Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8897026, 21 pages
https://doi.org/10.1155/2020/8897026

Research Article

Dimensionality Reduction for Internet of Things Using the Cuckoo Search Algorithm: Reduced Implications of Mesh Sensor Technologies

Azeema Yaseen,[1] **Mohsin Nazir**,[2,3] **Aneeqa Sabah**,[3] **Shahzadi Tayyaba**,[4] **Zuhaib Ashfaq Khan**,[5] **Muhammad Waseem Ashraf**,[6] **and Muhammad Ovais Ahmad**[7]

[1] *Maynooth University, Ireland*
[2] *Asian Institute of Technology, Thailand*
[3] *Lahore College for Women University, Pakistan*
[4] *The University of Lahore, Pakistan*
[5] *Comsats University Islamabad, Attock Campus, Pakistan*
[6] *Government College University Lahore, Pakistan*
[7] *Dept. of Mathematics and Computer Science, Karlstad University, Sweden*

Correspondence should be addressed to Muhammad Ovais Ahmad; ovais.ahmad@kau.se

Received 24 April 2020; Revised 21 September 2020; Accepted 15 October 2020; Published 15 December 2020

Academic Editor: Sungchang Lee

The internet of things is used as a demonstrative keyword for evolution of the internet and physical realms, by means of pervasive distributed commodities with embedded identification, sensing, and actuation abilities. Imminent intellectual technologies are subsidizing internet of things for information transmission within physical and autonomous digital entities to provide amended services, leading towards a new communication era. Substantial amounts of heterogeneous hardware devices, e.g., radio frequency identification (RFID) tags, sensors, and various network protocols are exploited to support object identification and network communication. Data generated by these digital objects is termed as "Big Data" and incorporates high dimensional space with noisy, irrelevant, and redundant features. Direct execution of mining techniques onto such kind of high dimensionality attribute space can increase cost and complexity. Data analytic mechanisms are embedded into internet of things to permit intelligent decision-making capabilities. These notions have raised new challenges regarding internet of things from a data and algorithm perspective. The proposed study identifies the problem in the internet of things network and proposes a novel cuckoo search-based outdoor data management. The technique of the feature extraction is used for the extraction of expedient information from raw and high-dimensional data. After the implementation for the cuckoo search-based feature extraction, few test benchmarks are introduced to evaluate the performance of mutated cuckoo search algorithms. The consequential low-dimensional data optimizes classification accuracy along with reduced complexity and cost.

1. Introduction

The next generation of internet and computers will decline the conventional approach of the internet, reaching to the end-users by promoting the model of interconnecting "smart" objects. It will not replace the internet but will be an addition to (internet) as an infrastructure by integration of physical objects with processing and transmission technologies delivering immense range of services and applications in a more reliable, fast, and accessible way. Such revolution leads towards ubiquitous computing in which every object is embedded with microprocessors and communicate proficiently [1]. It will make physical objects "smart" and let them integrate with worldwide cyber physical frameworks. This trend will pave a way for new openings and innovations in information and communication technologies (ICT), offering new services and applications by connecting physical and virtual objects. It is emerging as a trend in which most of the surrounding objects will be on network in various forms. This is a shift from conventional internet approaches to the internet for connecting physical objects that interact with each other and humans [2].

The scope of this research work is limited to the application of the cuckoo search algorithm for dimensionality reduction issues in data mining for internet of things application. This work is more related to feature extraction as compared to the feature selection. The further scope of this research work is based on transformation of existing features by using cuckoo search algorithms for data mining. Among indoor and outdoor data service, our focus would be on outdoor data services. The outdoor data could be of many types, e.g., text, images, hypertext, audio, and video, but we are only considering textual or numerical regression of our data for the proposed model [3].

The internet of things enables us to connect with anyone, anytime, and at any place. Technological realms are constructing societies, where everything will be connected. Things have capability to be identified uniquely, operating in smart environments with the help of intellectual interfaces to connect and communicate within the physical world. Interconnected smart objects are heterogeneous and equipped with smart devices (sensors and actuators). Data sensed/captured by these objects is huge in amount and can be termed as "big data" [4]. Knowledge discovery and data mining techniques (classification, clustering, and pattern analysis) are proposed for the internet of things (IoT) by researchers to provide a suitable environment and quality services to people. The extensive volume of data (big data) produced by smart commodities with high dimensions, noisy, irrelevant, and redundant attributes generate a huge search space. If mining techniques are applied on such rough and fuzzy data, it can reduce performance, increase cost, and computation of mining algorithms. Therefore, preprocessing techniques are required to map the original datasets onto new reduced attribute subset, which can represent the original space with high accuracy [5].

In the current era, scope of real time networks and applications is not limited to social and enterprise activities. They are emerging as an extensive discipline to provide advanced and competitive environments for diverse activities including health, home, and business processes. To maintain network robustness and accessibility of proficient services, data analytic techniques are crucial. Data purification to reduce computational complexity of preprocessing and mining models is mandatory. Existing techniques are complex, thus involve large computations [6]. These facts introduce research gap, and formulation of the proposed research work is based on the following objectives:

(i) Outlining untaken challenges concerned to deal with big data analysis for the internet of things and with performance analysis and limitations of existing techniques

(ii) Signifying importance of preliminary processing paradigms for self-organized networks to reduce complexity and enhance performance of mining techniques

(iii) Presenting a framework to reduce curse of dimensionality for the internet of things, based on a nonlinear metaheuristic approach, i.e., cuckoo search algorithm

(iv) Investigating a proposed framework performance

The conducted research work identifies challenges for the outdoor data management in the internet of things and the purpose of feature extraction to extract expedient information from raw and high-dimensional data by the technique of the cuckoo search algorithm [7]. The consequential low-dimensional data optimizes classification accuracy along with reduced complexity and cost [8]. The presented work is fundamentally focused to propose a suitable preprocessing framework for internet of things to outdoor data services [9]. Moreover, the proposed technique is only related to single objective optimization, and it can be expanded for multiobjective nonlinear optimization [10]. Implementation within this document only covers the basic concept of the proposed technique and explicitly for the internet of things' outdoor data services [11]. Dataset used in this research is based on the results of virtual hardware devices and may contain some ambitious attributes as well [12].

1.1. Dimensionality Reduction. In an IoT network, data collected by various sensing entities (e.g., sensors and RFIDs) includes redundant, irrelevant, and noisy information about a dataset. The improved performance and effectiveness of IoT services are achieved by applying frequent mining techniques. Data is collected in unwavering speed which increases the complexity of mining algorithm classifiers due to high data dimensions [13]. Data dimension depicts the several aspects or features to describe an input dataset.

$$L = \{f_{i=1}^{n}\}\mathrm{f} \to \left(L^{D} = n\right) \tag{1}$$

Here, D is the dimension, and f_i represents the n features describing the dataset L.

As already mentioned, the accuracy of mining techniques can be affected by high dimensional datasets. So, some preprocessing strategies are necessitated for the transformation of high-dimensional data into lower dimensions (see Figure 1). Data with low dimensionality will optimize classification accuracy along with reduced complexity and cost. Mining the data for IoT is dissimilar to conventional data approaches [14].

In IoT, mining problems occur due to traditional data mining algorithms and hence, they need to be revised for handling scalability and big data issues [15]. Classification problems symbolize many challenges and issues in mining and machine learning research areas [16]. From the IoT point of view, the aim is to classify each object falling into big data according to the characteristics illustrated by its features [17]. Appropriate data representations are needed to describe the physical world data [18], and it is problematic to distinguish the advantageous features. Noisy, redundant, irrelevant, and distorted data can reduce the performance, diminish the classification accuracy, increase cost, and

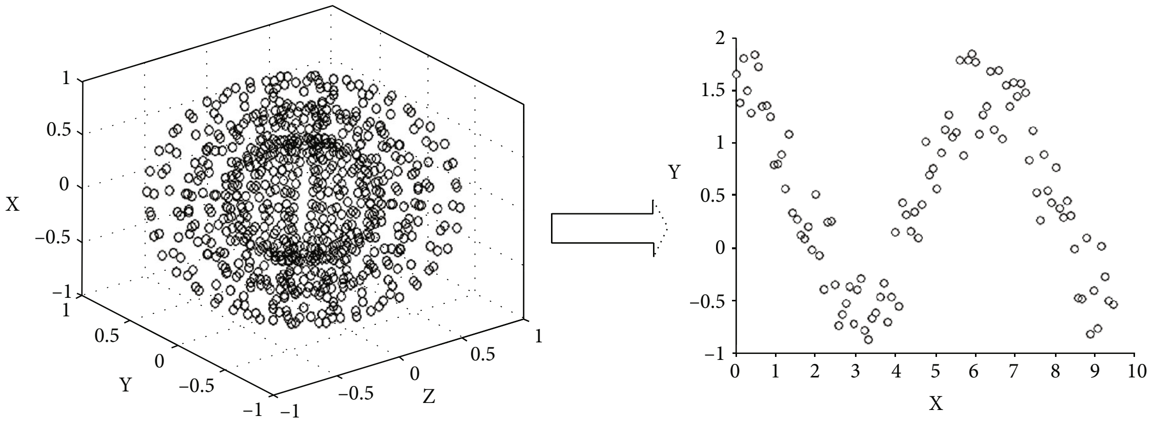

Figure 1: High-dimensional data to low dimensionality showing maximum dimensions of data in (a) 3D (XYZ) plane while graph 2D (XY) on (b) shows data represented by fewer dimensions.

computation of mining algorithms [19]. Eliminating these factors, data mining techniques can benefit and work more efficiently [20]. According to the work proposed in [21], reducing data dimensions (dimensionality reduction) can handle this problem by the selection of relevant and meaningful features only.

If L is a dataset with dimension D, such that $L = \{f^n_{i=1}\} \in \mathbb{R}^D$, then dimensionality reduction is implied, such that there will be a dataset M having fewer dimensions d, D, and $M = \{f^n_{i=1}\} \in \mathbb{R}^d$, where $d \ll D$ in such a way that M represents the dataset L with fewer feature subset or dimensions d.

The reduced set of data can expand the performance and speed of mining algorithms, which leads towards optimal classification results and better network performance. Dimensionality reduction is a complicated challenge due to a large search space as the size of data increases exponentially based on its attributes. An attribute may become relevant or redundant in various scenarios according to properties specified by its dimensions. So, optimal search techniques are indispensable for exhaustive search, which is impossible when a search space is indefinite. An inclusive range of searching techniques has been proposed (e.g., sequential forward/backward selection) to select profound attributes to reduce data dimensions which can represent the data in the most appropriate form for classification and other mining strategies. Despite these approaches, attribute reduction practices are undermined from data and algorithm perspectives and going through challenges for dynamic local optimum, exceeding cost and computational complexity [22].

1.2. Techniques for Dimensionality Reduction. Reducing the number of dimensions from the extensive search space is a challenging and demanding call for current networks and computing technologies. Figure 2 describes an abstract process for knowledge discovery and dimensionality reduction. Feature selection (FS) and feature extraction (FE) are two approaches for dimensionality reduction and will be discussed in following sections [23].

Dimensionality reduction (branch of statistics and machine learning) is emerging as a new realm in the solution domain for big data issues to map an original feature space onto a new space This process minimizes the number of random variables (attributes) under consideration and transforms data from high dimensions to low dimensions. Two approaches for mapping can either be choosing a subset of the original feature space (feature selection) or by forming a new space using a transformation function (feature extraction). Feature selection (or feature subset selection) from available datasets is considered more efficient to represent original data comparatively [24].

1.2.1. Feature Selection. Preprocessing techniques are indispensable for data mining to reduce complexity, processing, storage, and cost of classifiers. In IoT, input datasets on data collection layers are represented by high dimension variables and raise the processing complexity of mining algorithms. Feature selection outlines the problem of selecting a feature subset from available candidate features representing originally measured datasets [25]. Feature selection and extraction are also used extensively in image processing and computer vision field of research [26].

A more technical work on the local manifold representation with usage of affinity matrix in the field of dimensionality reduction in hyperspectral imagery [27]. In the field of hyperspectral dimensionality reduction for remote senting data, machine learning models are also extensively used for labeling the graphs along with learning features [28].

