



# A new fuzzy MLE-clustering approach based on object-to-group probabilistic distance measure: from anomaly detection to multi-fault classification in datacenter computational nodes

Saloua El Motaki<sup>1</sup> · Badr Hirchoua<sup>2</sup> · Ali Yahyaouy<sup>1</sup>

Received: 4 January 2022 / Accepted: 15 June 2022 / Published online: 2 July 2022  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

## Abstract

Datacenters are expanding in size and complexity to the point where anomaly detection and infrastructure monitoring become critical challenges. One potential strategy for dealing with the reliability of computational nodes in a datacenter is to identify cluster nodes or virtual machines exhibiting anomalous behavior. Throughout this paper, we introduce a novel clustering approach for analyzing cluster node behavior while running various workloads in a system based on resource usage details (CPU utilization, network events, etc.). The new clustering technique aims at boosting the efficiency of fuzzy clustering algorithms based on the maximum likelihood estimation (MLE) scheme. We propose the use of a recently developed object-to-group distance since it does not involve the computation of distances among all pairs of objects to assign the objects to the most appropriate group. The experimental findings under realistic settings demonstrate that the newly implemented algorithm outperforms many similar algorithms that have been used frequently in such tasks.

**Keywords** Datacenter · Computational node · Anomaly detection · Fuzzy clustering · Object-to-group distance

## 1 Introduction

Datacenters are increasingly recognized as the primary IT infrastructure of the modern digital society. As datacenters are handling an ever-increasing number of users and increasingly advanced computing applications, the resulting workloads are undergoing rapid scalability. Many of these workloads experience substantial performance fluctuations. Orphaned processes dropped from previous runs; CPU throttling due to material failures; memory leaks; firmware faults; and shared resource contention constitute the common hardware- and software-related types of anomalies that lead to

the performance variation (Brandt et al. 2010; Agelastos et al. 2015; Snir et al. 2014; Bhatele et al. 2013).

Data center operators typically lack knowledge of what tasks are running in the system at some point. Many users are sharing physical hosts and simultaneously performing hundreds or thousands of applications per hour (Bari et al. 2013). No static workload analysis was proven to be effective in detecting the actual behavior of applications and tasks as they run (Egele et al. 2014). In addition, applications running in a virtualized or high-performance computing (HPC) environment incur a workload that is quantified in terms of memory usage, storage, network bandwidth, and processing overhead (El Motaki et al. 2019). In most cases, the amount of resources claimed by the customer is higher than the actual resource usage. Accordingly, identifying workload and application behavior using naive methods, such as considering information provided by users as a resource request or a predefined service level agreement (SLA), are not informative.

To deal with the above challenges, in this work, we target the adoption of a fuzzy clustering approach for workload characterization and anomaly identification in computational nodes. A new algorithm is presented to auto-detect compute nodes suffering from previously observed and unobserved

---

Badr Hirchoua and Ali Yahyaouy have contributed equally to this work.

---

✉ Saloua El Motaki  
saloua.elmotaki@usmba.ac.ma

<sup>1</sup> Computer Science, University Sidi Mohammed Ben Abdellah-USMBA, Fez, Morocco

<sup>2</sup> Computer Science, National Higher School of Arts and Crafts (ENSAM), Hassan II University, Casablanca, Morocco

anomalies and to identify the type of anomaly regardless of the applications running on the compute nodes. Using a novel unsupervised learning algorithm, our approach detects and diagnoses anomalies based on resource utilization measurements and performance statistics.

We propose a new fuzzy clustering algorithm that consists of the use of a new object-to-group probabilistic distance measure (PDM), introduced by Tavakkol et al. (2017). Clustering refers to the concept of grouping in different ways. In this paper, we are concerned with a clustering algorithm as a mapping function, into partitions of multiple instances of a finite; unlabeled; and multivariate data set. that is, the probabilistic or uncertain aspect of the features is taken into account when representing the data instances (observations). The key idea involves partitioning the dataset into groups (clusters) so that the data belonging to one cluster are as similar to each other as possible and as different as possible from the existing data pertaining to the other clusters. The similarities are determined by using new measure object-to-group PDM function between a particular uncertain object and a set of uncertain objects. Specifically, the contributions of this paper are the following:

- We introduce the fuzzy-PDM clustering algorithm: a novel clustering technique that aims at improving the efficiency of the fuzzy clustering algorithms based on the MLE concept, such as the Gath-Geva clustering algorithm (Gath and Geva 1989). We propose the use of the recently developed measure object-to-group PDM (Tavakkol et al. 2017), which does not involve the calculation of distances between all object pairs, to assign an object to the most appropriate cluster, which leads to fewer computation requirements.
- The proposed algorithm is used for two main tasks: (i) checking whether the anomaly occurs or not (regardless of its type) and (ii) the multi-fault classification task, in which the algorithm is used to recognize and classify each anomaly detected.
- To validate the accuracy of the proposed algorithm, we use a real-world dataset (monitoring data collected periodically at runtime from a computing cluster). The clustering results are compared to the ground truth (the real classification of the workloads) using clustering validation indices.

