IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 March 2024, accepted 31 March 2024, date of publication 5 April 2024, date of current version 25 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3385628

RESEARCH ARTICLE

KIHCDP: An Incremental Hierarchical Clustering Approach for IoT Data Using Dirichlet Process

ABISHI CHOWDHURY[1], (Member, IEEE), AMRIT PAL[1], (Member, IEEE), ASHWIN RAUT[2], AND MANISH KUMAR[2], (Senior Member, IEEE)

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[2]Department of Information Technology, Indian Institute of Information Technology Allahabad, Prayagraj 211015, India

Corresponding author: Abishi Chowdhury (abishi.chowdhury@vit.ac.in)

ABSTRACT Internet of Things (IoT) devices are constantly producing vast amounts of data, necessitating efficient storage and processing to extract useful information. However, the models used to extract relevant information from IoT data are often hindered by the lack of useful data and the ever-changing distribution of this data. This paper introduces an incremental data clustering technique on a continuous stream of data through a Dirichlet process-based approach that is adept at handling the formation of clusters in streaming data. The complete approach is twofold; firstly, it starts with an estimated distribution of data and allocates an incoming data point to the estimated data distribution. Secondly, it refines the estimated data distribution after the allocation of the current point and over the subsequent arrival of data points. The influx of data leads to greater challenges in determining clusters for incoming points and preserving the current clusters for improved decision-making. In this context, our proposed approach deals with the increasing amount of data using a selective elimination technique on both existing and incoming data. To assess the performance of the proposed approach, benchmark experiments have been performed using benchmark datasets. The results of the experiments demonstrate that the proposed model has a gain ranging from 2% to 4% as compared to the existing state-of-the-art and recent adaptive clustering approaches in terms of clustering accuracy with incremental data addition and variable clustering parameters. The proposed method shows a high gain in terms of running time ranging from 2% to 20% as compared to the existing approaches depending on the data reduction parameter. Furthermore, research findings through this work indicate that it is possible to set a trade-off between accuracy and running time by adjusting the elimination parameter depending on the requirements of the considered application.

INDEX TERMS Internet of Things, data clustering, streaming data, K-means clustering, Dirichlet process, incremental learning.

I. INTRODUCTION

The number of Internet of Things (IoT) devices is growing rapidly, resulting in a significant change in the sources of data generation. IoT gadgets, for example, industrial sensors, self-driving vehicles, cell phones, and so on, are consistently creating data. It is needless to say this data should be collected and interpreted efficaciously to make certain decisions and realize newfangled products as shown in Figure 1. In this context, clustering is a prominent unsupervised learning technique for extracting knowledge from the collected data. Cluster analysis is widely used in many applications, such as pattern recognition, market research, image processing, and others [1], [2]. Fundamentally, it perceives the structure in a collection of unlabeled datasets and facilitates understanding the overall distribution of the data. IoT data is in general a continuous stream [3] of data points generated by the allied sensor devices. This type of data requires an adaptive learning approach that can adapt to the changing distribution of the data. Clustering over this type of data is a challenging task [4] due to the evolving shape of the clusters and their count. There are various types of clustering [5] algorithms that

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

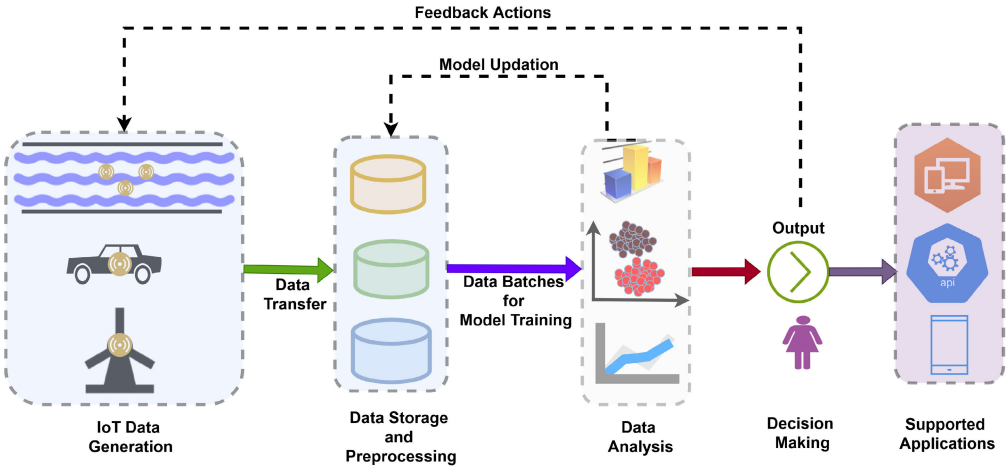


FIGURE 1. An overview of IoT data generation from different sources and its transformation to information for decision-making and user applications.

can be broadly classified as partition-based, hierarchical, and density-based. The most popular partition-based algorithm is the K-Means algorithm [6], which uses centroids represented by mean vectors to model the clusters. Finding an optimal cluster count is a tedious task, even for experts in the domain. Various techniques, such as the elbow method, can be employed to decide on the most suitable cluster count; however, it is too computationally expensive to be used in practical applications. Another disadvantage of K-Means clustering [6] is that it cannot find outliers (points that are not expected in the dataset) present in the dataset. It assigns outliers to the heads of the corresponding clusters resulting in increasing errors. The whole K-Means process must be repeated whenever new data points are added to the dataset. Hierarchical-based clustering includes agglomerative clustering, BIRCH (Balanced Iterative Reduced Clustering using Hierarchies) [7], [8], and Bayesian Hierarchical Clustering (BHC) [9], [10]. Hierarchical clustering is characterized by the development of a tree-like structure for the entire dataset. The complexity of generating and maintaining the tree structure increases when there is a rise in the number of clusters. In this case, BHC constructs a tree structure of the internal nodes to symbolize each cluster, yet these do not precisely reflect the overall significance of the dataset. This issue can be addressed by using incremental Bayesian hierarchical clustering [11], which has the benefit of combining data that changes over time and approximating the number of clusters. Density-based clustering techniques, such as DBSCAN [12], [13], are used to identify high-density regions in data based on a minimum distance measure. However, this kind of algorithm requires knowledge of the distance measure and can be difficult to implement when there are large variations in the density of the clusters.

Partition-based clustering approaches are used mainly when the clusters do not overlap with each other. Density-based clustering algorithms are highly adjustable to the irregular shapes of clusters. Nevertheless, both of these types of clustering algorithms necessitate prior knowledge of parameters. Finding these parameters from IoT data streaming is a challenge because of its growing size and drifting nature. Hierarchical clustering approaches take the lead over other approaches in the context of required prior information. Hierarchical clustering, particularly BHC, is a good choice for IoT streaming data due to its less dependence on pre-defined parameters and its ability to adjust itself to the data's distribution as it progresses through its continuous iterations. BHC approximates a probabilistic model of the data as it cycles through the dataset. BHC is based on posterior probability, which is defined by the Bayes theorem and works in a bottom-up fashion. It initializes all data elements to their individual clusters and then iteratively merges the clusters until no further merging occurs. Dirichlet process mixer model (DPMM) [14] acts as a base for this type of clustering technique. DPMM provides the prior information for the allocation of new data points to the existing clusters.

The impetus behind this research is based on the fact that the most commonly used clustering algorithms require prior knowledge of the clusters and are not able to adjust to the changes in the data's distribution and size. The clustering approaches are iterative, which in turn, results in over-dependence on the computing power of the current systems. Reducing this dependence while maintaining appropriate clustering accuracy is a challenge that calls for new research.

Reviewing the aforementioned challenges, this manuscript examines the application of BHC and DPMM techniques in IoT streaming data, which enable the estimation of the number of clusters and the assignment of new data points to the existing clusters. However, the calculation of intermediate steps, such as the Gamma function of a larger number of readings, is an intricate task. Thus, in this paper, we have proposed a kernel-based incremental hierarchical clustering approach for IoT data using Dirichlet process (KIHCDP) that further reduces the dependence on the computing power of the current system.

Paper Contributions: The key contributions of this manuscript are listed below:

- KIHCDP handles the evolving distribution of the data generated by IoT devices and adapts the cluster's information and data to each cluster as the incremental addition of data points takes place.
- No prior determination of the number of clusters is necessary during clustering, as it updates the information of the cluster by increasing and decreasing the number of clusters using the join-and-divide phase of the proposed approach.
- KIHCDP improves memory utilization by reducing the overhead of maintaining data points in system memory by selectively removing the unwanted points from the existing pool of data points.
- It retains derived information while eliminating data points from the data pool and uses this information to further evolve the clustering process.

Roadmap: The succeeding sections in this manuscript evince the progression towards the objectives of this research. Section II outlines the related work done in this field. Section III provides the necessary background information for the proposed approach. Section IV describes the proposed approach, which involves assigning points to clusters and reassessing the clusters. Section V presents a comprehensive experimental analysis of the proposed approach using benchmark datasets. Section VI discusses a few noteworthy opinions based on experiments and analyses done, and Section VII marks the concluding remarks.