Let input to a mining or training classifier is a set of n datasets. Each dataset L can be a set of N features describing the original set of features. The instance L is a tuple including D dimensions as given below.

$$\begin{aligned} &L = \{F_1(L), F_2(L), F_3(L), \cdots\cdots, F_i(L)\}, L \in \mathscr{R}^D \\ &|L| = N, \rightarrow (N = D) \end{aligned} \tag{2}$$

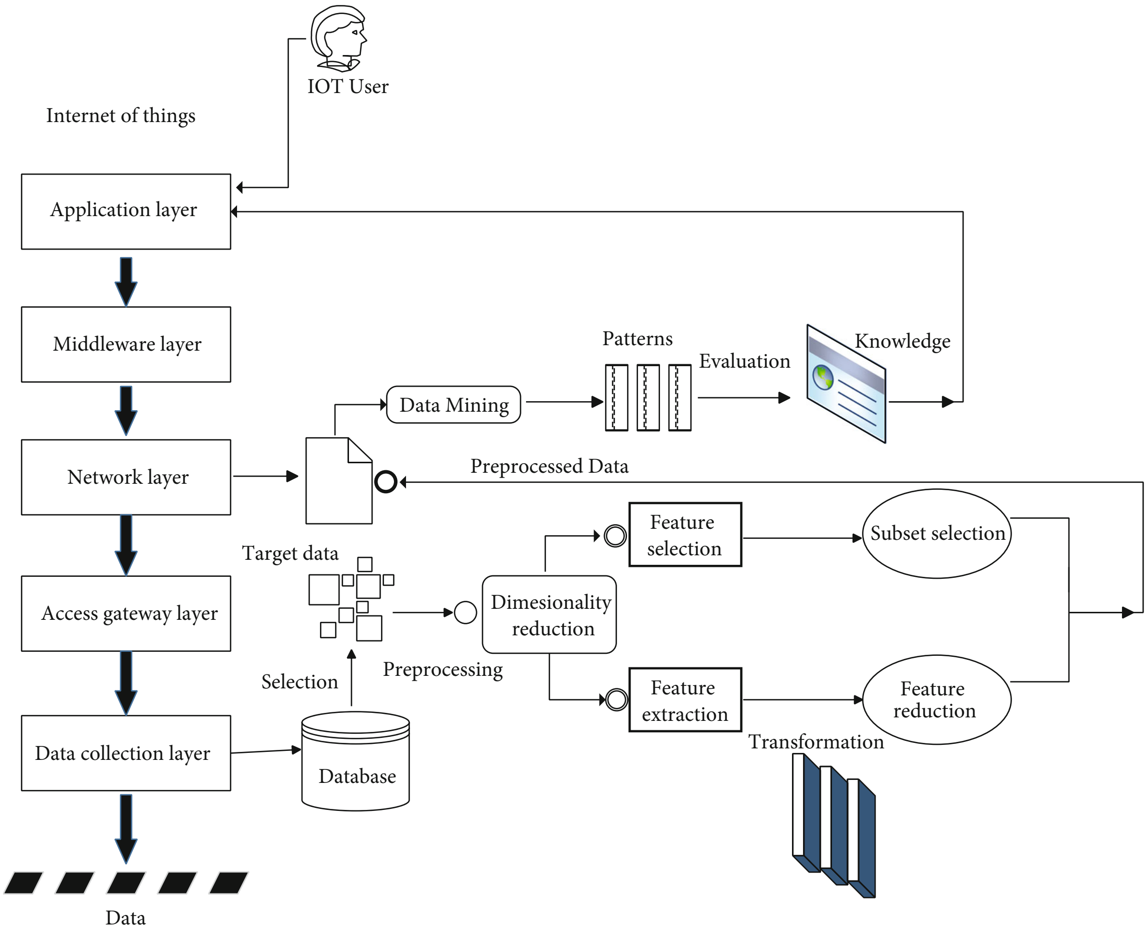

Figure 2: Knowledge discovery and dimensionality reduction for internet of things representing techniques of feature selection or feature extraction to reduce the number of original dimensions for outdoor data in internet of things introduced on the network layer.

where F_i is the domain of the i^{th} feature, and cardinality of L is N, having dimensions D. Let n represents the selected dimensions d in $M, M \subseteq L$.

$$M = \{f_i(M), f_{i+1}(M)\} \to M \subseteq L \subseteq \mathscr{R}^D, \tag{3}$$

$$|M| = n \to (n = d). \tag{4}$$

Equations (3) and (4) define a new subset of L which is smaller than the original set and belong to the same search space that is $M \subseteq L \subseteq \mathscr{R}^D$, where number of features d in M is equal to n. The process of feature selection is shown in Figure 3 which is based on two phases. The first part of the process selects subset from original spaces, and the second section evaluates the newly generated subset. Let L be the representation vector and $P(M)$ be the selection criteria for optimal subset M. Formally, feature selection maps the high-dimensional space to low dimension by finding a subset $M \subseteq L$ where $|\mathrm{M}| = d$.

$$\mathrm{P(M)} = \max_{K \subseteq L, d \ll D} P(K). \tag{5}$$

According to equation (5), the higher value of $P(M)$ signifies an improved feature subset. The selected subset performs as a best input to the classifiers and expands the accuracy rate. It is given by the feature selection criterion that only a subset is selected that forms a large set of variables to represent datasets and does not incorporate any transformation and mapping to extract new information from existing high-dimensional datasets.

1.2.2. Feature Extraction. Feature extraction, an imperative data preprocessing technique, adds value to the mining techniques as a performance enhancer for IoT networks to transform existing high-dimensional datasets, i.e., uploading high accuracy data representation model for the original feature space [29].

$$\begin{gathered} X = \{x_1, x_2, x_3, x_4, \cdots \cdots \cdots, x_n\}, \\ X_i \in \mathscr{R}^d \Rightarrow Y_i \in \mathscr{R}^p | (p \ll d). \end{gathered} \tag{6}$$

The existence of inappropriate and redundant facts in original datasets demand a prerequisite process (feature extraction). The feature extraction process undermines two research issues: search technique and evaluation measures. The search space includes complete and feature subset,

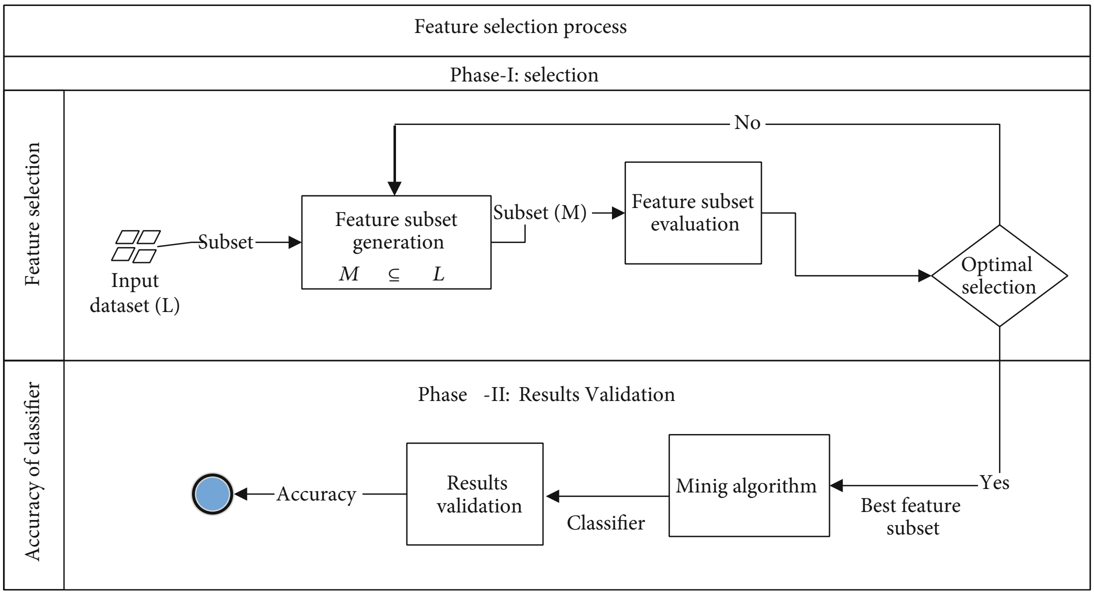

Figure 3: Feature selection-based dimensionality reduction: selection is made in phase I while phase II ensures the accuracy for chosen classifier after the feature subset selection.

and feature extraction transforms existing features to find optimal solution set [30].

$$\sum_{f}^{D} = \mathrm{O}\binom{D}{f} = (1+1)^{D} = 2^{D}. \tag{7}$$

In equation (7), D represents dimension, and f denotes the recent feature subset size. Various evaluation techniques have emerged for the optimal subset selection. Searching for an optimal feature subset is termed as a nondeterministic polynomial- (NP-) hard problem. Traditional searching algorithms are not efficient to handle the high-dimensional search space. Evolutionary computing (EC) algorithms are prominent for their optimal global search competency [31]. Dimensionality reduction is a fact-based problem that determines two basic reasons including reduction of the feature space and to enlarge the accuracy of mining strategies which are demonstrated in Figure 4.

1.3. Heuristic and Metaheuristic Search Methods. Internet of things requires abstract data representation with relatively less number of features, which is a fundamental to data analysis and decision-making tasks. Optimization tools are necessarily a way to find an optimal solution driven by dynamic optimal parameters [32] in a live network like IoT. The dynamically attuned algorithms are applicable when optimization is multiobjective, e.g., maximizing a search optima parallel to another network optimization goal. In such a dynamic network environment, where the data streams are fluent, repititive, and continuous, optimization becomes a dynamic functional requirement and hence, the passive- and problem-specific algorithms do not offer sustainable search optimization for uninterrupted service delivery. The heuristic techniques are problem-dependent and implicitly deliver an approximate solution for a particular situation without an exact accuracy level. The combinatory and low-rank representations based on the heuristic search algorithm in [33] provide an evidence in favour of heuristic techniques but this is only applicable when the objective function is not dependent or followed by another global minima or maxima function.

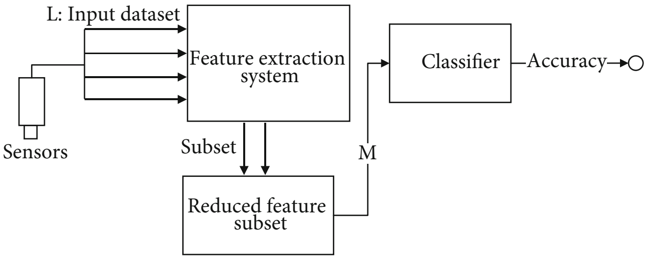

Figure 4: Feature extraction-based dimensionality reduction: data collected by sensors and input L to the feature extraction system for reducing dimensions to get a reduced feature subset M which is sent to a classifier for improved accuracy.

The studies presented in [34, 35] propose optimization algorithms to get an optimal solution which also ensures the quality and efficiency of the solution with some proving statements.

Metaheuristics are problem-independent techniques that can be applied to a broad range of problems. A heuristic is, for example, choosing a random element for pivoting in Quicksort. A metaheuristic knows nothing about the problem it will be applied, but it can treat functions as black boxes. As a general distribution, the algorithms are defined by two representative categories: (i) deterministic and (ii) stochastic algorithms. In contrast to stochastic algorithms, deterministic techniques are linear, where initial variables control and determine the output with no random variables, hence, does not need to be adapted for random optimization problems. In stochastic algorithms, random output could be the end outcome of same or random input parameters depending on the triggered operations. The Stochastic algorithm can further be classified by two types of algorithms: evolutionary and metaheuristic algorithms. As given in [36], metaheuristics are naturally and biologically inspired algorithms, offering their applications in various global optimization and real-time problems. Some of these are ant colony optimization [37], particle swarm optimization (PSO), and state transition algorithms (STA) [38, 39]. In spite of the extensive optimum nature, these algorithms have some degree of randomness; it means they reduce the global search ability and easily fall into local optima. To address this, a mutated cuckoo search algorithm is proposed which establishes a solution space as a global function.