The motivation for developing a new unsupervised approach based on object-to-group distance measure for the classification of abnormalities is twofold: (i) the existing literature has shown that classification algorithms can accurately detect anomalous behavior of cluster nodes, improving monitoring and management of a datacenter (Bashir et al. 2019). However, the available techniques suffer from the lack of labelled data extracted from these systems. (ii) The existing fuzzy

MLE clustering approaches are proven to be effective for anomaly detection; yet, the high amount of data that should be processed requires a better computation strategy that optimizes the number of distance measure instructions. Thus, our algorithm enables us to maintain the high performance of the fuzzy MLE clustering algorithm while computing the distance between the item and all the group (of data) instead of each single datum of the group.

The remaining content of the paper is organized as follows. We review, in Sect. 2, the existing clustering algorithms applied to workload and application anomalies identification in a datacenter and HPC. Then, in Sect. 3, we briefly outline the background techniques that are useful to implement the proposed work. Section 4 describes the proposed clustering algorithm. Section 5 outlines the data pipeline involved in the experimental evaluation of our proposal. It provides brief descriptions of the data extraction, feature acquisition, and feature selection process. Using the resulting experimental dataset, Sect. 6 validates the proposed algorithm. In Sect. 7, we synthesize the present work and highlight some directions for future work.

## 2 Related work

A vast and expanding collection of literature has studied clustering methods for providing a compact representation of the policy behavior while performing diverse workloads. Rugwiro and Chunhua (2017) have investigated the K-means clustering technique for optimizing the allocation strategy for VMs. Their suggested clustering model considers transforming a collection of tasks to VMs. The authors have concentrated on jobs' CPU, memory, and bandwidth employment. They pointed to decreasing energy consumption by effective resource allocation. Khan et al. (2012) have studied a co-clustering technique to recognize workload patterns performed on a server that allows the forecast of every VM workload.

In Daradkeh et al. (2020), authors have proposed an improved, driven K-Means clustering method to fit the cloud elastic model relying on KDE and the Silverman model for the discovery of optimal clustering centers and class boundaries. Based on the kernel density estimator, the proposed method evaluates the workloads and configuration traces of the datacenter in search of class number and centroid. It considers the types of workloads based on the submitted job and the existing datacenter capabilities, as well as the capacity and configuration of servers. The suggested method could lower the complexity of the K-means and improve the clustering accuracy.

Abdelsalam et al. (2017) have elaborated an unsupervised learning system to identify anomalous VM behavior in situations indicating automated scaling in Infrastructure

as a Service (IaaS) clouds. The authors have adopted a K-means sequential clustering algorithm variant to discover abnormalities based on resource usage changes. These variants have been found while insiders or other malware try to execute malicious jobs on VMs of cloud clients.

Ismaeel et al. (2016) have introduced a model to forecast VM terms in a datacenter. The model fuses the K-means algorithm with the extreme learning machine (ELM). Precisely, it considers the clustering incorporation of users collectively with memory and CPU workload in a prediction method. Furthermore, they performed a comparative study connecting k-means clustering and FCM to examine VMs and user behavior, where each VM request is mapped to a particular cluster.

Likewise, the achievement of the principal component analysis (PCA) and K-means has been studied by Shirazi et al. (2014). These methods have been assessed in a monitored cloud testbed environment. The authors have considered attack and migration events occurring simultaneously or separately. This ended in measures such as performance measurement. They have imagined that the undefined amount of undesired attacks and false alerts generated while migrating VMs influence the VM behavior making it unpredictable. Hence, they have affirmed that the utilized methods can be adopted straightly in the live migration features. Essentially, this work has concentrated on the impact of VM migration on anomaly detection methods.

Researchers in Bi et al. (2019) have introduced a machine learning-based workload forecasting system named SGW-SCN. The SGW-SCN predicts the expected quantity of assignments coming in the future in geo-distributed cloud datacenters. The authors have combined the Savitzky-Golay filter and wavelet decomposition (SGW) with stochastic configuration networks (SCNs). The filter excludes potential noises and outliers and smooths the non-stationary workload time series, which are decomposed into various elements. With SCNs, a combined model is constructed to capture the statistical characteristics of the detail and trend elements.

For early failure detection, Amruthnath and Gupta (2018) have started a testbed to assess various unsupervised learning algorithms. They have adopted a basic vibration dataset assembled by an extraction fan and adjusted several unsupervised learning algorithms, including hierarchical clustering, K-Means, PCA, and FCM, among others. Next, they introduced a methodology to evaluate distinct algorithms and determine the decisive model.

In addition, Hui (2018) has designed a standard clustering algorithm for clustering data using a similarity metric. The VM performance is watched continuously by the cloud platform; hence, Hui has created the Incremental VM workload clustering algorithm to collect performance data with identical VM workloads inside a single cluster. This algorithm's

results have revealed that it could efficiently collect performance data with similar VM workloads in a unique cluster, which assists in the prevention of negative actions when anomalies are discovered.

