 4          | 0.683541  | 0.6151877 | 0.5946814 | 0.5331626    |
| Scenario 5          | 0.0789247 | 0.0710322 | 0.0686645 | 0.0615612    |
| Scenario 6          | 0.2486312 | 0.2237681 | 0.216309  | 0.1939323    |

2) IoE SERVICE COMPARISON OF THE SERVICE LOAD
In this part, the workload for the IoE service is analysed. The proposed method involves applying Eq. 8, it is evident that the suggested method better the state-of-the-art ABCO, MOPSO, and MOWOA algorithms, as shown in Table 4. The suggested method gives the IoE service model minimum service load values, with the results depicted in Figure 5. It is evident from Figure 5 that the suggested method yields satisfactory results; the X-axis shows the number of generations and the Y-axis shows the load computation.

3) IoE SERVICE COMPARISON OF THE DELAY
The delay of the IoE service, which is produced using Eq. 9, is analysed in this section. In Table 5, it is shown that the suggested method outperforms the state-of-the-art ABCO, MOPSO, and MOWOA algorithms by a minimum margin.

FIGURE 11. Three objective based IoE framework of the lifetime of sensors.
It is obvious that the suggested technique yields positive results.
| Sr No. | Scenario 1 | Sr No. | Scenario 2 | Sr No. | Scenario 3 | Sr No. | Scenario 4 | Sr No. | Scenario 5 | Sr No. | Scenario 6 |
|-------|------------|-------|------------|-------|------------|-------|------------|-------|------------|-------|------------|
|       | No. of Runs | No. of Generation |       |         |         |       | No. of Runs | No. of Generation |       |         |         |         |
| 1     | 100        | 0.036756 | 0.035091 | 0.036337 | 0.035091 | 0.036756 | 0.035091 | 0.036337 | 0.035091 | 0.036756 | 0.035091 | 0.036337 |
| 2     | 0.298525 | 0.305882 | 0.308564 | 0.305000 | 0.308564 | 0.298525 | 0.305882 | 0.308564 | 0.305000 | 0.308564 | 0.298525 | 0.305882 |
| 3     | 0.294126 | 0.297531 | 0.294126 | 0.297531 | 0.294126 | 0.297531 | 0.294126 | 0.297531 | 0.294126 | 0.297531 | 0.294126 | 0.297531 |
| 4     | 0.309612 | 0.309612 | 0.309612 | 0.309612 | 0.309612 | 0.309612 | 0.309612 | 0.309612 | 0.309612 | 0.309612 | 0.309612 | 0.309612 |
| 5     | 0.310909 | 0.310909 | 0.310909 | 0.310909 | 0.310909 | 0.310909 | 0.310909 | 0.310909 | 0.310909 | 0.310909 | 0.310909 | 0.310909 |
| 6     | 0.311224 | 0.311224 | 0.311224 | 0.311224 | 0.311224 | 0.311224 | 0.311224 | 0.311224 | 0.311224 | 0.311224 | 0.311224 | 0.311224 |

**TABLE 6.** Comparison of fitness cost for six scenarios based IoT service model.

| ABCO Algorithm | MOPSO Algorithm | WOA Algorithm | Proposed Algorithm |
|----------------|----------------|---------------|--------------------|
| Best Fit | Average Fit | Worst Fit | Best Fit | Average Fit | Worst Fit | Best Fit | Average Fit | Worst Fit | Best Fit | Average Fit | Worst Fit |
| 0.036756 | 0.305091 | 0.36337 | 0.027056 | 0.326804 | 0.387938 | 2.03232 | 0.362462 | 0.317749 | 0.240024 | 0.284064 | 0.337056 |
| 0.298525 | 0.305882 | 0.308564 | 0.255518 | 0.318616 | 0.371257 | 0.250767 | 0.309024 | 0.331074 | 0.234016 | 0.283319 | 0.330224 |
| 0.294126 | 0.305882 | 0.308564 | 0.254922 | 0.305078 | 0.359248 | 0.240104 | 0.284129 | 0.336151 | 0.225984 | 0.271180 | 0.316376 |
| 0.309612 | 0.309612 | 0.309612 | 0.244831 | 0.293172 | 0.340205 | 0.238835 | 0.272718 | 0.321702 | 0.201672 | 0.260174 | 0.301596 |
| 0.310909 | 0.310909 | 0.310909 | 0.243731 | 0.292196 | 0.338232 | 0.238134 | 0.268944 | 0.326373 | 0.203195 | 0.253512 | 0.293504 |
| 0.311224 | 0.311224 | 0.311224 | 0.242634 | 0.290129 | 0.336234 | 0.238378 | 0.258924 | 0.322437 | 0.202823 | 0.247752 | 0.288956 |