II. RELATED WORK

Deciding the number of clusters before applying the clustering approach results in an accurate estimation of the final clusters. If the number of clusters is known, many approaches are available to perform clustering, such as the K-Means [6], [15] clustering algorithm. In machine learning, clustering is also addressed by models such as Gaussian mixture models (GMM) [16] which is derived from the Bayes parametric model. Representative models such as DPMM [14], Chinese Restaurant Process (CRP) [17],

and Hierarchical Dirichlet Process (HDP) [18] are also used for clustering. Frequently used in clustering, Bayesian non-parametric methods offer a flexible framework to handle an unknown number of clusters. Much research has been conducted on clustering algorithms while considering static datasets; however, due to the uncertainty of the amount and rate of arrival, as well as the gradual changes over time, these algorithms are not suitable for data streams.

Clustering approaches are widely used in remote sensing considering high-dimensional data. Remote sensing data is prone to noise, and handling this noise is a challenge. In the case of high-dimensional data, subspace clustering methods can be used to perform clustering. Using a mixture of Gaussian noise in high-dimension, data clustering can be performed [19]. Identifying different segments and performing further classification on new images using deep learning models has recently gained attention due to the high-performance capabilities of deep learning models. This method is used for the classification of hyperspectral images using deep learning frameworks. It helps to understand and classify the given land area [20].

IoT data stream requires adaptive clustering approaches to efficiently perform clustering on the data [21]. This area has gained substantial attention in the early stages of IoT data processing [3], [22]. Determining the amount of clusters in an IoT data stream is a difficult undertaking and poses challenges. An adaptive clustering approach first estimates clusters based on the characteristics distribution using the symbolic aggregate approximation (SAX) [23] algorithm and then uses online clustering to group the data points coming from a data stream. The dynamic nature of the underlying distribution of the data requires a change in the learning model. In this regard, probabilistic estimation of the data distribution in a dynamic environment has recently gained attention. Approaches such as dynamic Gaussian mixture model-based probabilistic clustering (GDPC) [24] have been developed to ensure the adaptiveness of the learned model. GDPC uses the Expectation Maximization algorithm for the estimation of model parameters and uses drift detection approaches to detect any concept drift in the data. An alternate strategy [25] applies the idea of micro-clusters to cluster the changing data set. A buffer is utilized to store the less significant microclusters, and an energy update technique is used to remove the unnecessary microclusters from the buffer. To perform online clustering on the streaming data, a non-parametric Dirichlet model can also be used. A context-sensitive scheme [26] uses a representation of the semantic term that establishes the relationship between the data points. The approach uses an episodic inference approach to minimize cluster sparsity. Dirichlet mixture model can be used for density estimation [27] to group high-dimensional data. Dirichlet process (DP) has also been introduced with deep learning to perform clustering [28] of the data. For better transportation planning [29], DPMM is used to group passengers visiting railway stations. DP is also applied for the detection of anomalies [30] in traffic with the help of super-pixels in the image data. It has also been applied to the clustering of a continuous stream of text data [31]. Although numerous studies have already been done on similar problems, the literature has revealed a lack of research in the area of ably managing stream data for IoT devices.

III. BACKGROUND

This section presents two main concepts that are used in this manuscript using the notation as shown in Table 1.

A. DIRICHLET PROCESS

The Dirichlet process is defined as a mixture model, using which the incoming data can be counted and grouped in a finite number of clusters. The Dirichlet process allows

TABLE 1. Description of the notations used in the paper.

Notation	Description
y_t	Datapoint at time t
y_k	Set of points in k cluster
K	Current total number of clusters
Y	Dataset
k^*	Predicted cluster for a new data point
N	Current total number of points
N_k	Current number of points in cluster k
N_{k_i}	Current number of points in cluster k_i
N_{k_j}	Current number of points in cluster k_j
m_k	Mixing weight proportion of a cluster
Gamma or γ	Learning rate
$Pr(y_t \mid Y)$	Probability distribution of data point at t time
$Pr(y_t, k^* \mid Y)$	Probability for current cluster for new point
$Pr(y_t, k+1 \mid Y)$	Probability for new cluster
Y_{-k}	The set of datasets except Y_k
D	Decision Step
$Pr(Y_k)$	Probability of partition Y_k
$Pr(\beta_k)$	Probability of participation of a cluster
X_{Divide}	Divide parameter
$X_{Join}(Y_{k_i}, Y_{k_j})$	Join parameter for clusters Y_{k_i} and Y_{k_j}
H	Distribution of points
G	Distribution of latent parameters
Π	Mixing proportion
η	Hyper parameter
θ	Latent parameters
ξ	Similarity Threshold

a prior estimation of the number of clusters in case of continued growth in the data. This process is stochastic in nature [32] because nothing is determined at any stage except a few variables, unlike a deterministic process in which every variable of a model is defined specifically. This stochastic property of the model allows it to be used in Bayesian non-parametric models [33] of data to analyze the data and specify them in clusters. The model can be defined as a process where data points are given as input and processing of those data points is done to create clusters or, in other words, components. A Bayesian mixture model (BMM) that contains K components can be parametrized using η and Π to map it to a specific cluster, as expressed in equation 1. This is why the model is said to have an infinite number of components.

$$\Pi \,\Big|\eta \sim \mathrm{Dir}\left(\frac{\eta}{K}, \ldots., \frac{\eta}{K}\right) \quad \theta^*\mathrm{k}\Big|\,\mathrm{H} \sim \mathrm{H} \tag{1}$$

Then using conjugate prior property [34], [35], the process of clustering these data points can be done using equation 2

$$\upsilon_i\,|\Pi \sim \mathrm{Mult}(\Pi) \quad x_i|\,\upsilon_i, \left\{\theta^*\mathrm{k}\right\} \sim \mathrm{F}\left(\theta^*\upsilon_i\right) \tag{2}$$

Mixing proportion is represented by Π, the pseudocount is represented by η, H represents the prior distribution, $\theta * k$ represents the component parameters, and component distribution is represented by $F(\theta)$. For a large value of K, the way Dirichlet prior over Π is parameterized, the modeling of n data points is independent of K and can be approximated in $O(\eta \log \mathrm{n})$ [36].

This step ensures that as K approaches infinity, the specified mixture model remains valid, which is also referred to as an infinite mixture model [37]. For a mixture, this model basically overcomes the difficulty of finding the total number of components, and it becomes a non-parametric model as compared to a finite-mixture model, as there is no restriction on the number of data points, and these can be increased anytime naturally. The current model is parametric in nature. To make it non-parametric, the use of the Dirichlet process was introduced, and through this process, the model is now called as Dirichlet Process Mixture Model [38]. Let the set of observations be denoted by $(x_1, \ldots x_n)$ which uses $(\theta_1, \ldots, \theta_n)$ as latent parameters. For each of the latent parameters θ_i in the set, a sample is taken independently and identically from G (since G is a random distribution, the sample can be drawn from G itself, equation 3).

$$\theta_i|G \sim G \tag{3}$$

Each observation x_i has a component distribution represented as $F(\theta_i)$ that is characterized by the parameter θ_i., the generative distribution F is configured by cluster parameters θ_i and is used to generate x_i using equation 4.

$$x_i\,\big|\theta_i \sim \mathrm{F}\left(\theta^*\upsilon_i\right) \tag{4}$$

Since G is a distribution over a set of latent parameters, the new distribution can be evaluated using the conjugated prior property of the Dirichlet process using equation 5.

$$G|\eta, \mathrm{H} \sim \mathrm{DP}(\eta, \mathrm{H}) \tag{5}$$

It is possible for multiple θ_i's to have the same value in several cases since G is a discrete distribution, making the model analogous to a mixture model in which multiple x_i's with the same value of θ_i can be clustered together.

B. BAYESIAN HIERARCHICAL CLUSTERING

Each point after its arrival is assigned to a cluster using the DPMM. After each assignment, some statistical tests need to be performed in the form of a statistical comparison. The reason for making a statistical comparison is that the addition of any new point might have made some changes to the previously clustered data. To detect that change, if that happens, BHC is chosen as the statistical test. BHC is a probabilistic model used for hierarchical clustering [39]. Marginal likelihood between data points is considered. This entire examination is based on the Dirichlet Process Mixture Model, which evaluates the precise condition of the data points and obtains the exact state. The situation can be:

1) Merging of any two clusters.
2) Dividing the cluster in which the new data point is allocated using the K-Means algorithm.
3) No merging or division takes place. Since DPM is used as a generative model in the given BHC statistical test, based on two given hypotheses, the test can identify which two clusters to merge or it can decide to divide the currently allocated cluster. The two hypotheses are:
   $\mathcal{H}_1$: Data points in any two clusters were generated from the same probabilistic model.
   $\mathcal{H}_2$: The data in any cluster has two or more clusters in it.
4) For any data point x_i, $\mathrm{P}(\mathcal{H}_1|\ X_i)$ and $\mathrm{P}(\mathcal{H}_2|\ X_i)$ is calculated from equation 6.