1.4. Cuckoo Search Optimization. Cuckoo search (CS) is a metaheuristic optimization technique, proposed in 2009 [40], inspired by some successful characteristics (e.g., breeding) of cuckoo's biological behavior. The growth of this algorithm depends on two terminologies: randomization (random walk) and stochastic search. The algorithm begins to explore the local search space (R_n) for local optima. Moreover, the algorithm is not bounded by local optima; instead, it expands as the problem becomes global and offers a global optimal solution [41]. Though the originally proposed algorithm is tuned to have relatively less number of parameters and dedicatedly targets local optimization, but according to the revised and enhanced CS algorithms in [32], it can be tailored as a dynamic and global optimization algorithm to amend its performance boundary by adjusting the step size and parametric values.

1.4.1. Cuckoo's Living Behavior. Cuckoo is an obligate and brood parasitic organism which depends on other host birds for their reproduction and to grow its offspring [42]. The cuckoo search (CS) is instigated by the influence of cuckoo's genetic activities, e.g., foraging (search for food) [43]. Cuckoo lays its egg in the nest of host birds where eggs hatch and offspring seeks for host attention to get food [44]. Moreover, it also imitates some exterior attributes of host eggs. It is based on two approaches: exploration and exploitation. CS makes use of levy flights to generate a new solution. Cuckoo may throw the eggs of a host bird to raise the hatching probability of its own egg [45].

1.4.2. Algorithm Constraints

(a) Each cuckoo lays one egg at a time and pitches it in a randomly chosen nest

(b) The nest with the best eggs will grow as a next generation

(c) The number of nests is fixed. There is a probability $P_\alpha(0, 1)$ that alien eggs can be identified by the host bird. If it happens, then the host bird either discards eggs or leaves the nest and builds a new one

1.4.3. Algorithm Formulation. Cuckoo selects a nest and dumps its egg into it that is owned by some host bird. The selection of nests depends on the random walk. The randomization is depicted by the foraging and flight behavior of a cuckoo. Each egg laid by a cuckoo represents a new solution V^{t+1}.

$$v^{t+1} = A_{\text{CS}}\left(v^t, P(t)\right), \tag{8}$$

where A_{CS} is a nonlinear cuckoo search algorithm that maps existing IoT d-dimensional vector v^t with parameters $P(t)$ to a relatively new vector v^{t+1}. According to several observations, it is deduced that the natural flying pattern of a cuckoo and the characteristics of a levy distribution are quite similar. These random walks are not isotropic, i.e., vary in directions and magnitude.

1.4.4. Random walk. The formulation of the CS algorithm is an equilibrium consolidation of local and global random walk. Hence, it does not only optimize outdoor IoT data but can also converge to local optima when required. A random walk is a sequence of successive random or stochastic processes.

$$X_i\,(\text{t} + 1) = X_i\,(\text{t}) + \text{step size} \oplus \mathscr{L}(\beta), \tag{9}$$

where $\mathscr{L}$ indicates levy flights and $\oplus$ denotes sequenced multiplication for each new step, which is then added to the previous candidate solutions. In each new iteration, a solution is generated through levy flight, and steps for search are taken from the levy distribution bound to heavy-tailed distribution. As compared to the normal distribution, the heavy-tailed distribution is not exponentially bound, and most of the values during generation meet the criteria of the fitness value (objective function) [46]. A random walk is shown in Error! Reference source not found. For 10,000 steps taken at a time t to choose a better position than the previous one, levy flight is preferable when the search space is exponentially unbounded and continuously expands in any dimension and size. Cuckoo search is highly recommendable because of its levy flights to handle network data for global optimization [47].

In information and communication technology, requirements for fast and self-organizing algorithms are indispensable when data is huge in amount and various data analytic activities that are initialized to improve network performance. Metaheuristic algorithms are one of the global optimization techniques that are designed to sort out current global optimization problems. Among several metaheuristics algorithms such as harmony search and bat algorithm, cuckoo search is a newly developed algorithm that can best fit the future smart IoT networks and their continuous outdoor data to provide valuable services with improved machine learning techniques [48]. Figure 5 shows a random walk graph with 10,000 steps starting from 0, where the x -axis represents the time lapse t, while the y-axis shows the position of movement.

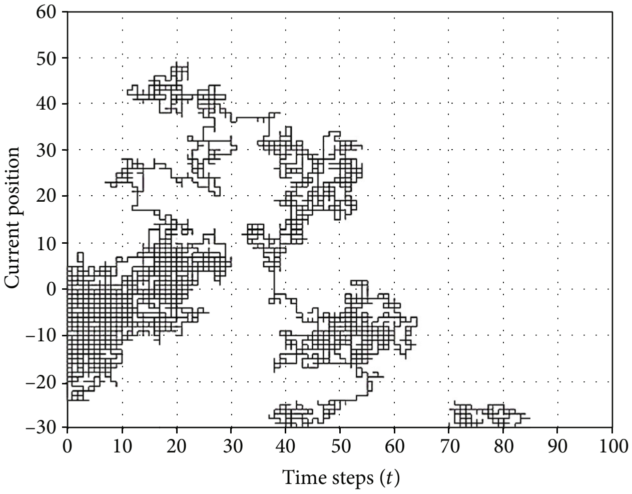

Figure 5: Random walk: a graph showing a random walk of 10,000 steps starting from 0, where y-axis represents the current position and x-axis represents time t.

1.4.5. Parameter Tuning. The Cuckoo Search Strategy (CSS), in its originated form, is optimal for local search optimization but its algorithmic constraint can be tuned to broaden its applications for global search optimization problems. In order to achieve the best performance, a number of parameters in the cuckoo search algorithm need to be tuned, namely, the nest size, the elitism probability (probabilistic selection of fittest candidates), and the repetition [49]. Similar to many nature-inspired algorithms, the algorithm starts with random parameters. On each step of the iteration, the parameters are tuned with varying step size. The selection of the step size is important to convergence or divergence of the algorithm. Based on different applications, the step size can be increased or decreased for speedy convergence or performance requirements.

1.4.6. Efficiency. Cuckoo search is a metaheuristic algorithm which is nature-inspired and is now among the most widely used algorithms for optimization. It has many advantages over conventional algorithms due to the inherent randomness in its approach. Metaheuristic algorithms are very diverse, including genetic algorithms, simulated annealing, differential evolution, ant and bee algorithms, bat algorithm, particle swarm optimization, harmony search, firefly algorithm, and cuckoo search [50]. These algorithms are nature-inspired and work without any central computing paradigm. Most of the parameters are tuned with neighbouring nodes interacting with each other. The interconnection among peers makes them as nonexponentially complex.

1.4.7. Limitations. The performance of the cuckoo search algorithm is compromised if the problem is discrete and multiobjective though it performs well for continuous optimization problems. Therefore, the algorithm has limited scope when processing some real-time problems; it demands further study to overcome its limitations. Other than the continuous problems, there has been much development in terms of the step size, parameter adjustment, intercoupling with other algorithms, and other factors used to improve the performance-related markers. Meanwhile, this algorithm also has problems in adaptability and getting the best possible search results, and its algorithmic ability to solve complex problems is inadequate for real-world applications. Future research should be focused towards studying and exploration of new methods and strategies to improve high coupling functions between variables [51].

1.5. Motivation. In wireless sensor networks and IoT-based system, data is generated in enormous volume. This enormous volume is cumbersome to analyze for any fruitful analysis of data. We have to clean this data in any of the phases before the analysis of results is generated from data. Sometimes, this huge volume of data is cleaned during the nonoperational time of data processing, and existing data is updated with removed redundant data. This is also possible that data is cleaned before any analytical processing during runtime [52]. Both approaches have its own pros and cons. With reduced dimensions on scientific and mathematical basis, the data is safe with less volume. The reduced amount of data based on less dimension is always a hot topic of research with diverse implementation of IoT-based systems [53].

The tricky difference between selection and reduction is compromises on selecting the features required during run time execution in feature selection and dropping the undesired and unimportant features during the data cleansing phase. In many real-time applications, searching the huge amount of data with the high-dimensional search space is not practically feasible. The existence of unimportant features causes interferences due to redundancy, irrelevancy, and triviality of the data search space. Many of the evolutionary algorithms lack this inherent attribute for accurate selection of features for deletion and reduction of the attributes. The results in the paper show that cuckoo search outperforms many of the existing and applied algorithms for dimensionality reduction phenomenon [54].

2. Proposed Technique and Implementation

In any IoT network, data from the physical world is highly nonlinear as its environment changes dynamically depending on local or global activities. So, to manage data, preprocessing to mining and decision-making purification is needed for accurate classification and reduced cost. Objects can leave or join the network from time to time, which will need to restart the mining algorithms to deal with immediate and abrupt changes. To reduce the huge search space, selection of reduced attribute subset, and minimizing the cost of mining algorithms, feature extraction (FE) can be an essential strategy to enhance the performance of classifiers and other mining techniques. Moreover, novel algorithms are needed to overcome shortcomings of traditional approaches [55].

2.1. IoT Vectors and Dimensions. The exhaustive search space $\mathbb{R}^d$ consisting of enormous dimensions D is reduced through feature extraction before any further tasks relevant to data analytics involving data mining techniques. The resultant moderated vectors will increase prediction accuracy with least complexity and cost.

2.1.1. Problem Description. Data gathered by sensors is stored in a database as a combination of rows and columns. Each row represents a distinct vector v_i or dataset and each column $\mathscr{C}$ with its corresponding dimensions d_j. A schema for IoT databases including dimensions and vectors is given in Table 1.

Suppose all distinctive objects in IoT are represented in vector set V, where each vector v_i stores data of a single object with several dimensions (attributes) $d^j \in D$, then the number of vectors in V is equal to the number of tuples in the database, so that $1 \leq V \leq n$. A set of n vectors can be demonstrated as a column matrix given below:

$$V = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix}, \forall^{v_1} V \in \mathbb{R}^n, \tag{10}$$

where each vector v_i can be represented along with its dimensions as a single row matrix.

$$v_i = [d_1\ d_2\ d_3 \cdots d_{m-1}\ d_m], \forall \in \mathbb{R}^D. \tag{11}$$

If we replace each vector with its dimensions, then we get a $m \times n$ matrix.

$$V_{m\times n} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1m} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2m} \\ d31 & d32 & d33 & \cdots & \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{m1} & d_{m2} & d_{m3} & \cdots & d_{mn} \end{bmatrix}, \forall V_{m\times n} \in \mathbb{R}^{m\times n}. \tag{12}$$

With respect to vector v_i and its dimensions, $[d_1\ d_2\ d_3 \cdots d_{m-1}\ d_m]$ transforms the existing features that belong to the original dimension space $\mathbb{R}^D$ into reduced and transformed vector v_i' and space $\mathbb{R}^d$ such that

$$\begin{aligned} &\text{If } v_i = [d_1\ d_2\ d_3 \cdots d_{m-1}\ d_m], \forall^{v_1} \in \mathbb{R}^D. \\ &\text{Then } T(v_i) = T[d_1\ d_2\ d_3 \cdots d_{m-1}\ d_m], \\ &\quad = > T(v_i) = v_i' = \left[d_1', d_2', d_3' \cdots \cdots \cdots, d_m'\right]. \end{aligned} \tag{13}$$

Before we formulate the algorithm for feature extraction, a brief description for used terms is given in Table 2.

Table 1: Two dimensional data for internet of things: representation for datasets $v_i \in V$ and their corresponding dimensions $d^j \in D$.

$V = v_i$			$D = d^j$		
	$d^{j=1}$	$d^{j=2}$	$d^{j=3}$	$\cdots$	$d^{j=m}$
$v_{i=1}$	d_{11}	d_{12}	d_{13}	$\cdots$	d_{1m}
$v_{i=2}$	$\ddots$	d_{22}	d_{23}	$\cdots$	d_{2m}
$v_{i=3}$	d_{31}	$\ddots$	d_{33}	$\cdots$	d_{3m}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$v_{i=n}$	d_{n1}	d_{n2}	d_{n3}	$\cdots$	d_{nm}

2.2. Feature Extraction. The component V is an original vectors' set of dimension n, and $\boldsymbol{v_i}$ is a distinct untransformed vector of dimension m. We can illustrate v_i' as a transformed vector of dimension m' such that $v_i' \in \mathbb{R}^d \ll \mathbb{R}^D$. The task of feature extraction is divided into the following steps:

(i) Feature construction (FC) or feature transformation (FT)

(ii) Features' subset selection (searching technique)

(iii) Result efficiency

Depending on these tasks the algorithm for feature extraction can be formulated to maintain a general procedure for meaningful and efficient feature extraction. An algorithm for extraction of high-quality features from the original space is given below.