**VOLUME 10, 2022**
4) IoE SERVICE COMPARISON OF THE LIFE TIME OF SENSORS

The proposed method is incorporated into the IoE model according to the best, average, and worst fitness functions, which checks the performance of different runs like 1, 5, 10, 15, 20, 25, and 30 on 100 generation from two objective and three objective-based scenarios. Tables 3, 4, 5, and 6 show that the performance of the proposed method is better than that of other standard optimization algorithms in terms (minimization) of energy consumption, delay, and service load. As a result, the proposed algorithm increases the life time of the sensors by two objective and three objective based scenarios in shown figures 10 and 11.

VI. CONCLUSION

In this paper, a novel MODE algorithm based on rapid adaptation-based mutation operators was introduced. The suggested approach tries to increase the MODE algorithms’ rate of convergence while achieving a sufficient level of diversity. The new mutation operator variation described in this research increases the optimal convergence rate for energy consumption, latency, service cost, and fitness cost while providing the DE algorithm with appropriate diversity. By adjusting the tuning settings, the introduced approach is also assessed on six scenarios problems for the increased life of sensors and decreased energy consumption, latency, and fitness cost. From the results on different benchmark functions, it is evident that the performance of the proposed method is significantly improved on maximum variants and performed well compared to the other variants of the MODE Algorithm from the IoE service on bi-objective and tri-objective functions.

REFERENCES

[1] S. Rainer and K. Price, “Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces,” Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep., 1995.
[2] S. Das, S. S. Mullick, and P. N. Suganthan, “Recent advances in differential evolution—An updated survey,” Swarm Evol. Comput., vol. 27, pp. 1–30, Apr. 2016.
[3] B. Chen, Y. Lin, W. Zeng, D. Zhang, and Y.-W. Si, “Modified differential evolution algorithm using a new diversity maintenance strategy for multi-objective optimization problems,” Int. J. Speech Technol., vol. 43, no. 1, pp. 49–73, Jul. 2015.
[4] S.-S. Fang, Z.-Y. Chai, and Y.-L. Li, “Dynamic multi-objective evolutionary algorithm for IoT services,” Int. J. Speech Technol., vol. 51, no. 3, pp. 1177–1200, Mar. 2021.
[5] P. S. Singh and K. Anoj, “Pareto based differential evolution with homeostasis based mutation,” J. Intell. Fuzzy Syst., vol. 32, no. 5, pp. 3245–3257, 2017.
[6] X. Chen, W. Du, and F. Qian, “Multi-objective differential evolution with ranking-based mutation operator and its application in chemical process optimization,” Chemometric Intell. Lab. Syst., vol. 136, pp. 85–96, Aug. 2014.
[7] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, “An efficient approach to nondominated sorting for evolutionary multiobjective optimization, evolutionary computation,” IEEE Trans. Evol. Comput., vol. 19, no. 2, pp. 201–213, Apr. 2015.
[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.
[9] Z.-M. Gu and G.-G. Wang, “Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization,” Future Gener. Comput. Syst., vol. 107, pp. 49–69, Jun. 2020.
[10] A. Got, A. Moussaoui, and D. Zouache, “A guided population archive whale optimization algorithm for solving multiobjective optimization problems,” Exp. Syst. Appl., vol. 141, Mar. 2020, Art. no. 112972.
[11] R. Chaudhry, S. Tapaswi, and N. Kumar, “FZ enabled multi-objective PSO for multicasting in IoT based wireless sensor networks,” Inf. Sci., vol. 498, pp. 1–20, Sep. 2019.
[12] O. Alsayyah, I. Mashal, and T.-Y. Chung, “Bi-objective optimization for energy aware Internet of Things service composition,” IEEE Access, vol. 6, pp. 26809–26819, 2018.
[13] M. E. Khanouche, H. Gadouche, Z. Farah, and A. Tari, “Flexible QoS-aware services composition for service computing environments,” Comput. Netw., vol. 166, Jan. 2020, Art. no. 106982.
[14] A. Chowdhury and S. A. Raut, “A survey study on Internet of Things resource management,” J. Netw. Comput. Appl., vol. 120, pp. 42–60, Oct. 2018.
[15] J. Wan, B. Chen, M. Imran, F. Tao, D. Li, C. Liu, and S. Ahmad, “Toward dynamic resource management for IoT-based manufacturing,” IEEE Commun. Mag., vol. 56, no. 2, pp. 52–59, Feb. 2018.
[16] D. Wu, Z. Zhang, S. Wu, J. Yang, and R. Wang, “Biologically inspired resource allocation for network slices in 5G-enabled Internet of Things,” IEEE Internet Things J., vol. 6, no. 6, pp. 9266–9279, Dec. 2019.
[17] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, “Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things,” IEEE Trans. Ind. Inform., vol. 14, no. 10, pp. 4702–4711, Oct. 2018.
[18] T. Qiu, K. Zheng, M. Han, C. L. P. Chen, and M. Xu, “A data-emergency-aware scheduling scheme for Internet of Things in smart cities,” IEEE Trans. Ind. Inform., vol. 14, no. 5, pp. 2042–2051, May 2018.
[19] J. Eaton, S. Yang, and M. Gongora, “Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 11, pp. 2980–2992, Nov. 2017.
[20] Z. Yang, Y. Jin, and K. Hao, “A bio-inspired self-learning coevolutionary dynamic multiobjective optimization algorithm for Internet of Things services,” IEEE Trans. Evol. Comput., vol. 23, no. 4, pp. 675–688, Aug. 2019.
[21] T. Snyder and G. Byrd, “The internet of everything,” Computer, vol. 50, no. 5, pp. 8–9, Jun. 2017.
[22] D. J. Langley, J. van Doorn, I. C. Ng, S. Stieglitz, A. Lazovic, and A. Boonstra, “The internet of everything: Smart things and their impact on business models,” J. Bus. Res., vol. 122, pp. 853–863, Jan. 2021.