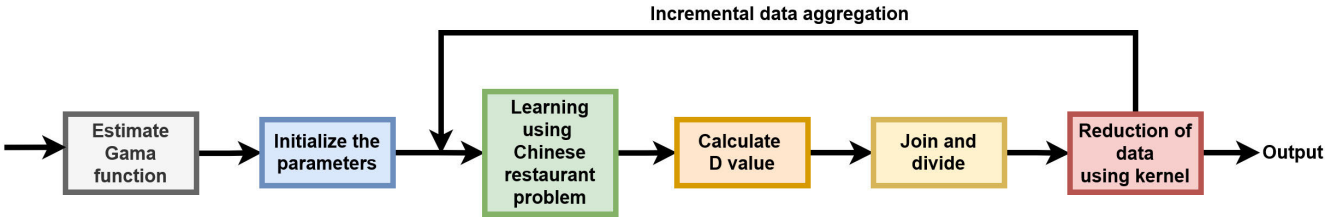


FIGURE 2. Workflow of the proposed KIHCDP approach.

If $P(\mathcal{H}_1 | X_i) \geq 0.5$, then $\mathcal{H}_1$ is more probable than $\mathcal{H}_2$.

$$p(\mathcal{H}_1|X_i) = p(\mathcal{H}_1)\, p(X_i|\mathcal{H}_1) \tag{6}$$

if $\mathcal{H}_1$, is more probable then, merge the two clusters.
If $\mathcal{H}_2$, is more probable then, divide the given cluster.

IV. PROPOSED APPROACH

The proposed kernel-based incremental hierarchical clustering using the Dirichlet process (KIHCDP) approach is twofold, where the first fold involves deciding to cluster the new data point and the second fold involves merging clusters or splitting any existing cluster. Initially, for the first input, the model has only one cluster and that point is by default assigned to that cluster. As additional points are received, the probability distribution function [40] for each one is calculated to take the initial step, which enables the proposed model to determine whether the incoming point is a part of already existing clusters or a new cluster should be created for it. The overall flowchart for the proposed approach is shown in Figure 2.

The Dirichlet process enables the model to accurately determine the number of clusters, regardless of the amount of data available, and even if the data increases steadily. Suppose, a continuous stream of data is coming, let, at some time t, a data point arrives that is represented by y_t. Until t, the model has encountered $(t-1)$ data points.

Let, the whole dataset at time t be represented by Y, where $Y = y_1, \ldots, y_t$.

The current state is the state of the model where the previous and initial numbers of clusters have already been predicted on the basis of previous data points. The model progresses to the next state when a new data point is received and its position in any of the existing clusters has not been determined. The probability of mapping y_t to a cluster is estimated with the Chinese restaurant process [41] using equation 7.

$$\Pr(y_t|Y_k) = \begin{cases} \dfrac{n_k}{n-1+\eta}, & \text{if point goes in } k^{\text{th}} \text{ cluster} \\ \dfrac{\eta}{n-1+\eta}, & \text{if point goes in new cluster} \end{cases} \tag{7}$$

The probability for the data point in the next state can be evaluated by applying equation 8:

$$\Pr(y_t|Y) = \sum_{k=1}^{K} \Pr(y_t|Y_k)\, m_k \tag{8}$$

This can be defined as the posterior probability of the new point in the next state. In equation 8, m_k is the mixing proportion for the k_{th} cluster, and Y_k is the set of data points in the k^{th} cluster. The illustration of the process of producing data incrementally from IoT devices and the proposed clustering technique is depicted in Figure 3. The new point can appear in any of the current clusters or in any possible partition. Each data point contribution must be considered, and therefore, the individual data point is considered while evaluating equation 1 by normalizing its weight over all clusters in the data set. The next step of the model is to determine the exact location of the new data point concerning the existing clusters in the following manner:

i. The position of the new data point can be in any of the previously defined clusters or
ii. The model assumes that the new data point has yet to be encountered by the model; hence it does not belong to any of the currently available clusters. As a result, a new cluster is created for the new data point.

Using the Bayesian clustering model, the probability of a data point being in the next state is calculated using the predictive distribution as given in equation 9.

$$\Pr(y_t k^*|Y) = \sum_{k=1}^{K} \Pr(y_t|Y_k)\Pr(Y_k) \tag{9}$$

The cluster (k *) is the one that has the highest probability of receiving the data point y_t when considering the set of data points Y_k and estimated using the equation 10.

$$k^* = \underbrace{\operatorname{argmax}}_{k} \Pr(y_v, c_t = t|Y_k)\Pr(Y_k) \tag{10}$$

where the marginal probability $\Pr(y_t, c_t = t|Y_k)\Pr(Y_k)$ is defined as $\int \Pr(y_t|\Theta_k)\Pr(\Theta_k|Y_k)\, d\Theta_k$, and a portion of this integral is traceable using the property of conjugate priors. Now, the model evaluates the possible probability of a data point y_t using predictive distribution as shown in equation 11:

$$\Pr(y_t t+1|Y) = \sum_{k=1}^{K} \Pr(Y_{ct}|Y_k)\Pr(Y_k) + \Pr(Y_r|Y_{k+1})\Pr(Y_{k+1}) \tag{11}$$

where, $Y - t$ is the set of datasets except Y_t.

After evaluating both possible scenarios for a data point, the next step of the model is to decide whether the data point belongs to the current cluster or needs to be classified into a

Algorithm 1 Kernel-Based Incremental Clustering

```
K = 1                                          ▷ Number of clusters
C = {}                                         ▷ Index for clusters
Y = {}                                         ▷ Dataset
Y_t                                            ▷ Arrival of data at time t
while Y_t do
    Update prior parameters for Pr(Y)
    ADD Y_i to Y
    Evaluate D
    Compute k*
    if D < 1 then
        Assign cluster k* to point Y_i
        Evaluate X_Divide(Y_k*)
        Initialize k_m = 2
        Apply K-Means on Cluster k* with k_m
        if X_Divide(Y_k*) > 1 then
            for k = 1toK do
                Evaluate X_Join(Y_k_i*, Y_k)
                Evaluate X_Join(Y_k_j*, Y_k)
            end for
            if min(X_Join(Y_k, Y_k_i*)) > 1 then
                Merge Y_k and Y_k_i*
            else
                k_i* ← k
            end if
            if min(X_Join(Y_k, Y_k_j*)) > 1 then
                Merge Y_k and Y_k_j*
            else
                k_i* ← k + 1
                K ← K + 1
            end if
        end if
    else
        Compute X_Join ∀ k_i × k_j
        if min(X_Join(Y(k_i, k_j)) > 1 then
            Merge k_i and k_j and drop k_j
            K ← K − 1
        end if
    end if
    for ∀x, y ∈ k* do
        sim=similarity(x,y)
        if sim > ξ then
            drop x from k*
        end if
    end for
end while
```

new cluster. Algorithm 1 represents the pseudocode for the proposed approach.

A. DECISION STEP

This is the step in which the model finally decides the exact position of the data point, that is, whether it belongs to any of the current clusters or should be associated with a new cluster. To decide the current state of a data point, the model uses equations 2 and 4. The model allocates the new data point to the current cluster if the probability of that data point in the current cluster (2) is greater than that of the new cluster (4), otherwise, the data point is assigned to the new cluster. Hence, this parameter is denoted by D and evaluated using equation 12:

$$\mathrm{D} = \frac{\Pr\left(\mathrm{y_t t^*|Y}\right)}{\Pr\left(\mathrm{y_t, t+1|Y}\right)} \tag{12}$$

Calculating these values in the current form is computationally hard, so the model uses marginal likelihood on the DPMM. Hence, D can be rewritten using the equation 13:

$$\mathrm{D} = \frac{\eta \Pr\left(\mathrm{y_t, t+1|Y}\right) \prod_{\mathrm{k_j}=1}^{\mathrm{K}+1} \varphi\left(\mathrm{N_{k_j}}\right)}{\Pr\left(\mathrm{y_t t^*|Y}\right) \prod_{\mathrm{k_i}=1}^{\mathrm{K}} \varphi\left(\mathrm{N_{k_i}}\right)} = \frac{\eta \Pr(\mathrm{y_t}|\theta_{\mathrm{k}+1})}{\Pr(\mathrm{y}_t\,|\theta_k*)\,\mathrm{N(k^*)}} \tag{13}$$

Whenever y_t, a new datapoint arrives, the given function in equation 13 is evaluated. In case $D > 1$, a new cluster is created and the point under consideration is assigned to the newly formed cluster otherwise, the point is assigned to k^* cluster.

Now that the data point has been classified, the model examines whether there is any notable change that has happened to the current data due to the addition of a new point. Thus, the next step checks for the probable change in the behavior of data.