2.3. Feature Construction. The feature extraction is a technique for transformation of a vast range of features or dimensions into a reduced set of features for various data analytics tasks. Transformation of original features needs some parameters and technique to construct new features from the existing one. These techniques can vary depending on the type of attributes to be constructed. In the following sections, algorithms are presented for nominal and numeric attributes. It will reduce dimensionality of data to enhance search optimization for any machine learning task, e.g., classification and data mining to target the suitable data for further processing in IoT networks. A vector with comprehensive and well-constructed attributes can benefit to achieve high prediction accuracy. Construction of nominal and numeric attributes required different operators and operations to identify hidden information which can be beneficial to data analytics for decision-making [56].

2.3.1. Algorithm for Numeric Attributes. The IoT database includes various dimensions, and each dimension can be represented as a numeric value or combination of characters and strings. In both cases, a set of operators are changed to construct features accordingly. Algorithm given below specifies the steps to construct new features for numeric

Table 2: List of variables and their representation for Internet of Things.

Term	Representation
Dataset/vector in IoT	$V = \{v_i\}$
Number of vectors	$n\{v_i, i = 1, 2, 3, \cdots., n\}$
Features/dimensions of vector	$D = \{d_j\}$
Number of dimensions	$m\{d_j, j = 1, 2, 3, \cdots., m\}$
Feature space	$\mathbb{R}^D$
Vector space	$\mathbb{R}^n$
Reduced feature space	$\mathbb{R}^d$

attributes based on arithmetic operators (+,/, -). Figure 6 gives the tree representation for feature construction with various mathematical operations performed for dimensions d^i. Here, the selection of the operators depends on the problem and desired outcome [57].

Algorithm starts with initial input vectors $v_i^{\sim}$, where each distinctive vector is a collection of characters or strings. First loop selects dimensions of an untransformed vector $\boldsymbol{v}_i^{\sim}$and copies each feature d^j to a new vectorv_i. After construction of selected attributes, original attributes are discarded to avoid duplication of similar attributes. Internal two loops choose attributes from a new vector v_i and select operators from a list of arithmetic operators. Last loop selects each vector one by one and adds all constructed features from $j = 1, 2, 3, \cdots, m$ for each vector $v_i, i = 1, 2, 3, \cdots., n$.

2.3.2. Algorithm for Nominal Attributes. Other than arithmetic operations, concatenation of strings or characters is used to generate new features, if the type of attributes is not numeric. Algorithm for construction of nominal attributes is described below.

Nominal attributes are concatenated to construct new features by making various pairs among all fields in a dataset $v_i^{\sim}$. First loop copies values from the original vector to a new vector v_i.

Last two loops select each dimension from a new array $\boldsymbol{v}_i$ and select each vector sequentially to construct features for all vectors $v_i, i = 1, 2, 3, \cdots., n$. Figure 7 gives the Heaviside function $\theta(p_a - \sigma)$ a representing dimension scaling factor at time t.

The local random walk for local optimum solution is isotropic and can be represented as follows:

$$v_i^{t+1} = v_i^t + \psi s \otimes \theta(p_a - \sigma) \otimes \bigg(v_i^j(t) - v_k^j(t), \tag{14}$$

where $v_i^j(t)$ and $v_k^j(t)$ represent two distinctive vectors $(i \wedge k)$ with the j^{th} d dimension at time (t). ψ is a scaling factor for size transformation to control the search space in IoT.

$$\begin{gathered}\theta(p_a - \sigma) = \begin{cases} 0, & p_{\mathrm{a}} > \sigma \\ 0.5, & p_a - \sigma, \\ 1, & p_a < \sigma \end{cases} \\ v_i^j(t+1) = v_i^j(t) + \psi \otimes \mathscr{L}(s, \lambda), \\ \mathscr{L} \sim \mathcal{u} = s^{-\lambda}(1 < \mathscr{L} \leq 3), \mathcal{u} \sim (0, 1), \\ \psi \otimes \mathscr{L} = \{\psi \cap \mathscr{L} : \forall \psi > 0, \forall 1 < \mathscr{L} \leq 3, \psi \cap \mathscr{L} \neq 0\}. \end{gathered} \tag{15}$$

Here, λ is taken from a uniform distribution, and each step in levy flight is taken from a heavy-tailed distribution. The levy distribution based on the heavy-tailed distribution increases the probability for selection of each dimension d^j in vector v_i.

2.4. Mutated Cuckoo Search=Based Feature Extraction (CSFE) Algorithm. The improved version of CS is used to extract old and new dimensions along with reduction of overall existing dimensions. Algorithm starts with the step of constructing new features based on original input vectors and selects enhanced attributes for each vector that is given in Table 3 [58].

The maximum optimization is achieved when in each iteration dimensions d^j at extensive distance are chosen to identify how they are compatible to each other in one vector v_i. This task is based on the dimension section (DS) performed according to the rule inspired by cuckoos' strategy of laying eggs in habitat. The DS for each new transformed vector can be calculated as

DS= ψ Number of current dimensions in v_i Total
Number of dimensions × $(d^m - d^i)$.

Algorithm input parameters are as follows:
Dataset environment: $V = \{v_i, i = 1, 2, 3, \cdots, n\}$
Number of datasets in V: n
Discarding probability: p
Scaling factor: ψ
Number of dimensions: $D = \{d^j, j = 1, 2, 3, \cdots, m\}$
Number of iterations: T
Output: globally optimized V'

Auxiliary parameters are as follows: fitness vector v_i with dimensions m', global fitness $\mathscr{G}$, and local fitness $\mathfrak{L}$.

The algorithm for the cuckoo search-based feature extraction is provided in the following section where the objective function is chosen according to the cited problem of outdoor IoT data. The algorithm will generate a feature subset in each iteration and continue this procedure until an optimized cost or performance is achieved.

2.4.1. End Procedure_CSFE. The mutated cuckoo search-based feature extraction (CSFE) algorithm includes three procedures demonstrated as procedure_1 for construction of numeric attributes, procedure_2 for nominal attributes construction, and last is procedure_3 for apply global selection strategy onto constructed features to find more appropriate features which describe each vector $\boldsymbol{v}_i$ to improve prediction exactness. Procedure_3 starts with two fitness

```
Input initial vectors set $V\sim$ with original features $D\sim$
**For** $(\forall v_i \in V\sim(t-1), i = 1, 2, 3, \cdots., n)$ **do**
        Construct new features $F_n$ for each $v_i$ **(transformation)**
        Update each vector $v_i$ and add it to new vectors set $V_c$
                $V_c(t) = C(V(t-1)) = F_c - v_i$
Select subset of constructed features from $V$ through **Cuckoo Search**
    $F(V_c(t)) = (V(t)) \rightarrow {}^{CS}V'(t)$
Generate optimized vectors set $V'$ with reduced dimensions $D'$
**End Procedure_1**
```

ALGORITHM 1: Initiate procedure_1.

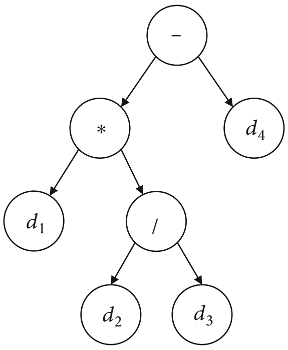

FIGURE 6: Representation for feature construction: various mathematical operations are performed for dimensions d^i.

```
**IF** (attributes ! = numeric attributes) **Then**
        Go to procedure_2 for Nominal Attributes
**Else** Start Procedure _1 for Numeric Attributes
Input vector $\mathbf{v}_{\mathbf{i}}^{\sim}$, a set of nominal attributes
$d^j \in v_i^{\sim}$, single feature in each vector
$V\sim = \{t_1, t_2, t_3, \cdots., t_n\}$, sequence of n tuples
$V_c(t) = \varnothing$, set of newly constructed features
$A$ : set of arithmetic operators
**For** each $(d^j \in v_i^{\sim})$**do**
        $v_i = v_i^{\sim} - d^j$ (Prevent duplication of a feature)
        **For**$(\forall d^j \in v_i)$
                **For**$(\forall a \in A)$
                        **For**$(\forall v_i \in V)$
                                $F_c = \sum_{i=1}^{m}(d^j, d^{j+1})$
                                        $V_c(t) = F_c \cup V_c(t)$
**End Procedure_2**
```

ALGORITHM 2: Initiate procedure_2.

functions $\mathcal{F}_V$ for global optimization and $\mathfrak{L}_{v_i}$ to evaluate each vector for local optimization, initially sets the existing n vectors $(v_i, i = 1, 2, 3, \cdots, n)$ as an input to procedure_3. First loop calculates fitness $\mathfrak{L}(v_i)$ for original n vectors sequentially before selection through CS. The next loops select each dimension from (v_i) at time T until it is not equal to the length of tuples in original vectors space and select dimensions randomly through levy flight and generate a new vector v_i' including optimal dimensions m'. In the next step, fitness for new vector v_i' is calculated. If the fitness of the reduced vector is maximum than the existing one, then the replacement is conducted and abandons the worst vector, assembles all best fitted vectors v_i' to V', and finds fitness to compare with the existing one and replace if necessary. Last, **IF** statement finds efficiency for newly built vector space to decide whether it should be discarded or placed for further processing.

Input to the algorithm is the original feature set, and construction of features is performed according to the identified type of input features (numeric or nominal). After the construction, updated space is relocated to cuckoo search for selection of appropriate features through random walk and levy flight. The output is a new and enhanced feature subset for each vector $\boldsymbol{v}$.

2.5. Dataset Generation. The internet of things has evolved as a preeminent and exquisite source to provide valuable services to consumers, business analysts, and industries in their daily professional and personal lives where things can connect themselves to the internet and serve without any delay. Despite this rapid evolution, familiarity and adeptness to the IoT network are quite gradual. Only few highly recognizable industries are providing valued services to their consumers. In literature, most of the work regarding IoT only demonstrates the fundamental concepts and architectural aspects. In the current era when technological advancement is more beyond than the internet, the real-time networks are facing challenges to accommodate continuous and abrupt amount of collected data for various mining and machine learning tasks to accomplish the goal of smart and intelligent networks with self-continuation ability without human intervention.

The internet of things maintains data collected by the entities that are part of it. These entities are computing devices (scanner, thermostat) that can communicate over the internet to share their information and services. These smart objects are distinct, and the EPC of each object can be stored as a primary key in the database to maintain its record. The organizations with advancement of IoT are not willing to share their private data publicly for security and confidentiality. That is why in literature and over the internet dataset relevant to IoT networks are quite unavailable. To apply proposed solutions (CSFE) for dimensionality

```
IF (attributes ! = Nominal attributes) Then
    Go to procedure_1 for Nominal Attributes
Else Start Procedure_2 for Numeric Attributes
Input vector $v_i^{\sim}$, a set of nominal attributes
$d^j \in v_i^{\sim}$, single feature in each vector
$V \sim = \{t_1, t_2, t_3, \cdots, t_n\}$, sequence of n tuples
$V_c(t) = \varnothing$, set of newly constructed features
$\mathcal{con}$ (): concatenate features in $v_i^{\sim}$
For each $(d^j \in v_i^{\sim})$do
    $v_i = v_i^{\sim} - d^j$ (Prevent duplication of a feature)
    For$(\forall d^j \in v_i)$
        For$(\forall v_i \in V)$
            $F_c = \mathcal{con}(d^j, d^{j+1})$
                $V_c(t) = F_c \cup V_c(t)$
End Procedure_3
```

Algorithm 3: Start procedure_3.