SHAILENDRA PRATAP SINGH received the B.Tech. degree in CSE from the University of U.P.T.U. Lucknow, in 2004, the M.E. degree from the MITS Gwalior, in 2008, and the Ph.D. degree in CSE from MNMIT Allahabad, Prayagraj, Uttar Pradesh, India, in 2017. He is currently working as an Associate Professor with the Department of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh. He is the author of more than 25 articles and more than two inventions. His research interests include optimization, software engineering, machine learning, and the IoT applications.
GAURAV DHIMAN (Senior Member, IEEE) received the master’s degree in computer applications and the Ph.D. degree in computer engineering from the Thapar Institute of Engineering and Technology, Patiala. He is currently working as an Assistant Professor with the Department of Computer Science, Government Bikram College of Commerce, Patiala. He is also associated with Chandigarh University, Graphic Era Deemed to be University, and Lebanese American University. He was selected as an Outstanding Reviewer of Knowledge-Based Systems (Elsevier). He has published more than 200 peer-reviewed research articles (indexed in SCI-SCIE) and ten international books. He is also serving as the lead guest editor of more than 40 special issues in various peer-reviewed journals. His research can also be seen in http://www.dhimangaurav.com.

WATTANA VIRIYASITAVAT (Senior Member, IEEE) received the D.Phil. degree (Ph.D.) in computer science from the University of Oxford, Oxford, U.K., in 2013. He is an Associate Professor of information technology with Chulalongkorn University, Bangkok, Thailand, where he is currently a Full-Time Lecturer and a Researcher with the Business Information Technology Division, Department of Statistics, Faculty of Commerce and Accountancy.

SANDEEP KAUTISH received the bachelor’s, master’s and doctorate degrees in computer science on intelligent systems in social networks and the PG Diploma degree in management. He is working as Professor and the Dean-Academics with LBEF Campus, Kathmandu, Nepal running in academic collaboration with Asia Pacific University of Technology and Innovation, Malaysia. He is an academician by choice and backed with more than 18 years of work experience in academics including over eight years in academic administration in various institutions of India and abroad. He has meritorious academic records throughout his academic career. His research interests include business analytics, machine learning, data mining, and information systems.

* * *

VOLUME 10, 2022

106811