B. JOIN AND DIVIDE STEP

The next step after deciding the exact position of the new data point is to check whether the state of the current state of the model has changed due to the addition of a new data point. Therefore, this step enables the model to look for any such possible change. Suppose, the current state of the model has k clusters along with some given dataset. The model looks for changes by considering the scenario to either join two current clusters or divide any current cluster. The divide step uses the approach used in the original BHC, each and every current cluster is considered for the divide step, denoted by (C_k). K-Means algorithm is applied with simple bisecting to obtain two clusters. Let these clusters be denoted by C_{ki} and C_{kj}. The marginal probability $\Pr(Y_k)$ of the data in the cluster C_k is given by equation 14.

$$\Pr\left(\mathrm{Y_k}\right) = \Pr\left(\mathrm{Y_k}|\beta_{\mathrm{k}}\right) \Pr\left(\beta_{\mathrm{k}}\right) \tag{14}$$

Similarly, both the clusters C_{ki} and C_{kj} are evaluated, and for both clusters, the marginal probability is given by equation 15.

$$\Pr\left(Y_{\mathrm{k_i}} \mathrm{U} Y_{\mathrm{k_j}}\right) = \Pr\left(\mathrm{Y_{k_i}}|\beta_{\mathrm{k_i}}\right) \Pr\left(\beta_{\mathrm{k_i}}\right) \Pr\left(\mathrm{Y_j}|\beta_{\mathrm{k_j}}\right) \Pr\left(\beta_{\mathrm{k_j}}\right) \tag{15}$$

After evaluating both $\Pr(Y_k)$ and $\Pr(Y_{ki}\ \mathrm{U}\ Y_{kj})$, the divide parameter (X_{Divide}) can be evaluated using equation 16.

$$X_{\mathrm{Divide}} = \frac{\Pr\left(Y_{\mathrm{k_i}}|\beta_{\mathrm{k_i}}\right) \Pr\left(\mathrm{Y_{k_j}}|\beta_{\mathrm{k_j}}\right) \Pr(\beta_{\mathrm{k_i}}) \Pr(\beta_{\mathrm{k_j}})}{\Pr\left(\mathrm{Y_k}|\beta_{\mathrm{k}}\right) \Pr\left(\beta_{\mathrm{k}}\right) \Pr(\beta_{\mathrm{k}}} \tag{16}$$

Equation 16 can also be viewed as dividing the k^{th} cluster if the numerator is greater than the denominator. This also specifies that it is better to divide the cluster as the addition of the new data point has led to a change in the behavior of

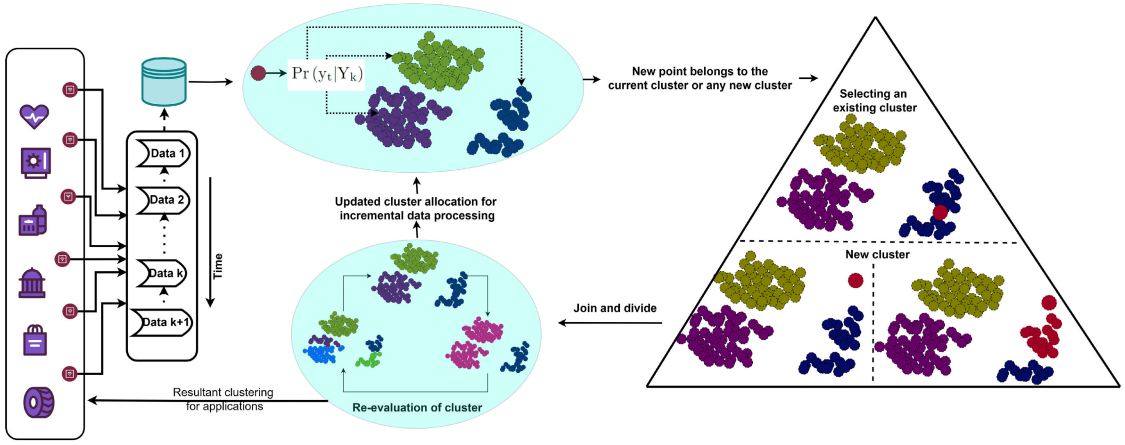


FIGURE 3. A graphical representation of the proposed approach.

the existing clusters. The given equation is similar to what has been done in the original BHC [11]. The comparison is based on the amount of data in the clusters rather than one point, so the model takes into account the number of data points in a cluster. This step allows the model to evaluate how much a cluster contributes to the entire data.

$$\Pr(\beta_k) = \frac{N_k}{N} \tag{17}$$

This step allows the model to use the Kernel for reducing data, where the model removes data points from the dataset however, the count of the number of points in each cluster is not altered at the time when the model is reducing data. This allows the model to retain information about the reduced points that can be used to calculate the division parameter even after the point is removed from the data pool. With the participation of the number of points, the above equation can be written as equation 18.

$$X_{\text{Divide}} = \frac{\Pr(Y_{k_i}|\beta_{k_i})\Pr(Y_{k_j}|\beta_{k_j})\,N_{k_i}N_{k_j}}{\Pr(Y_k|\beta_k)\,N_k * N} \tag{18}$$

where $\Pr(Y_k \mid \beta)$ is calculated with a conjugate prior.

If the value of $X_{Divide}(Y_k)$ is less than 1, the cluster C_k is kept with the addition of the new data point. On the other hand, if the value of $X_{Divide}(Y_k)$ is greater than 1, the cluster C_k is divided into two clusters, C_{k_i} and C_{k_j}, and cluster count k is increased by 1.

The model also considers the other case where two clusters can represent the same information, hence, these two clusters need to be joined. Therefore, whenever processing a new data point, each possible combination of two clusters is considered. The model evaluates whether any two clusters represent the same information. If they do, then it joins them together. Let, C_{ki} and C_{kj} be two cluster considered for merging. After merging, let those be denoted by C_k.

The marginal probability for the merged cluster C_k is given by equation 19.

$$\Pr(Y_k) = \Pr(Y_k|\beta_k)\Pr(\beta_k) \tag{19}$$

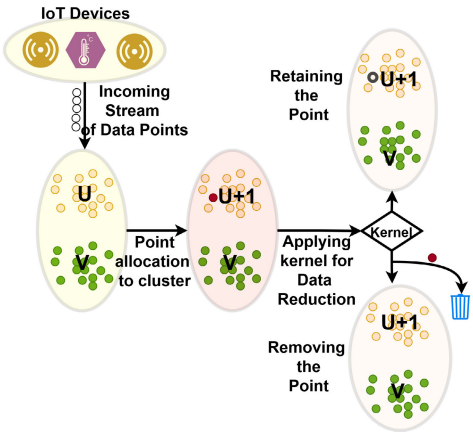


FIGURE 4. Incremental data reduction process using kernel.

Similarly for two considered clusters C_{ki} and C_{kj}, it is given by equation 20.

$$\Pr(Y_{k_i} \cup Y_{k_j}) = \Pr(Y_{k_i}|\beta_{k_i})\Pr(\beta_{k_i})\Pr(Y_{k_j}|\beta_{k_j})\Pr(\beta_{k_j}) \tag{20}$$

Hence, the join parameter ($X_{Join}(Y_{k_i}, Y_{k_j})$), can be evaluated using equation 21.

$$X_{\text{Join}}\left(Y_{k_i'} Y_{k_j}\right) = \frac{\Pr(Y_k|\beta_k)\Pr(\beta_k)}{\Pr(Y_{k_i}|\beta_{k_i})\Pr(\beta_{k_i})\Pr(Y_{k_j}|\beta_{k_j})\Pr(\beta_{k_j})} \tag{21}$$

With the participation of the number of data points, the above equation can be written as equation 22.

$$X_{\text{Join}}\left(Y_{k_i'} Y_{k_j}\right) = \frac{\Pr(Y_k|\beta_k)\,N_k * N}{\Pr(Y_{k_i}|\beta_{k_i})\Pr(Y_{k_j}|\beta_{k_j})\,N_{k_i} * N_{k_j}} \tag{22}$$

If $X_{Join}(Y_{k_i}, Y_{k_j})$ is larger than 1, we determine to merge clusters k_i and k_j into k. The subsequent step is to decrease the data by utilizing a kernel.

C. INCREMENTAL KERNEL-BASED DATA REDUCTION

A kernel-based approach is proposed to reduce the need for intensive computation. This approach is based on the

concept of mapping non-separable data in lower dimensions to separable data in higher dimensions with the aid of a similarity function. The similarity function is used to identify the optimal position in the higher dimension. The purpose of this approach is to reduce the amount of data by recognizing two points that are similar enough to represent the same information, and thus one of the points can be removed. To do this, the similarity between two points is calculated using a Gaussian kernel-based function as represented in equation 23.

$$k(x, y) = e^{\frac{-||x-y||^2}{2\sigma_c^2}} \tag{23}$$

Whenever a point is classified into a cluster, then, sigma is evaluated using all the points of that specific cluster. x being the new point, is compared to all other points in the group in which it is added (represented by y). If $k(x, y) > \xi$, then the new point is not added to the group as shown in Figure 4. This approach is used to reduce the data, as two points, if they are sufficiently similar, represent the same information, and hence the new data point can be removed. The next question is, will we lose information? Surely, we will. However, it will hardly affect the model as we are taking cluster participation, where the count is not lost, only the point is removed.

D. COMPLEXITY ANALYSIS

Considering the n number of d dimension points and k number of clusters, the computational complexity for k-means algorithm is $O(n*k*d)$. This complexity scales up by the iteration factor for finding the number of clusters. The complexity of GMM is $O(n*k*d2)$ which also increases with the number of iterations. The complexity of the adaptive clustering is $O(c*k)$ where c is the complexity for fuzzy C-Means clustering. The proposed approach takes $O(n*k)$ for the allocation of the n points to the initial cluster. The allocation of the new data point to one of the clusters is $O(n*k)$. The decision step using the marginal likelihood takes $O(n*k)$. The calculation of the join-and-divide decision takes $O(2n)$. The complexity of the data reduction step is $O(n*k)$. The proposed approach also takes advantage of the data reduction approach, which reduces n as the new data gets added.

V. EVALUATION

We intend to determine the effectiveness of the proposed approach by conducting benchmark experiments. The purpose of this analysis is to answer the following research questions related to the incremental addition of data into the right cluster.