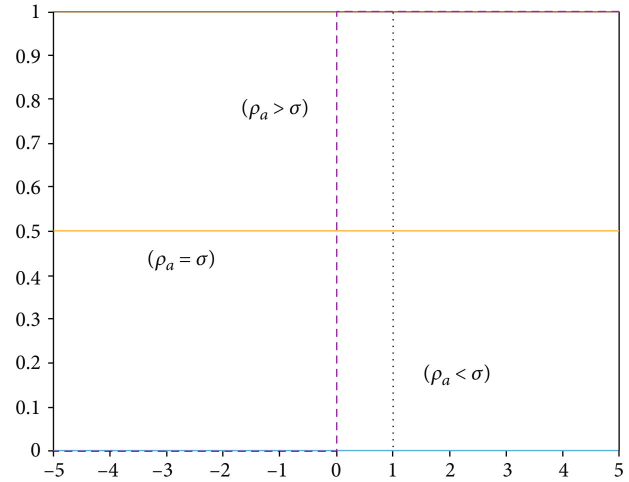

Figure 7: Heaviside function: $\theta\,(p_a - \sigma)$, representing dimension scaling factor at time t.

reduction onto outdoor IoT data, an appropriate and relevant dataset is required. It will facilitate to produce appropriate results through implementation of suggested technique in MATLAB for reduction of the overall extensive search space $\mathbb{R}^n$ and feature space $\mathbb{R}^D$.

IoTify is a web-based platform for simulation to develop IoT applications by using virtual hardware devices, e.g., sensors. It facilitates a virtual lab and enables the creation and building of virtual IoT devices in JavaScript. The IoTify database is generated using JavaScript object naming (JSON) with extension JSON. Table 4 demonstrates the dataset that is used to accomplish results and for analysis of proposed algorithms (CSFE). IoT-based devices can extract specific and required facts from a patient's blood and will share the generated report to the doctor when an alarming situation arises. Figure 8 outlies a flowchart for the mutated cuckoo search-based feature extraction including procedures for nominal and numerical attributes with generation of final reduced feature subset.

Table 3: List of parameters and their symbolic representation in the cuckoo search-based feature extraction algorithm for dimensionality reduction.

Parameters	Algorithmic representation
Dataset environment	V
Fitness function	$\mathcal{F}(x)$
Number of datasets	N
Feature/dimension in i^{th} vector	j
Vector in dataset environment	i
j^{th} dimension in i^{th} vector	v_i^j
Probability of discarding a vector	p_i
Levy flight	$\mathcal{L}\,(\lambda)$
Number of iterations	T
Time instance	T
Step size	S
Step size scaling factor	$\psi > 0$
Normal distribution	u (0,1)

Reduction of original dimensions for outdoor IoT data is performed through the task of feature extraction. Here, the task of feature extraction as subtasks of feature construction and selection from newly constructed features is introduced. Selection from the extensive new search space is done using the cuckoo search-based optimization technique. To evaluate the results for suggested research techniques, an IoT-based dataset is used. This technique will lessen the exhaustive search space and generate a new organized search space that will improve the accuracy of machine learning tasks or mining classifiers.

2.6. Algorithm Result Analysis and Visualization. The mutated cuckoo search-based feature extraction is the proposed algorithm implemented in MATLAB, and results are visualized with graphs, plots, distributions, and statistical operation (mean, minimum, and maximum). Figure 9 represents the plots for newly generated space V' including n rows and m' reduced dimensions. Plots are relatively at distance and scattered that indicates that dimensions are chosen from the extensive search space. Subplot shows the number of iterations for random generations, and residuals are calculated for each dimension to check its weightage for selection. The dimension selection is constructed through levy flight (λ) and step size scaling factor ψ, and the steps are chosen randomly from the levy distribution.

Selection through CSFE searches through the extensive search space $\mathbb{R}^n$ and $\mathbb{R}^d$. Search is exponentially increasing as new objects enter the IoT and need an algorithm to modify itself to adjust for immediate changes, since data generated by IoT is continuous and needs a global optimization solution to enhance network efficiency.

Figure 10 shows the heavy-tailed distribution for CSFE produced for the input dataset of IoT-based Patients' CBC

```
IF (Attributes ! = Nominal A ttributes) Then
Go to Procedure_1 for Numeric Attributes
Else Start Procedure_2 for Numeric Attributes
Start Procedure_3 for Optimal Subset Selection
$\mathscr{F}_V = n/execution\ time(V) - execution\ time(V')$
$\mathfrak{L}_{v_i} = m - m'/execution\ time(v_i')$
Global Fit= $(\mathscr{G}) = \mathscr{F}_V$
Local Fitness = $\mathfrak{L}_{v_i}$ = max_ fitness
Initiate Population of $n$ vectors in $V$
For each vector $v_i(\forall i, i = 1, 2, 3, \cdots., n)$
    Calculate fitness for current vector $\mathfrak{L}(v_i)$
    For each dimension $d^j(\forall j,\ j = 1, 2, 3, \cdots., m)$ in
                While $(T \neq n)$ at time instance $t$, do
                    Find $v_i'$ through levy flight for $v_i$ in which $(v_i \neq 0)$
                        $v_i^j(t+1) = v_i^j(t) + \psi \bigotimes \mathscr{L}(s, \lambda$
                  Compute fitness for $v_i'$: $\mathfrak{L}(v_i')$
                  If $(\mathfrak{L}((v_i') > \mathfrak{L}(v_i))$ Then
                        Max_ fitness = $\mathfrak{L}(v_i') \wedge$ discard worst vector $\mathfrak{L}(v_i)$
                  Else max_ fitness = $\mathfrak{L}(v_i) \wedge$ discard worst vector $\mathfrak{L}(v_i')$
                    Set best fitted vector $v_i^j$ as a new reduced vector to $V'$
                  If( $F_v' > F_V$) Then
                      Set $V'$ as a new solution with reduced dimensions
                    Else build new vectors to get required fitness
```

Algorithm 4: Start procedure_CSFE.

Table 4: Representation for number of vectors and dimensions for each vector in dataset: internet of things-based patients' CBC results.

Dataset name	IoT-based patients' CBC results
Number of vectors	498
Number of dimensions	59

results. The highest peak represents the global optimization solution for IoT data as tail is exponentially increasing and consistently provides best fitness. The best selection is estimated trough DS and coherence estimation factor π . Area under the curve is not exponentially bounded, that means that as data becomes extinct, it increases the number of fitness values more close to best optimum. In Figure 11, the size of bars shows that most of the fitness values were globally best.

The term "Internet of Things" is considered to represent innovation that relies on both the resulting network by the integration of smart objects along with developed internet technologies and a variety of supporting devices, equipment, and machines that are important to ensure this technological evolution. Applications and services are developed to take advantage of these technologies for new business trends and offering daily life conveniences. IoT is an infrastructure based on networked smart objects and integrated networks as a supplement to internet services by ensuring availability for all kinds of services anytime and anywhere to anyone. It is emerging as a trend in which most of the objects in our surroundings will be on network in various forms. This shifts from conventional internet approaches to the internet for connecting physical objects that interact with each other and humans. These kinds of technologies are producing immense amounts of data, and it becomes critical when analytics and machine learning techniques are applied to make them intelligent with self-organizing capabilities.

2.7. Performance Comparison and Evaluation. Many standards are given in literature to test the efficiency, modality, or validity of any new optimization algorithm. After implementation of the CSFE algorithm, results are analyzed through global optimization test functions.

2.7.1. Rastrigin's Function. Rastrigin's function is a nonlinear optimization function introduced by Rastrigin as a 2-dimensional and extended by Mühlenbein et al. For a D dimensional space, this function can be illustrated as

$$\text{Ras}\left(V'\right) = Ad + \sum_{j=1}^{m}\left(v^j - A\cos\left(2\pi v^j\right)\right), \qquad (16)$$

where $A = 10$ and range for this function are $v^j \in [-5.12,5.12]$. Figure 12 shows results for Rastrigin's function for the D dimensional space with individual dimensions d_i. As compared to the cuckoo search-based feature extraction, the scattered plots show that the dimension space is still extensive, and each selected dimension is similar to previous selected which is not good to represent the whole data.

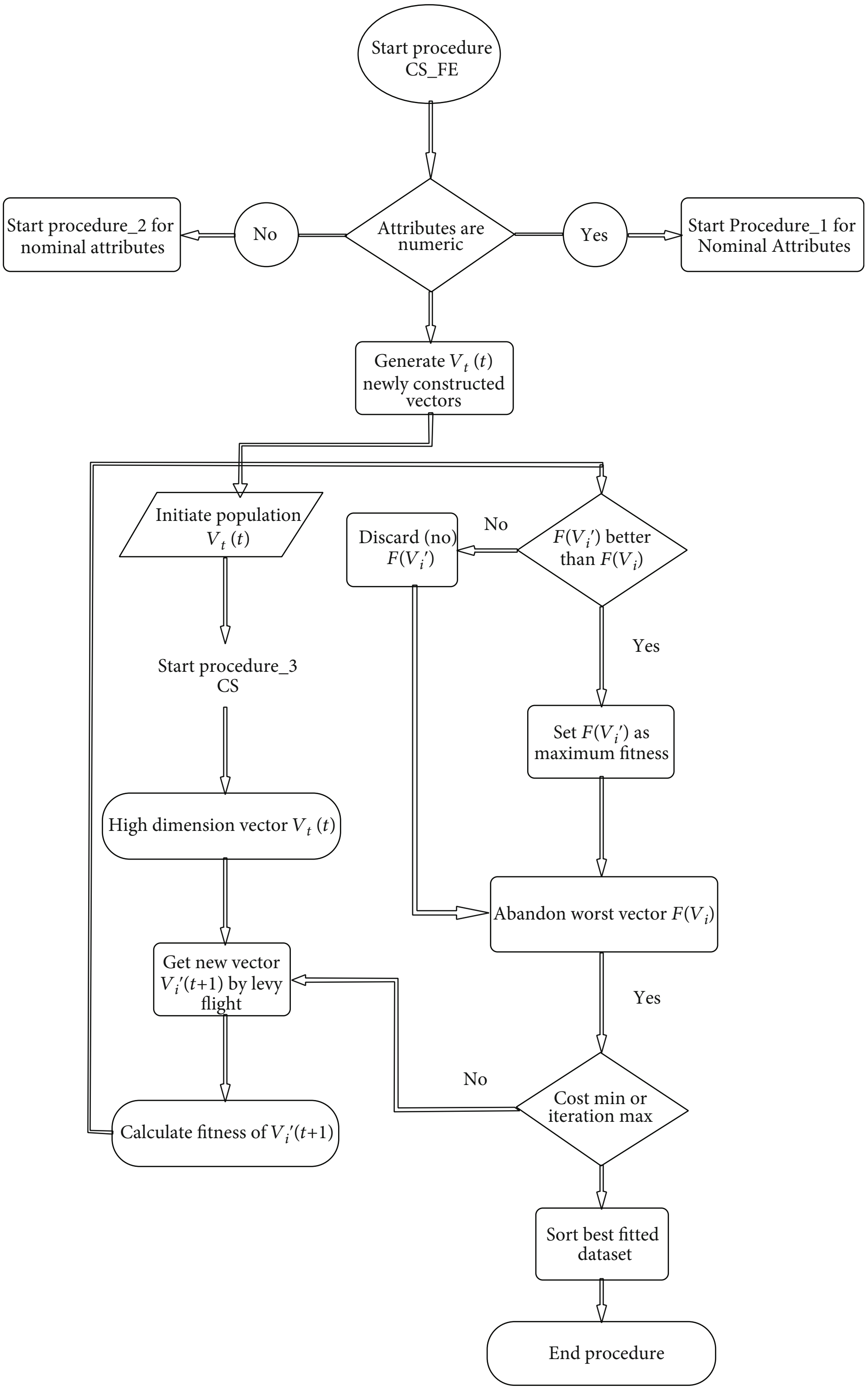
Start procedure CS_FE
Attributes are numeric
No
Start procedure_2 for nominal attributes
Yes
Start Procedure_1 for Nominal Attributes
Generate V_t (t) newly constructed vectors
Initiate population V_t (t)
Start procedure_3 CS
High dimension vector V_t (t)
Get new vector $V_i'(t+1)$ by levy flight
Calculate fitness of $V_i'(t+1)$
Discard (no) $F(V_i')$
No
$F(V_i')$ better than $F(V_i)$
Yes
Set $F(V_i')$ as maximum fitness
Abandon worst vector $F(V_i)$
Yes
No
Cost min or iteration max
Sort best fitted dataset
End procedure

FIGURE 8: Flowchart for the mutated cuckoo search-based feature extraction including procedures for nominal and numerical attributes while generation of final reduced feature subset through the cuckoo search-based feature extraction.

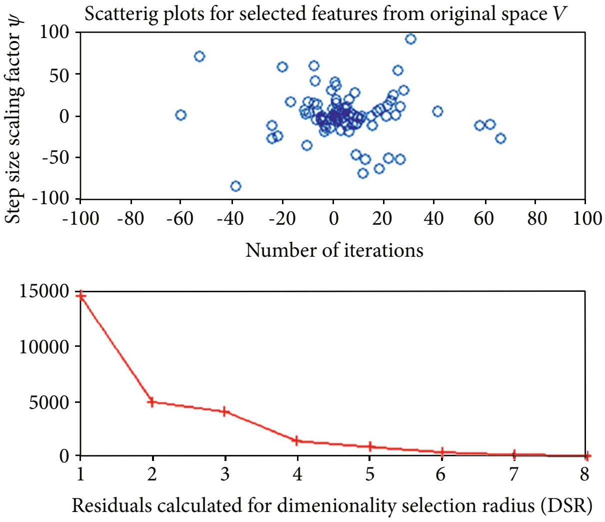

Figure 9: Scattering plot for the cuckoo search-based feature extraction, where each point represents the reduced and compressed amount of dimension extracted by CSFE.

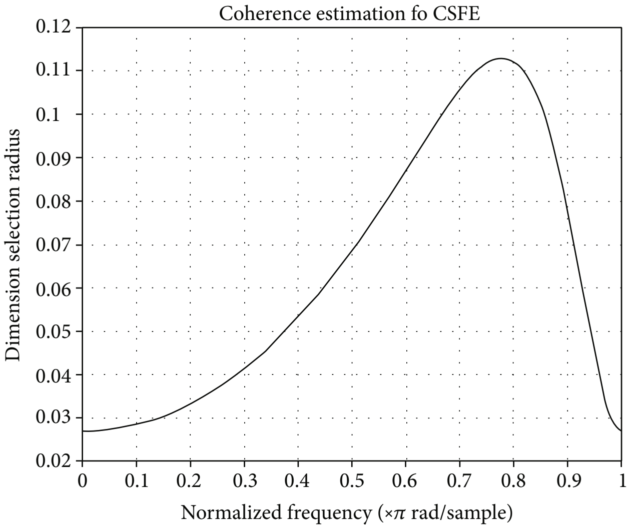

Figure 10: Heavy-tailed distribution: representing consistency for the cuckoo search-based feature extraction algorithm, whereas the highest peak level shows that most of the values chosen in each iteration are globally best.

2.7.2. Mccormick's Function. Mccormick's function is a benchmark to test an optimization algorithm. It is defined as given below:

$$\mathrm{Mck}\left(V'\right) = \mathrm{Mck}\left(v_i^j\right) = \sin\left(i + j\right) + \left(i - j\right)^2 - 1.5i + 2.5j + 1, \tag{17}$$

where i and j are vector number and dimension number sequentially. Searching range function is $-1.5 \le i \le 4$ and $-3 \le j \le 4$.

2.7.3. Cross-in-Tray Function. The cross-in-tray function is a continuous and multimodal test standard based on two-dimensional space initially and extended later on. The equation for this function takes the following form:

$$CIT\left(V'\right) = CIT\left(v_i^j\right) - 0.0001\left[\left|\sin\left(i\right)\sin\left(j\right)\exp \cdot \left(\left|100 - \frac{\sqrt{i^2 + j^2}}{\pi}\right|\right)\right| + 1\right]^{0.1}, \tag{18}$$

where V' is a D dimensional space and domain range for the cross-in-tray function that is $(i, j)\epsilon[-10, 10]$.

2.7.4. Rosenbrock Function. The Rosenbrock function is a nonlinear benchmark to test the performance of optimization problems, introduced by Howard H. Rosenbrock in 1960. It is also termed as Rosenbrock's Valley or Rosenbrock's banana function.

The mathematical definition for Rosenbrock is mentioned below:

$$\mathrm{Ros}\left(v_i^j\right) = \left(1 - i\right)^2 + 100\left(j - i\right)^2, \tag{19}$$

subjected to $(i - 1)^3 - j + 1$ and $x + y - 2 < 0$.

Range for the Rosenbrock function is $i \in [-1.5,1.5]$ and $j \in [-0.5,2.5]$. Figure 13 represents the fitness performance for these functions. Local maximum and global minimum for Rosenbrock are shown in Figure 14.

2.7.5. Easom Function. Easom is a multimodal and nonscalable test function to find the global minimum for a search space. It is defined as a following mathematical equation:

$$\mathrm{Easom}\left(v_i^j\right) = -\cos\left(i\right)\cos\left(j\right)\exp\left(-\left(\left(i - \pi\right)^2 + \left(j - \pi\right)^2\right)\right). \tag{20}$$

Search domain for the Eason function is $-100 \le i, j \le 100$. These test functions are used for the comparison of CSFE with particle swarm optimization and harmony search optimization algorithms.

Figure 15 displays the fitness curve for both local and global optimization. For N generations, global fitness is achieved at the early stage that indicates that the running time for the cuckoo search-based feature extraction will be minimal. Figure 16 provides a graph for the best cost value through harmony search (HS). Minimum cost for the firefly algorithm (FFA) is shown in Figure 17. As compared to HS, PSO, and FFA, CSFE gives the minimum cost value in minimum iteration and less elapsed time.

Table 5 provides an overview for the assessment of CSFE against PSO and HS. Performance is measured according to the minimum cost and elapsed time corresponding to each algorithm for maximum generations to calculate the best fitness value (cost). Here, CSFE is compared with few global

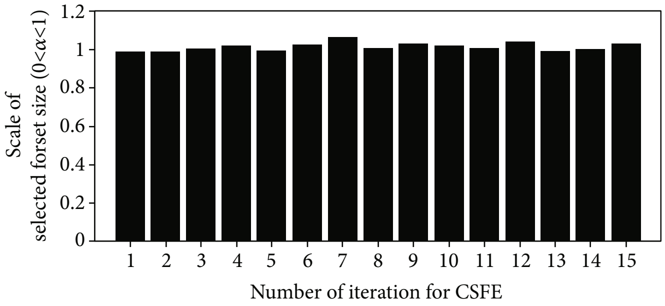

Figure 11: Bar representation for the feature selection during each iteration representing maximum value near to best maximum fitness.

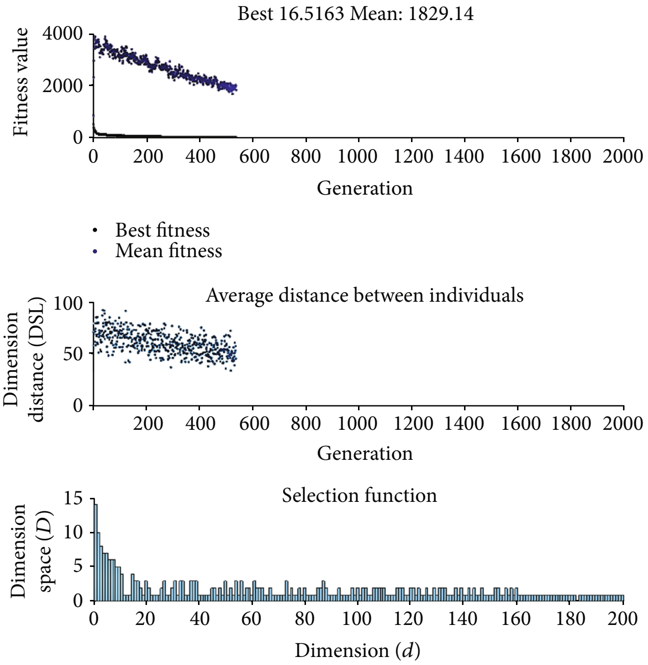

Figure 12: Fitness plot for Rastrigin's function: demonstration of best fitness and mean value for given number of dimensions and selection radius for each individual dimension with selection function for the overall D dimensional space.

optimization techniques. Particle swarm optimization introduced by Kennedy and Eberhart in 1995 provides best mutation results but it is not suitable for complex tasks as it is slow due to its complex structure and mutation.

As compared to CSFE, the peak and tail for the normal distribution are narrow and exponentially bounded from which the fitness values for PSO are generated. It means that local optimization points available through PSO are relatively rare. All global optimization techniques can provide the best possible solution for continuous data generated by the internet of things for a given problem of interest. Algorithms other than CSFE took more time to run and provide minimum cost in more number of iterations.

Another optimization technique is the firefly algorithm introduced by Xin-She Yang in 2008 inspired by the flashing behavior of fireflies. The random numbers for FA are drawn from the uniform distribution $[0, 1]$ with constant probability. Because of constant probability, this optimization technique is not appropriate for continuous and multiobjective optimization problems. In Table 6, PSO and CSFE are compared after 1000 runs for abovementioned test functions. Figure 18 shows the normal distribution $u(0, 1)$ for particle swarm optimization, where maximum height of tail shows that only best values are limited to this small area.

After the evolution of PSO and its limitations, harmony search (HS) was developed by Geem et al. in 2001 based on the concepts of music composition. In Table 7, mutated CSFE is compared with HS. After implementing CSFE and harmony search in MATLAB, the algorithms are compared according to their results generated by almost 1000 iterations.

After comparison, it is analyzed that CSFE has given more accuracy and global fitness for few mentioned test functions. Performance for both algorithms is measured corresponding to each test function as a pair of mean and standard deviation while the accuracy rate is given as

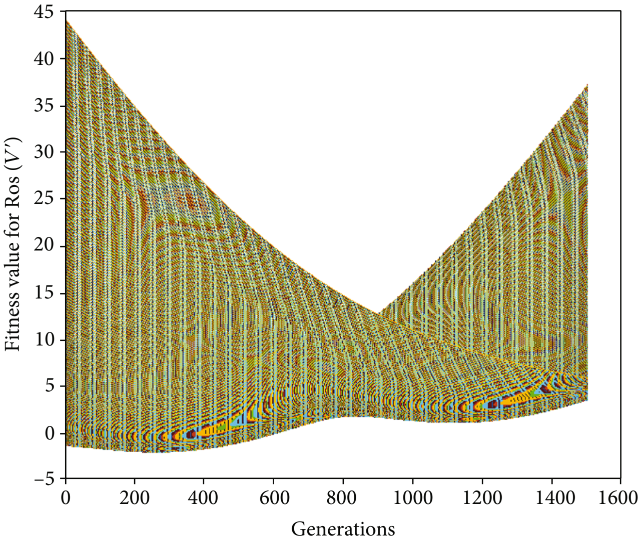

Figure 13: Plot for the Rosenbrock function: as the number of generations increases along the x-axis, the fitness value onto the y -axis is taking the form of local maxima for the given problem.

percentage. CSFE has given more success rate as compared to PSO and HS due to randomization and exploration. CSFE can converge to a global maximum state when required.

In literature, many test benchmarks are introduced to evaluate the performance of any new optimization technique. In this section, few test functions have been used to assess the functionality of the proposed cuckoo search-based feature extraction technique. CSFE is evaluated individually, and performance comparison is established using few global optimization techniques. At last, the accuracy rate delivered by CSFE is more stable and consistent than PSO and HS for global optimization.

2.8. Contribution. The major contribution of this paper is analysis of different aspects regarding dimensionality reduction and discussion of evolutionary approach with a special focus on cuckoo search algorithms. We have given the detailed discussion on existing dimensionality reduction techniques outlining feature selection and extraction. A comparison of heuristic and metaheuristic search methods are described in this paper. Cuckoo search is an optimization technique which is widely used in resource allocations in operations research, and here, we have used it for finding the attributes in data which may be dropped without affecting the meaning and information coherently in the database. The cuckoo living behavior is analyzed with reference to our own problem formulation. This algorithm also possesses some inherent features which limits its working for dimensionality reduction scenario in IoT and discussed in this paper. An algorithm corresponding to the problem of dimensionality reduction in internet of things scenario is formulated to further investigate the performance measures of the cuckoo search optimization algorithm.