- How is the data distribution changing as the new instances of the data get recorded by the IoT devices?
- How does the proposed model perform as compared to the existing models in terms of accuracy while data clustering?
- What is the effect and relation of incremental addition of data with the varying number of clusters?
- How does the proposed approach contribute to reducing the amount of data as the new data points are added?
- How much gain does the proposed approach acquire concerning the existing approaches in the context of the computation time?
- How much is the proposed model capable of maintaining the trade-off between the running time and the accuracy?

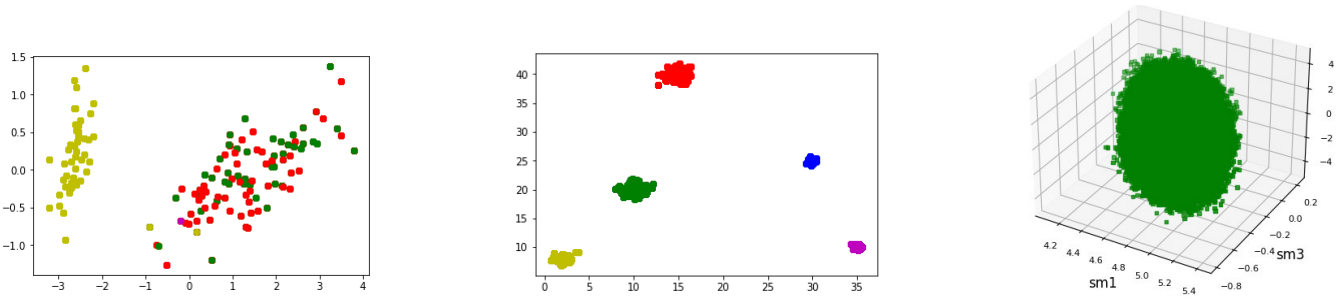
TABLE 2. Datasets used for the experimental analysis.

S.No.	Dataset	Features	Instances
1.	KDD99 [44]	23	384519
2.	Iris [45]	5	150
3.	Synthetic Data	4	100000
4.	Water Monitoring [46]	6	46893
5.	AWSIoT [47]	7	6319485
6.	Wind turbine [48]	9	388151

Dataset Description and Experimental Setup: The proposed approach is evaluated on several benchmark datasets compared to the existing approaches. The datasets used for this analysis are shown in Table 2. The Iris dataset includes fifty samples from three different species of Iris. The dataset is not linearly separable, therefore, a good candidate for testing the proposed approach. The synthetic dataset consists of 100,000 observations of a four-dimensional random vector taken from a combination of six multivariate Gaussian distributions. The water monitoring [46] data set contains a stream of five sensors that continuously monitor the level of pollution in the river stream. A sample visualization of three datasets is shown in Figure 5. KDD99 [44] dataset contains a network traffic stream dataset used for network intrusion detection. The wind turbine [48] dataset is generated by the sensors mounted on the wind turbines to monitor their health. The AWSIoT dataset [47] is generated by sensors that monitor smart devices connected to Amazon web services. The experiments are conducted using Python 3.6 supported by a system equipped with a RAM of 16GB and Intel Corei7 12th generation processor. The comparison is made with the widely used K-Means clustering approach. The proposed approach is also compared with a recent adaptive clustering approach [42]. The result shows a significant improvement in the clustering performance. Moreover, the suggested technique is evaluated to determine if there is a decrease in the amount of data points when new data is added to the existing collection of datasets. To analyze this aspect of the proposed approach, the accuracy and number of points available in the pool are analyzed.

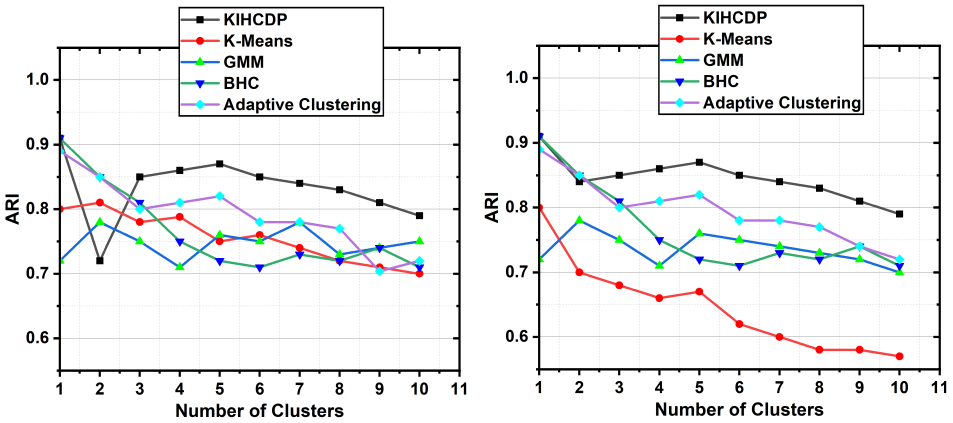
Comparables: The proposed KIHCDP model utilizes Adjusted Rand Index (ARI) [43] which is a corrected form of Rand Index ($\mathbf{R}$) [43] as an accuracy parameter. The Rand index is used to find similarity between two clusters taking into account all pairs of samples. It counts the number of pairs that are classified into the same or different clusters in the resulting clusters and actual clusters. The $\mathbf{R}$ score produced is then normalized for randomness in the Adjusted Rand Index (ARI) using equation 24.

$$ARI = \frac{\mathbf{R} - Expexted(\mathbf{R})}{max(\mathbf{R}) - Expected(\mathbf{R})} \tag{24}$$



(a) IRIS Dataset (b) Sythetic Dataset (c) AWSIoT Dataset

FIGURE 5. A sample visualization of the datasets.



(a) Water monitoring dataset (b) AWSIoT dataset

FIGURE 6. A comparative analysis of the existing approach for different datasets.

TABLE 3. ARI analysis of the proposed approach over different datasets.

S.No.	Dataset	KIHCDP	K-Means	GMM	Adaptive Clustering	BHC
1.	KDD99 [44]	0.86	0.78	0.8	0.84	0.82
2.	Iris [45]	0.9	0.86	0.88	0.89	0.88
3.	Synthetic Data	0.92	0.82	0.87	0.9	0.88
4.	Water Monitoring [46]	0.87	0.74	0.79	0.83	0.81
5.	AWSIoT [47]	0.91	0.83	0.88	0.9	0.89
6.	Wind turbine [48]	0.87	0.77	0.81	0.85	0.83

A. CLUSTERING ACCURACY ANALYSIS

In this section, we present an analysis of the proposed approach compared to the state-of-the-art K-Means [6] clustering approach, GMM [16], BHC [9] and adaptive clustering [42] approaches are also compared with the proposed approach. This analysis is conducted using AWSIoT and water monitoring datasets. The AWSIoT dataset has a concentrated distribution of data and the water monitoring dataset has a very scattered distribution. The data's arrangement alters as new points are included in the existing set of data. The testing over these two datasets is done to confirm the robustness of the proposed approach over a range of datasets. It has been observed that there is an average gain of 4% over the recent adaptive clustering approach for water monitoring dataset. Figure 6a shows a comparative analysis of the proposed approach with existing approaches for the water monitoring dataset. Figure 6b shows the performance of the proposed approach over the AWSIoT dataset. The suggested technique shows an average increase of 3% compared to the recent adaptive clustering approach with a varying number of clusters. The decrease in ARI is the result of the overfitting that occurs in the clustering process. This result can be interpreted in the sense that the formed clusters are becoming unclear and there is no clear separation among the clusters. There is very little distinction among the points belonging to the different clusters. As the number of clusters increases, the performance decreases because the ideal number of clusters for the water monitoring dataset is 5. At this value, the performance of the proposed approach is maximum. The accuracy analysis for different datasets is presented in Table 3. The proposed approach yields an increase of 8%, 6%, 2%, and 4% compared to the K-Means, GMM, adaptive clustering, and BHC approach for the KDD99 dataset respectively. For Iris dataset, a gain of 4%, 2%, 1%, and 2% is observed, while for synthetic dataset, a gain of 10%, 5%, 2%, and 4% is achieved. The

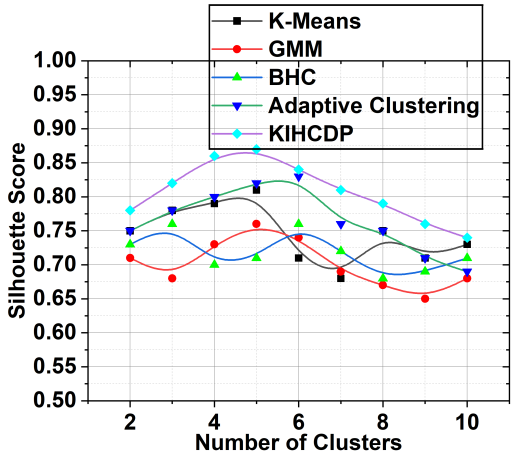


FIGURE 7. A comparative analysis of the proposed approach with the existing approaches using the water monitoring dataset.

water monitoring dataset, which is highly drifting, shows a high accuracy gain, with a gain of 13%, 8%, 4%, and 6% compared to the K-Means, GMM, adaptive clustering, and BHC approach respectively. The AWSIot and wind turbine datasets also demonstrate a gain of 8%, 3%, 1%, and 2% and 10%, 6%, 2%, and 4%, respectively, compared to the K-Means, GMM, adaptive clustering, and BHC approach.