We have transformed our problem into IoT vectors having distinct dimensions and explained the feature selection phenomenon. An algorithm is used for feature construction with numeric and nominal attributes. We have specially introduced mutated cuckoo search-based feature extraction algorithms to work with the generated dataset. We have analyzed the result of application of the cuckoo search algorithm on dimensionality reduction and compared its performance. This cuckoo search optimization algorithm proved to be very effective in feature selection and dimensionality reduction techniques and can be used in similar kinds of future applications.

3. Recommendation and Future Work

In information and communication technology, a number of innovative trends have emerged to facilitate humans, businesses, and industries with improved and efficient services. These next generation technologies manifest new challenges and complexities. Homogenize objects, wireless, and sensor networks, addressing schemes, and visualization build a multiplex structure of IoT. Data storage and analytics is one of the most important elements that formulate the network and emphasize dealing with unpredictable amounts of raw data collected by smart objects. Requirement for cost, time, and energy is directly proportional to an incredibly increasing amount of data. In the coming era, scientists are introducing "Green Computing Devices and Networks" with reduced cost, least time, and minimum energy resources.

3.1. Energy Proficient Algorithm for Green Internet of Things (GIoTs). The fundamental aim for IoT is to empower the smart world without greenhouse influences. To interact with real-world objects, these kinds of networks are equipped with numerous sensors, protocols, and communication technologies with high amounts of energy, sufficient cost, and complexity. Efficient algorithms are required for IoT services and applications to reduce the existing greenhouse effects or to build a new with minimum energy consumption. The proposed technique can be used to get maximum accuracy for any machine learning task with minimum cost without complex computation.

Cuckoo search-based techniques are suitable to build future GIoT with maximum accuracy and lower complexity. Cuckoo search is a global optimization technique and provides global maximum solutions for real-time systems (IoT) who generate continuous and high amounts of data with massive dimensionalities. The reason for this recommendation is that CS does not implicate a lot of mathematical computation which will definitely decrease the overall complexity of the system.

3.2. Future Internet of Things for Patient's Monitoring. In medical scenarios, patients are monitored manually, e.g., patient's history, current disease, and their daily health report. Individual files with distinct patient numbers are maintained including some health parameters such as heart rate, blood pressure temperature, and blood samples. These records are assessed by the concerned doctor for further treatment. Instead of all these manual procedures, a smart health monitoring IoT device can perform these actions smartly without extensive human intervention.

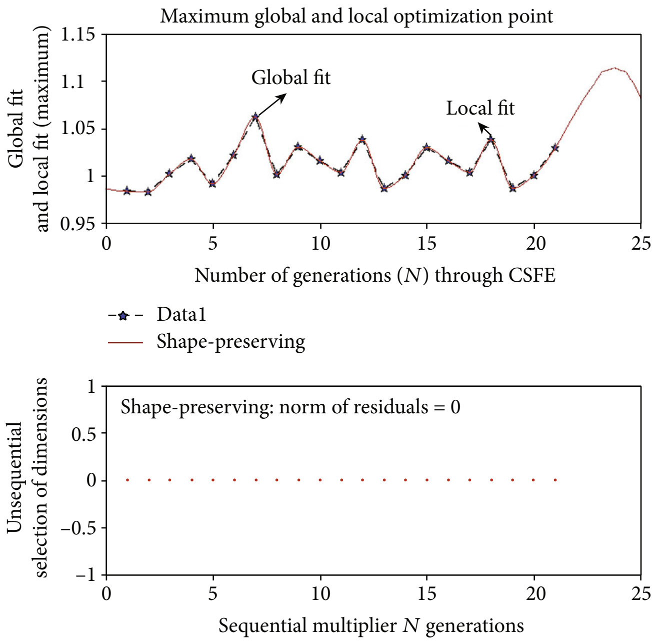

Figure 14: Curve fitting plot for the cuckoo search-based feature extraction: indication of global and local maximum for N generations. Subplot provides the nonsequential flow during multiplication of each element at time t.

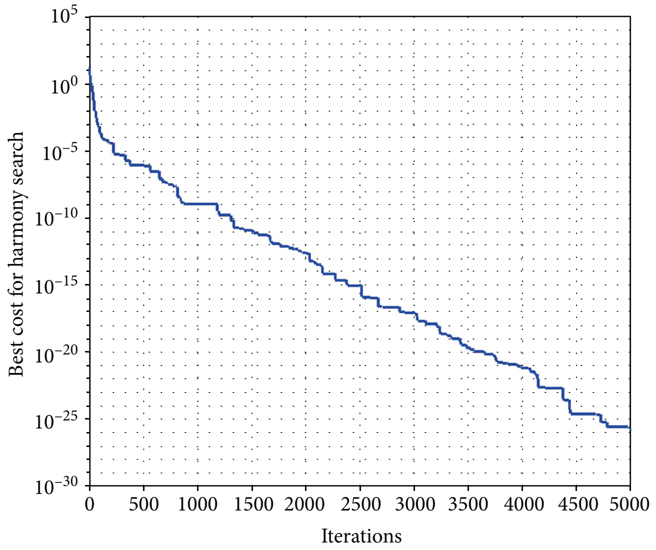

Figure 15: Cost estimation for harmony search shows the best cost value for harmony search after 4500 iteration, and it will increase overall algorithm running time.

The architecture in Figure 19 demonstrates an abstract layout for a complete health care system (CHCS) based on an intelligent IoT device to monitor health of patients. The patient's record managed and handled by smart IoT devices is available to concerned doctors and for users as well. A situation handled by an IoT device for patient's monitoring can be a complete blood count (CBC) report of the patient. A continuous blood report will be generated by this intelligent IoT device and whenever there is an alarming situation some action would be triggered.

In spite of tremendous efforts made regarding ICT, there is a need to execute the emerging and evolutionary trends without negative environmental effects to compensate for the increasing amount of data with a smaller amount of energy and computations. Indispensable measurements are required to minimize negative technological effects on the health and society. It is concluded that the mutated cuckoo search-based feature extraction can be an advantageous approach towards the recent internet of things and for future green internet of things as well. Moreover, it can be

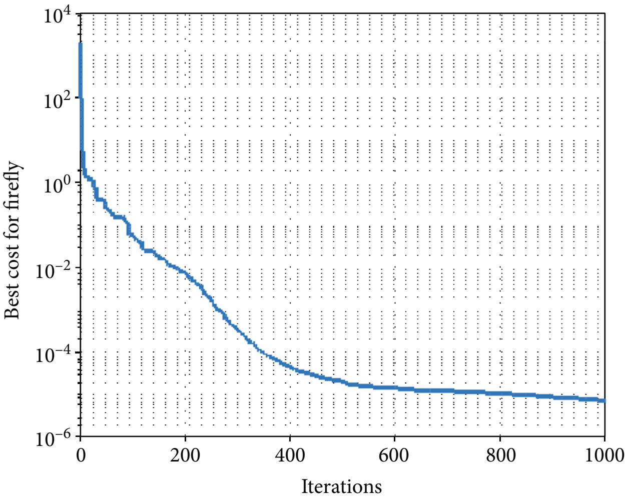

Figure 16: Cost estimation for the firefly algorithm: the best cost value is available after a large number of iterations.

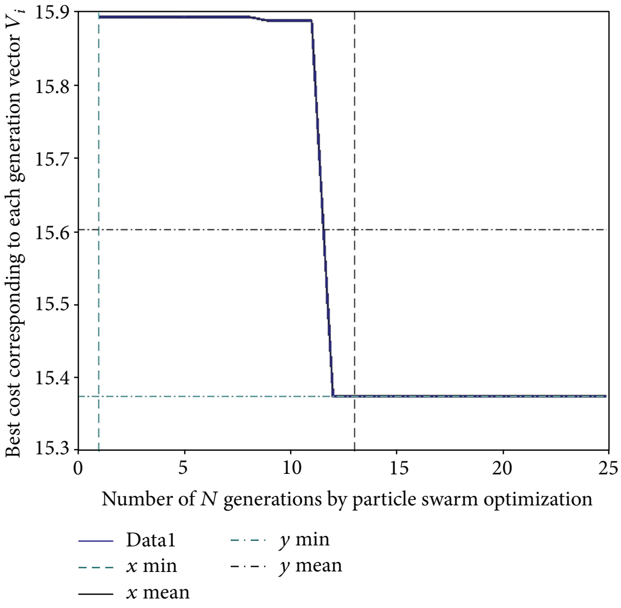

Figure 17: Minimum and maximum cost by particle swarm optimization to generate transformed datasets v_i'.

Table 5: Summarized performance table for HS, PSO, CSFE, and firefly algorithm: analysis of metaheuristic global optimization algorithms with respect to minimum cost and elapsed time.

Algorithm	Best cost value	Elapsed time
Cuckoo search-based feature extraction	4.0788e-26	11.228997 s
Particle swarm optimization	14.375e-4	42.284616 s
Harmony search	7.9050e-06	12.435746 s
Firefly algorithm	7.2753e-06	21.244669 s

Table 6: Comparison of particle swarm optimization and cuckoo search-based feature extraction.

Algorithm	Particle swarm optimization	Cuckoo search-based feature extraction
Rastrigin	4112 ± 279 (93%)	1025 ± 103 (97%)
Mccormick	7877 ± 503 (97%)	3421 ± 209 (100%)
Cross-in-tray	13901 ± 2131 (95%)	4239 ± 276 (100%)
Rosenbrock	90571 ± 3036 (100%)	42330 ± 9211 (100%)
Easom	55491 ± 4023 (97%)	78901 ± 1043 (98%)

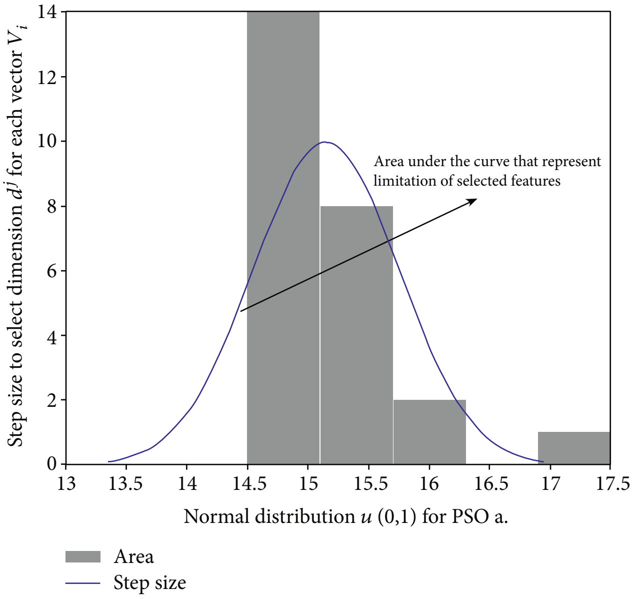

Figure 18: Normal distribution $u(0,1)$ for particle swarm optimization, where maximum height of tail shows that only best values are limited to this small area.

Table 7: Comparison of harmony search and cuckoo search-based feature extraction.

Test functions	Harmony search	Cuckoo search-based feature extraction
Rastrigin	54412 ± 2301 (94%)	1025 ± 103 (97%)
Mccormick	60925 ± 3324 (87%)	3421 ± 209 (100%)
Cross-in-tray	70215 ± 1051 (100%)	4239 ± 276 (100%)
Rosenbrock	39571 ± 3036 (100%)	42330 ± 9211 (100%)
Easom	10033 ± 4023 (97%)	78901 ± 1043 (98%)

utilized to enhance the performance for future IoT devices for uninterrupted monitoring of patients. As a brief description, cuckoo search optimization is a metaheuristic approach and applicable to situations where the system is in the local state or will grow up towards a global phenomenon in future.