Further analysis has been conducted to verify the accuracy of the proposed approach as compared to the existing approaches. This analysis as shown in Figure 7 has been conducted using the water monitoring dataset. The proposed approach has shown an average gain of 6% over the K-Means clustering, 9% over the GMM, 8% over the Bayesian hierarchal clustering, and 4% as compared to the recent adaptive clustering approach.

B. RUNNING TIME ANALYSIS

To analyze the running time, the proposed approach is tested on different datasets mentioned. Running-time analysis includes the time required to perform clustering and decide on merging or splitting the clusters. Figure 8 shows a logarithmic time analysis of the proposed approach on different datasets and also compares it with the existing approaches. The result shown in the figure clearly indicates the gain of the proposed approach over the existing approaches. KIHCDP takes advantage of incremental cluster updates and considers only the important data points, hence reducing the running time.

For the Iris dataset, KIHCDP demonstrated a performance increase of 8%, 11%, 5%, and 2% compared to BHC, K-Means, GMM, and adaptive clustering, respectively. With the AWSIoT dataset, KIHCDP shows an improvement of 13%, 22%, 19%, and 17% over BHC, K-Means, GMM, and adaptive clustering approaches respectively. This high gain is attributed to the concentrated distribution of the dataset. A similar high gain was observed for the synthetic and KDD99 datasets, with KIHCDP achieving gains of 22%, 24%, 21%, and 13% and 24%, 10%, 15%, and 10% over the BHC, K-Means, GMM, and adaptive clustering approaches, respectively. For the water monitoring dataset, KIHCDP achieves a gain of 14%, 16%, 13%, and 10% compared to BHC, K-Means, GMM and adaptive clustering. Lastly, for wind turbine dataset, KIHCDP shows an improvement of 12%, 13%, 11%, and 10% compared to BHC, K-Means, GMM, and adaptive clustering approaches.

C. INCREMENTAL ANALYSIS

In order to examine the change in the distribution of data, a visualization of the data is presented in Figure 9. The water monitoring dataset is used for this analysis, as the data is highly drifting due to the changes in the levels of pollution in river water. Figure 9 (a) shows the first batch of sensor readings. The subsequent figures from b to f show the evolving data distribution. The observed change is further quantified using the change in entropy as more number of instances are added. Furthermore, for analyzing the performance of the proposed approach with the continuous addition of data, an incremental analysis has been conducted. Data are continuously added to the input dataset pool to analyze changes in the formed clusters. The complete dataset is divided into batches, and each time a batch is added to the input pool, the performance is evaluated. The variable factors in this analysis are a number of clusters and the batch of data. With respect to these two parameters, the performance is evaluated. Figure 10 shows the performance of the proposed approach. It is evident that the accuracy of the clusters created in the context of ARI increases as the number of instances grows. It shows the trade-off between the clustering accuracy of the proposed approach with respect to the streaming data. The final clusters are used as a benchmark for the compression of the intermediate clusters. There is a sharp increase of almost 8% while the number of clusters changes from 2 to 4 for data instances ranging from 10% to 100%. The approach maintains a consistent slope in the context of the number of instances and accuracy, as shown in Figure 10. However, it has been observed that the slope is not consistent in the context of an increasing number of clusters with respect to accuracy. The same can be observed when the number of clusters increases from 4 to 6 and from 9 to 10. The proposed approach tries to split or merge the existing clusters, which results in this kind of deflection.

To justify this deflection of accuracy an analysis of change in entropy has been conducted. Figure 11 shows this analysis for the water monitoring and AWSIoT datasets. It has been observed that there is a similar deflection of entropy when the data are streaming to the processing system.

D. DATA REDUCTION AND RUNTIME ANALYSIS

Our approach reduced the amount of data as new information is gradually added. Initially, most of the data points remain different, resulting in less reduction, as shown in Figure 12 and 13. An average reduction of 5% is observed in the first batch of data. Subsequently, in the next three batches,

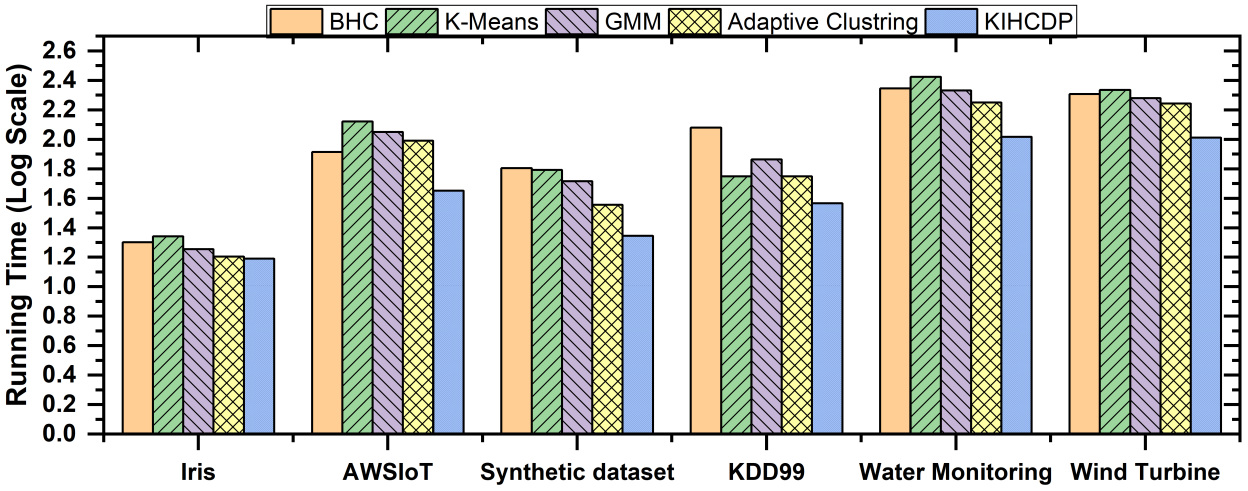
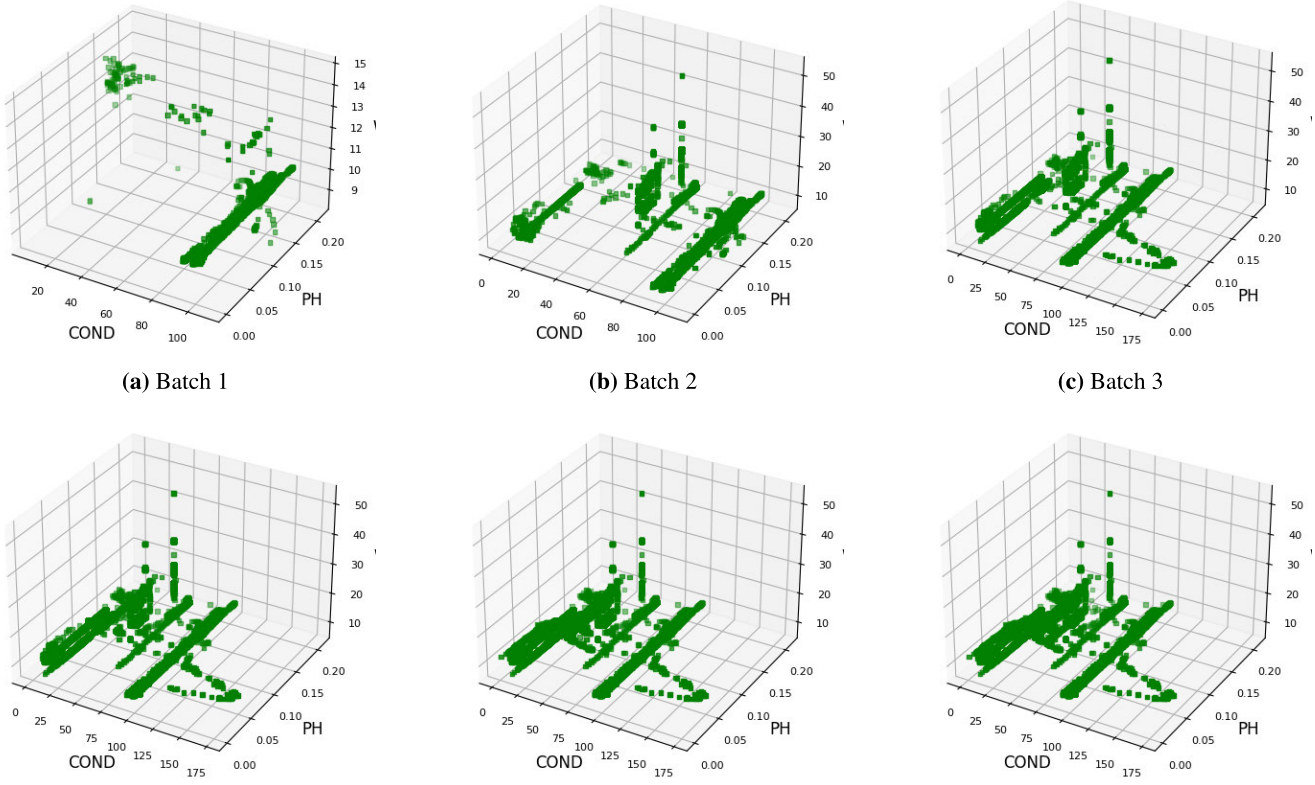


FIGURE 8. Running time analysis of the proposed approach as compared to existing approaches while using different datasets.



(a) Batch 1 **(b)** Batch 2 **(c)** Batch 3

(d) Batch 4 **(e)** Batch 5 **(f)** Batch 6

FIGURE 9. Incremental data streaming resulting in change of data distribution.

an average reduction of 12% is observed with a gain of approximately 6% in each batch. Up to the first five batches, an average reduction of 16% is observed. In the subsequent addition of data batches after the fifth batch, there is only an improvement of at most 3% reduction in data. This reduction is due to the presence of similar points in the incoming stream of data. This reduction results in a shorter processing time for the available dataset.

A trade-off analysis of the water monitoring dataset is presented in Figure 14 to assess the alteration in clustering accuracy and running time with respect to the similarity threshold. As the similarity threshold rises, the number of data points removed from the data pool decreases, and the running time increases. Conversely, when the similarity threshold is lowered, the number of data points eliminated from the data pool increases, and the running

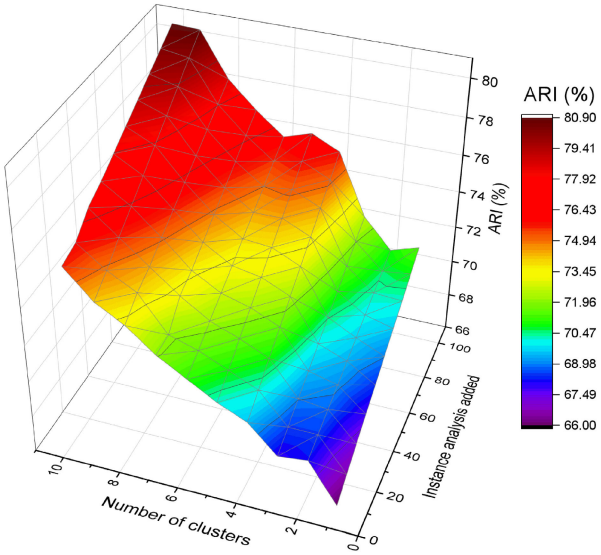


FIGURE 10. A trade-off analysis for accuracy with varying number clusters and number of data instances.

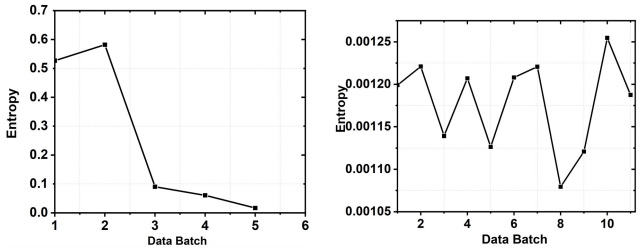


FIGURE 11. Entropy analysis for (a) Water monitoring and (b) AWSIoT datasets.

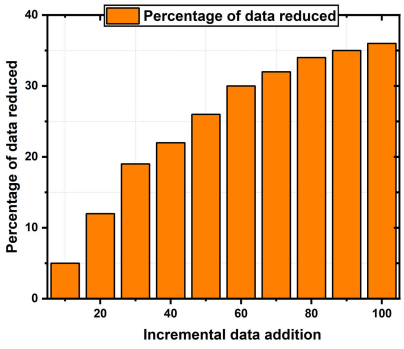


FIGURE 12. Analysis of data reduction while considering incremental addition of data for AWSIoT dataset.

time decreases. It has been observed that the running time is initially low when the similarity threshold is low. As the similarity threshold is increased, more points are needed for computation, resulting in a longer running time. The ARI (Adjusted Rand Index) ranges from 0.5 to 0.68 with this change. A performance improvement is seen in terms of ARI when the similarity threshold is increased. However, there is no significant change in the ARI after the threshold is raised to 0.8, as seen in Figure 14. At this point, the running time increases sharply. This analysis suggests that a balance between running time and ARI can be achieved by adjusting

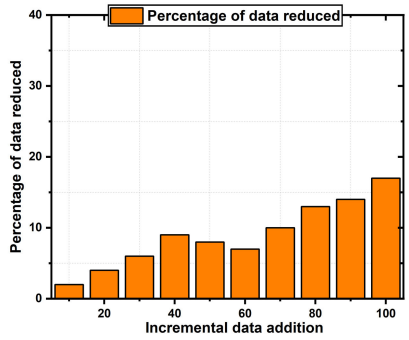


FIGURE 13. Analysis of data reduction while considering incremental addition of data for Water Monitoring dataset.

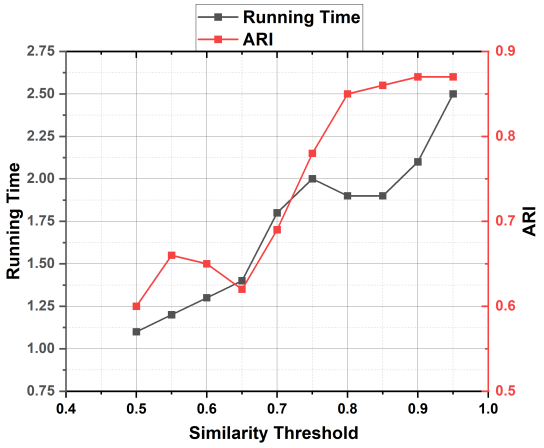


FIGURE 14. A trade-off analysis of the running time and ARI with changing value of the similarity threshold ξ.

the similarity threshold, such as 0.8 for the water monitoring dataset.

VI. DISCUSSION

The proposed approach, KIHCDP, can be used to perform data clustering on a continuous stream of data. KIHCDP employs a Dirichlet process model at its center and handles the evolution of the clusters in the streaming data. KIHCDP decides based on the Dirichlet process whether to divide and join existing clusters. As additional information is included in the collection, the general arrangement of the data alters, and the suggested technique deals with this alteration effectively. The proposed approach uses a probabilistic decision-making process to decide whether to retain a point in the data pool or just to keep the information about the point. Using the available benchmark datasets, the KIHCDP is compared with the K-Means clustering, GMM, BHC, and adaptive clustering approaches to assess its performance. Based on the performed experiments and analyses, the following perceptions can be made:

- The experimental study has revealed that the suggested technique is capable of dealing with the gradual incorporation of data while carrying out data clustering. The proposed approach opts to decrease the data in accordance with the data distribution. There is less gain in performance for datasets where data points are

concentrated in a specific region. However, the proposed approach has shown better performance in a scenario where the data distribution is changing.

- The proposed model can be utilized in analogous situations where similar difficulties are present. To back up our assertion, we have provided an examination of datasets from various fields, for example, the wind turbine monitoring dataset.
- As the amount of data keeps increasing, the proposed approach can take care of allied data readings just by keeping the information about the data. However, in cases where the data distribution is highly drifting, the proposed approach retains the data, which can lead to high running time. We have presented this trade-off analysis and the decision about speed-up and accuracy should be taken based on the specific application domain.
- The proposed approach is scalable for a large amount of IoT data. The proposed approach can be generalized to similar applications where clustering over the continuous stream is required. The proposed approach also has real-time adaptability, which is very essential for applications like anomaly detection with the help of smart monitoring.

VII. CONCLUSION AND FUTURE REMARKS

This data clustering approach for IoT data is designed to tackle the issue of streaming data and can be applied to other applications with related problems. After conducting an extensive experimental analysis with datasets from various domains, the proposed approach has demonstrated consistent performance. The performance has been evaluated using parameters such as running time, clustering accuracy (considering ARI), and data reduction. The results suggest that it is possible to reduce the data points from streaming data to minimize the processing complexity. It can be further concluded that it is not necessary to retrain all the data points; a balance can be achieved between training time and accuracy. The experiments conducted show that the proposed KIHCDP model has improved performance of 5-10% as compared to the existing standard approaches and recently published approaches. The presented approach is not only limited to one particular situation but can be adapted to equivalent applications with few modifications and by adjusting some parameters. With the rapid expansion of the Industrial Internet of Things (IIoT), more and more IoT devices are being used to collect data and make decisions. The proposed approach seeks to take advantage of these devices to make better decisions. It is necessary to investigate further to assess how the proposed or related methods can be applied to a small business and to convert it into a smart factory on a small scale. Furthermore, the application of the proposed approach can be explored for the image segmentation application for finding the different segments in an image. The data elimination strategy of the proposed approach can be useful for devices with low computational power and an analysis can be done to understand the effect on latency between the cloud storage and edge side while considering a tradeoff between resource management and overall performance.

ACKNOWLEDGMENT

The authors express sincere thanks to Vellore Institute of Technology, Chennai, India, for providing necessary research support for this work. They also would like to extend their gratitude to Indian Institute of Information Technology Allahabad, India, for providing the water monitoring dataset to test the effectiveness of the proposed approach.

REFERENCES

[1] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2020.

[2] K. L. Tsui, V. Chen, W. Jiang, F. Yang, and C. Kan, "Data mining methods and applications," in *Springer Handbook of Engineering Statistics*. London, U.K.: Springer, 2023, pp. 797–816.