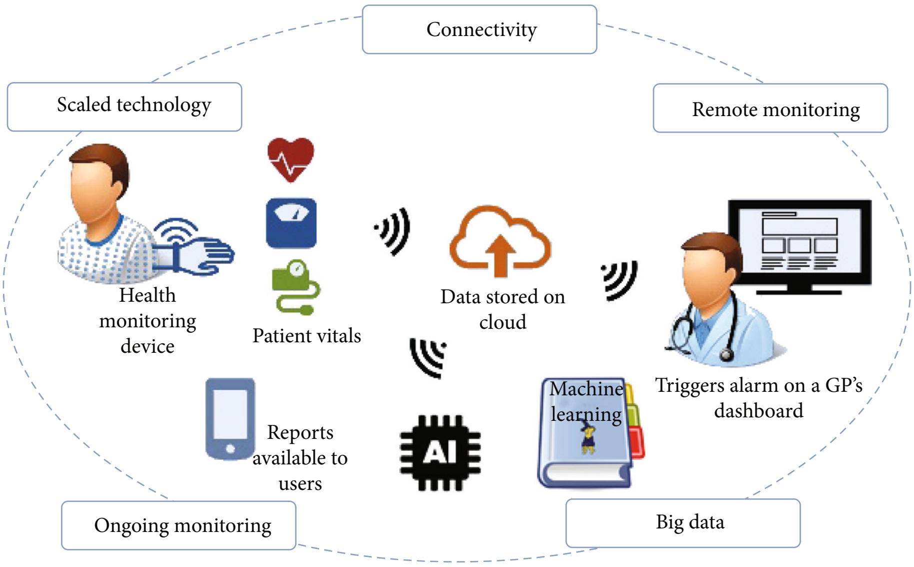

Figure 19: A future IoT for patient's monitoring: an overview for an internet of things-based complete health care system with big data analytics and machine learning techniques.

4. Conclusion

Although IoT networks have emerged and performing exacting tasks competitively but to stable their performance, enhancement for future challenges is mandatory. It becomes possible when networks are not only fast with preeminent servers but also smart enough to cope with unpredictable circumstances. Introducing efficient and global optimization algorithms can help to achieve this target. In this research, a metaheuristic global optimization algorithm is established to reduce dimensions of outdoor data for IoT. The cuckoo search-based feature extraction is a mutated algorithm that organizes itself according to the unpredictable amount of data and produces a new and an enhanced feature space. The newly generated feature space and proposed algorithm benefit for improving the accuracy for classifiers and mining algorithms. It also makes the algorithm computationally feasible, flexible, and efficient for obtaining the target of convergence. This mutated algorithm is further generalizable to IoT indoor activities as the need of near future. The scenario to train the IoT network for future challenges and increase sphere of knowledge is also discussed. Among all abovementioned facts, CSFE can perform all activities with minimum cost and less time as evaluated. This algorithm can be further improved for multiobjective optimization problems. It can provide tremendous support to build an IoT-based smart world with no negative impacts and minimum resources.

Data Availability

Dataset is generated through IoTify. This is a web-based platform for simulation to develop IoT application by using virtual hardware devices, e.g., sensors. It facilitates like a virtual lab and enables to create a virtual IoT device in JavaScript. The IoTify database is generated using JavaScript object naming (JSON) with extension.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] O. Vermesan, P. Friess, P. Guillemin et al., "Internet of things strategic research roadmap," *Internet things-global Technol. Soc. trends*, vol. 1, no. 2011, pp. 9–52, 2011.

[2] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, 2014.

[3] H. Chiroma, T. Herawan, I. Fister Jr. et al., "Bio-inspired computation: Recent development on the modifications of the cuckoo search algorithm," *Applied Soft Computing*, vol. 61, pp. 149–173, 2017.

[4] O. Vermesan, P. Friess, P. Guillemin et al., *Internet of things strategic research and innovation agenda*, River Publ. Ser. Commun., 2013.

[5] K. K. Patel and S. M. Patel, "Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges," *International Journal of Engineering in Computer Science*, vol. 6, no. 5, 2016.

[6] Y. Yoo, "Computing in Everyday Life: A Call for Research on Experiential Computing," *MIS Quarterly*, vol. 34, no. 2, pp. 213–231, 2010.

[7] S. M. Thampi, O. Marques, S. Krishnan, K.-C. Li, D. Ciuonzo, and M. H. Kolekar, "Advances in signal processing and intelligent recognition systems: 4th International Symposium SIRS 2018, Bangalore, India, September 19-22, 2018," *Revised Selected Papers*, vol. 968, 2019.

[8] W. Fan, W. Lee, S. J. Stolfo, and M. Miller, "A multiple model cost-sensitive approach for intrusion detection," *in European conference on machine learning*, pp. 142–154, Springer, Barcelona, Catalonia, Spain, 2000.

[9] N. Zhang, H. Chen, X. Chen, and J. Chen, "Semantic framework of internet of things for smart cities: Case studies," *Sensors*, vol. 16, no. 9, p. 1501, 2016.

[10] P. Maghouli, S. H. Hosseini, M. O. Buygi, and M. Shahidehpour, "A Multi-Objective Framework for Transmission Expansion Planning in Deregulated Environments," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 1051–1061, 2009.

[11] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communication Surveys and Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[12] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of Network and Computer Applications*, vol. 98, pp. 27–42, 2017.

[13] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel, "Data quality in internet of things: a state-of-the-art survey," *Journal of Network and Computer Applications*, vol. 73, pp. 57–81, 2016.

[14] D. Del Vecchio and D. Carter, "Enhancing presence awareness in instant messaging," in *in Innovations Through Information Technology: 2004 Information Resources Management Association International Conference*, vol. 1, p. 242, New Orleans, Louisiana, USA, 2004.

[15] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.

[16] J. Z. Kolter and M. J. Johnson, "REDD: a public data set for energy disaggregation research," in *in Workshop on data mining applications in sustainability (SIGKDD)*, vol. 25, pp. 59–62, San Diego, CA, 2011.

[17] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *Journal of Big Data*, vol. 2, no. 1, 2015.

[18] J. Kim, *Mind in a Physical World: An Essay on the Mind-Body Problem and Mental Causation*, MIT press, 1998.

[19] R. M. Ramadan and R. F. Abdel-Kader, "Face recognition using particle swarm optimization-based selected features," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 2, no. 2, pp. 51–65, 2009.

[20] J.-Y. Yeh, T.-H. Wu, and C.-W. Tsao, "Using data mining techniques to predict hospitalization of hemodialysis patients," *Decision Support Systems*, vol. 50, no. 2, pp. 439–448, 2011.

[21] C. A. Bhatt and M. S. Kankanhalli, "Multimedia data mining: state of the art and challenges," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 35–76, 2011.

[22] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[23] P. Ristoski and H. Paulheim, "Semantic Web in data mining and knowledge discovery: A comprehensive survey," *Journal of Web Semantics*, vol. 36, pp. 1–22, 2016.

[24] X. Xu, L. He, H. Lu, L. Gao, and Y. Ji, "Deep adversarial metric learning for cross-modal retrieval," *World Wide Web*, vol. 22, no. 2, pp. 657–672, 2019.

[25] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, 2017.

[26] B. Rasti, D. Hong, R. Hang et al., "Feature extraction for hyperspectral imagery: the evolution from shallow to deep," 2020, http://arxiv.org/abs/2003.02822.

[27] D. Hong, N. Yokoya, and X. X. Zhu, "Learning a robust local manifold representation for hyperspectral dimensionality reduction," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 2960–2975, 2017.

[28] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Learning to propagate labels on graphs: an iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 35–49, 2019.

[29] A. Keramati, R. Jafari-Marandi, M. Aliannejadi, I. Ahmadian, M. Mozaffari, and U. Abbasi, "Improved churn prediction in telecommunication industry using data mining techniques," *Applied Soft Computing*, vol. 24, pp. 994–1012, 2014.

[30] W. Ding, X. Jing, Z. Yan, and L. T. Yang, "A survey on data fusion in internet of things: towards secure and privacy-preserving fusion," *Information Fusion*, vol. 51, pp. 129–144, 2019.

[31] B. Gendron and T. G. Crainic, "Parallel branch-and-branch algorithms: survey and synthesis," *Operations Research*, vol. 42, no. 6, pp. 1042–1066, 1994.

[32] L. Huang, S. Ding, S. Yu, J. Wang, and K. Lu, "Chaos-enhanced cuckoo search optimization algorithms for global optimization," *Applied Mathematical Modelling*, vol. 40, no. 5–6, pp. 3860–3875, 2016.

[33] B. He, S. Shah, C. Maung, G. Arnold, G. Wan, and H. Schweitzer, "Heuristic search algorithm for dimensionality reduction optimally combining feature selection and feature extraction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 2280–2287, 2019.

[34] B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "Optimal selection of components value for analog active filter design using simplex particle swarm optimization," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 4, pp. 621–636, 2015.

[35] S. M. Piryonesi and M. Tavakolan, "A mathematical programming model for solving cost-safety optimization (CSO) problems in the maintenance of structures," *KSCE Journal of Civil Engineering*, vol. 21, no. 6, pp. 2226–2234, 2017.

[36] X. Zhou, C. Yang, and W. Gui, "Nonlinear system identification and control using state transition algorithm," *Applied Mathematics and Computation*, vol. 226, pp. 169–179, 2014.

[37] A. R. Malisia and H. R. Tizhoosh, "Applying Opposition-Based Ideas to the Ant Colony System," in *in 2007 IEEE Swarm Intelligence Symposium*, pp. 182–189, Honolulu, Hawaii, USA, 2007.

[38] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *in 2007 IEEE Swarm Intelligence Symposium*, pp. 120–127, Honolulu, Hawaii, USA, 2007.

[39] X. Zhou, D. Y. Gao, and C. Yang, "A Comparative Study of State Transition Algorithm with Harmony Search and Artificial Bee Colony," in *Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013*, pp. 651–659, China, 2013.

[40] X.-y. YANG and W. U. Dan, "Atomic simulations for surface-initiated melting of Nb(111)," *Transactions of the Nonferrous Metals Society of China*, vol. 19, no. 1, pp. 210–214, 2009.

[41] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, *Swarm Intelligence and Bio-Inspired Computation*, Newnes, 2013.

[42] R. B. Payne, "The Ecology of Brood Parasitism in Birds," *Annual Review of Ecology and Systematics*, vol. 8, no. 1, pp. 1–28, 1977.

[43] H. Rathore, *Mapping Biological Systems to Network Systems*, Springer, 2016.

[44] N. Davies, *Cuckoos, Cowbirds and Other Cheats*, A&C Black, 2010.

[45] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[46] K. Maulik and B. Zwart, "Tail asymptotics for exponential functionals of Lévy processes," *Stochastic Processes and their Applications*, vol. 116, no. 2, pp. 156–177, 2006.

[47] S. Fong, "Opportunities and challenges of integrating bio-inspired optimization and data mining algorithms," in *in Swarm Intelligence and Bio-Inspired Computation*, pp. 385–402, Elsevier, 2013.

[48] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communication Surveys and Tutorials*, vol. 19, no. 4, pp. 2392–2431, 2017.

[49] A. B. Nasser, A. R. A. Alsewari, and K. Z. Zamli, "Tuning of cuckoo search based strategy for t-way testing," *in International conference on electrical and electronic engineering*, vol. 9, p. 10, 2015.

[50] X.-S. Yang, S. F. Chien, and T. O. Ting, "Computational Intelligence and Metaheuristic Algorithms with Applications," *The Scientific World Journal*, vol. 2014, Article ID 425853, 4 pages, 2014.

[51] G. Wang, "A comparative study of cuckoo algorithm and ant colony algorithm in optimal path problems," *MATEC Web of Conferences*, vol. 232, p. 03003, 2018.

[52] V. Mayer-Schonberger and K. Cukier, *Big Data: The Essential Guide to Work, Life and Learning in the Age of Insight*, Hachette UK, 2013.

[53] P. Raj, T. Poongodi, B. Balusamy, and M. Khari, *The Internet of Things and Big Data Analytics: Integrated Platforms and Industry Use Cases*, CRC Press, 2020.

[54] P. Tan, X. Wang, and Y. Wang, "Dimensionality reduction in evolutionary algorithms-based feature selection for motor imagery brain-computer interface," *Swarm and Evolutionary Computation*, vol. 52, p. 100597, 2020.

[55] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, 2018.

[56] S. Pittner and S. V. Kamarthi, "Feature extraction from wavelet coefficients for pattern recognition tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 1, pp. 83–88, 1999.

[57] M. R. Genesereth and N. J. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, 2012.

[58] D. Mladenić, "Feature selection for dimensionality reduction, in International Statistical and Optimization Perspectives Workshop," in *Subspace, Latent Structure and Feature Selection*, pp. 84–102, Berlin, Heidelberg, Springer, 2005.