[3] D. Puschmann, P. Barnaghi, and R. Tafazolli, "Adaptive clustering for dynamic IoT data streams," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 64–74, Feb. 2017.

[4] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[5] M. Hosseinzadeh, A. Hemmati, and A. M. Rahmani, "Clustering for smart cities in the Internet of Things: A review," *Cluster Comput.*, vol. 25, no. 6, pp. 4097–4127, Dec. 2022.

[6] H. Hu, J. Liu, X. Zhang, and M. Fang, "An effective and adaptable K-means algorithm for big data cluster analysis," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109404.

[7] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, Jun. 1996.

[8] M. Arora, S. Agrawal, and R. Patel, "User location prediction using hybrid BIRCH clustering and machine learning approach," *J. Integr. Sci. Technol.*, vol. 12, no. 1, p. 701, 2024.

[9] K. A. Heller and Z. Ghahramani, "Bayesian hierarchical clustering," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 297–304.

[10] Y. Liu and B. Li, "Bayesian hierarchical K-means clustering," *Intell. Data Anal.*, vol. 24, no. 5, pp. 977–992, 2020.

[11] H. Lee, K. Kwak, and S. Jo, "An incremental nonparametric Bayesian clustering-based traversable region detection method," *Auton. Robots* 41, pp. 795–810, Apr. 2017.

[12] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231).

[13] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 1–21, 2017.

[14] Y. Li, E. Schofield, and M. Gönen, "A tutorial on Dirichlet process mixture modeling," *J. Math. Psychol.*, vol. 91, pp. 128–144, Aug. 2019.

[15] R. Mussabayev, N. Mladenovic, B. Jarboui, and R. Mussabayev, "How to use K-means for big data clustering?" *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109269.

[16] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia Biometrics*, vol. 741, pp. 659–663, Jul. 2009.

[17] D. J. Aldous, I. A. Ibragimov, J. Jacod, and D. J. Aldous, *Exchangeability and Related Topics*. Berlin, Germany: Springer, 1985, pp. 1–198.

[18] S. Feng, "Hierarchical Dirichlet process and relative entropy," *Electron. Commun. Probab.*, vol. 28, pp. 1–12, Jan. 2023.

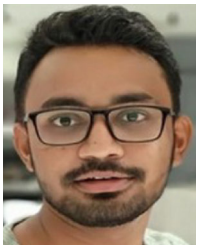
[19] J. Yao, X. Cao, Q. Zhao, D. Meng, and Z. Xu, "Robust subspace clustering via penalized mixture of Gaussians," *Neurocomputing*, vol. 278, pp. 4–11, Feb. 2018.

[20] J. Yao, B. Zhang, C. Li, D. Hong, and J. Chanussot, "Extended vision transformer (ExViT) for land use and land cover classification: A multimodal deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5514415.

[21] A. Zubaroglu and V. Atalay, "Data stream clustering: A review," *Artif. Intell. Rev.*, vol. 54, no. 2, pp. 1201–1236, 2021.
[22] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, "Data mining for Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 77–97, 1st Quart., 2014.
[23] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery*, Jun. 2003, pp. 2–11.
[24] J. Diaz-Rozo, C. Bielza, and P. Larrañaga, "Clustering of data streams with dynamic Gaussian mixture models: An IoT application in industrial processes," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3533–3547, Oct. 2018.
[25] M. K. Islam, M. M. Ahmed, and K. Z. Zamli, "A buffer-based online clustering for evolving data stream," *Inf. Sci.*, vol. 489, pp. 113–135, Jul. 2019.
[26] J. Kumar, J. Shao, R. Kumar, S. U. Din, C. B. Mawuli, and Q. Yang, "A context-enhanced Dirichlet model for online clustering in short text streams," *Exp. Syst. Appl.*, vol. 228, Oct. 2023, Art. no. 120262.
[27] W. Jing, M. Papathomas, and S. Liverani, "Variance matrix priors for Dirichlet process mixture models with Gaussian kernels," 2022, *arXiv:2202.03946*.
[28] Z. Bing, Y. Meng, Y. Yun, H. Su, X. Su, K. Huang, and A. Knoll, "DIVA: A Dirichlet process mixtures based incremental deep clustering algorithm via variational auto-encoder," 2023, *arXiv:2305.14067*.
[29] Z. Li, H. Yan, C. Zhang, A. Wang, W. Ketter, L. Sun, and F. Tsung, "Tensor Dirichlet process multinomial mixture model for passenger trajectory clustering," 2023, *arXiv:2306.13794*.
[30] V. Kaltsa, A. Briassouli, I. Kompatsiaris, and M. G. Strintzis, "Multiple hierarchical Dirichlet processes for anomaly detection in traffic," *Comput. Vis. Image Understand.*, vol. 169, pp. 28–39, Apr. 2018.
[31] M. Bilancia, M. Di Nanni, F. Manca, and G. Pio, "Variational Bayes estimation of hierarchical Dirichlet-multinomial mixtures for text clustering," *Comput. Statist.*, vol. 38, pp. 2015–2051, Dec. 2023.
[32] C. Li, "Stochastic processes," in *Handbook of Medical Statistics*. 2001, pp. 241–267.
[33] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, *Bayesian Nonparametrics*, vol. 28. Cambridge, U.K.: Cambridge Univ. Press, 2010.
[34] P. Diaconis and D. Ylvisaker, "Conjugate priors for exponential families," *Ann. Statist.*, vol. 7, no. 2, pp. 269–281, 1979.
[35] Y. W. Teh, "Dirichlet process," *Encyclopedia Mach. Learn.*, vol. 1063, pp. 280–287, Jan. 2010.
[36] M. West, "Hyperparameter estimation in Dirichlet process mixture models," Duke Univ., Durham, NC, USA, ISDS Discuss. Paper 92-A03, 1992.
[37] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 554–560.
[38] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *J. Comput. Graphical Statist.*, vol. 9, no. 2, pp. 249–265, 2000.
[39] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967.
[40] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, Sep. 1962.
[41] Y. Teh, M. Jordan, M. Beal, and D. Blei, "Sharing clusters among related groups: Hierarchical Dirichlet processes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1385–1392.
[42] A. Raut, A. Shivhare, V. K. Chaurasiya, and M. Kumar, "AEDS-IoT: Adaptive clustering-based event detection scheme for IoT data streams," *Internet of Things*, vol. 22, Jul. 2023, Art. no. 100704.
[43] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.
[44] KDD. *0000. KDD-CUP-99 Task Description—UCI KDD Archive*. Accessed: Apr. 17, 2024. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/task.html
[45] J. P. Pinto, S. Kelur, and J. Shetty, "Iris flower species identification using machine learning approach," in *Proc. 4th Int. Conf. Converg. Technol. (ICT)*, Oct. 2018, pp. 1–4.
[46] M. Kumar, T. Singh, M. K. Maurya, A. Shivhare, A. Raut, and P. K. Singh, "Quality assessment and monitoring of river water using IoT infrastructure," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10280–10290, Jun. 2023.
[47] Aws-Samples. (2016). *AWS-Samples/AWS-IoT-Examples*. [Online]. Available: https://github.com/aws-samples/aws-iotexamples/tree/master/predictionDataSimulator
[48] B. Huang, Y. Di, C. Jin, and J. Lee, "Review of data-driven prognostics and health management techniques: Lessions learned from PHM data challenge competitions," *Mach. Failure Prevention Technol.*, vol. 2017, pp. 1–17, May 2017.

ABISHI CHOWDHURY (Member, IEEE) received the B.E. degree from the University Institute of Technology, West Bengal, India, in 2011, the M.Tech. degree from the National Institute of Technical Teachers' Training and Research, Bhopal, Madhya Pradesh, India, in 2014, and the Ph.D. degree from the Visvesvaraya National Institute of Technology, Nagpur, India, in 2020. She is currently an Assistant Professor with Vellore Institute of Technology, Chennai, India. Her research interests include cloud computing, cloud resource scheduling, machine learning, and the Internet of Things.

AMRIT PAL (Member, IEEE) received the B.Tech. degree from Kurukshetra University, Kurukshetra, India, in 2011, the M.Tech. degree from the National Institute of Technical Teachers' Training and Research, Bhopal, India, in 2014, and the Ph.D. degree from Indian Institute of Information Technology Allahabad, India, in 2020. He was an Assistant Professor with the Centre for Advanced Studies, AKTU, Lucknow, India. He is currently an Assistant Professor with Vellore Institute of Technology, Chennai, India. His research interests include big data analytics, cloud computing, machine learning, and the Internet of Things.



ASHWIN RAUT received the M.Tech. degree in software engineering from Indian Institute of Information Technology Allahabad, India. He is currently a Research Scholar with the Department of Information Technology, Indian Institute of Information Technology Allahabad. His research interests include the IoT, data analytics, data mining, and adaptive learning.

MANISH KUMAR (Senior Member, IEEE) received the Ph.D. degree from the Department of Information Technology, Indian Institute of Information Technology Allahabad (IIIT-Allahabad), Allahabad, India, in 2011. He is currently an Associate Professor with the Department of Information Technology, IIIT-Allahabad, where he is the Coordinator of the Data Analytics Laboratory. His research interests include big data analytics, machine learning and data analytics, the IoT and smart cities, and wireless sensor networks.