Zhang et al. (2016) have proposed a task-level anomaly detection method in a software agnostic way. Using FN-DBSCAN, an unsupervised learning algorithm, they learned the patterned behavior given the appropriate task profiling degree measurements within the unlabeled historical data. Then, they utilized the clustering output to discover the possible performance anomaly. They built the association between the network connection and the task.

Pandeeswari and Kumar (2016) have introduced a hybrid system that used the FCM clustering algorithm alongside artificial neural networks (FCM-ANN) to identify the VMs' abnormal behavior. The system identifies the attack patterns saved before in a frequently updated database. Hence, in favor of avoiding the old-fashioned manual update of the database, the authors have considered the FCM clustering algorithm to automatically obtain the new attacks. Identically, in our earlier work (El Motaki et al. 2019), we have adopted the Gath-Geva clustering algorithm for revealing similarities between various applications operating on the HPC system.

Statistical methods have proven to provide an encouraging answer for anomaly detection. Lorido-Botran et al. (2017) have developed an online unsupervised anomalies detection algorithm. The algorithm analyzes models for diverse lags. They adopted Dirichlet Process Gaussian Mixture Models to identify the resource usage profile and distance measures to obtain the similarity among various models.

Sauvanaud et al. (2018) have introduced an anomaly detection system (ADS), which detects errors and service level agreements violations while affording two investigation stages: distinguishing the anomalous virtual machine and the error's class making the anomaly. The ADS incorporates a system monitoring entity that gathers software counters describing the cloud service alongside a detection entity. This second entity is based on machine learning models trained through a fault injection entity (FIE). Furthermore, The FIE is adopted to verify and evaluate the ADS.

Authors in Xiao et al. (2021) consider two concerns of operation and maintenance (O & M), the cluster machines' unnoticed release and stability detection. They have proposed the Ensemble Learning on Partition Interval (ELPI) method for anomaly detection. On the one hand, it starts by splitting the data set into two main intervals: the stable and the unsteady. This splitting encourages diversification in the used methods. On the other hand, they introduce a module and execute related integrated learning for various factors related to each interval to highlight abnormal data.

In our previous paper (El Motaki et al. 2021), we have introduced a new clustering algorithm to analyze the running

behaviour of virtual machines. The introduced method is aimed at recognizing workloads that over-utilize shared resources. We have considered the CPU and memory utilization, as well as the downtime occurrence, as the main metrics. Based on the experimental results, it was found that sbWFCM was able to generate reasonable partitions for given numbers of iterations according to a domain expert's validation.

### 3 Background

Let consider a numerical vector-based feature data  $X = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^d$ , representing a set of objects  $O = \{o_1, o_2, \dots, o_N\}$ . The coordinates of  $x_i$  are the patterns that describe the object  $o_i$ . A partition matrix corresponding to the patterns  $x_i$  is established as a set of values  $u_{ki}$ , where  $k = 1, \dots, C$ , and  $C \leq N$  represents the number of clusters. Each value (in  $[0, 1]$ ) depicts the membership degree of the pattern  $x_i$  to the  $k$ th cluster, forming the vectors  $u_i$  which are arrayed into  $C \times N$ -matrix  $U = [u_{ki}]_{C \times N}$ .

Generally, the clustering algorithm consists of assigning each element in  $X$  to a partition  $P \subset \mathcal{P}$ , where  $\mathcal{P}$  is the set of all data partitions. Furthermore, each partition  $P$  has a prototype (center) denoted by the matrix  $V = [v_k]$ . Each element  $v_k$  refers to the center of the  $k$ th cluster.

Fuzzy c-means is the most widely applied fuzzy clustering algorithm in literature. The fuzzy c-means algorithm is defined as the constrained minimization of the following function:

$$Q_{FCM} = \sum_{i=1}^N \sum_{k=1}^C u_{ki}^m D^2(x_i, v_k) \quad (1)$$

Where  $\sum_{i=1}^N u_{ki} > 0$ ,  $\sum_{k=1}^C u_{ki} = 1$ , and  $m > 1$  is the fuzziness parameter that regulates the the overlapping among the clusters. The optimization of Eq. 1 through an iterative process builds the fuzzy partitions. The algorithm is outlined in Algorithm 1.

---

#### Algorithm 1 Fuzzy c-means clustering algorithm

---

**Input:**  $X$ ,  $m$ , initial  $C$ , and  $\epsilon$

- 1: Choose primary centroids  $v_k$
- 2: **while**  $(u_{ki}^{(t+1)} - u_{ki}^{(t)} \geq \epsilon)$  **do**
- 3:     Compute the membership degree of all feature vectors in all clusters

$$u_{ki}^{(t)} = \frac{1}{\sum_{j=1}^C \left( \frac{D^2(x_i, v_k)}{D^2(x_i, v_j)} \right)^{\frac{2}{m-1}}} \quad (2)$$

- 4:     Update the centroids  $v_k$

$$v_k^{(t)} = \frac{\sum_{i=1}^N u_{ki}^m x_i}{\sum_{i=1}^N u_{ki}^m} \quad (3)$$

- 5: **end while**
- 

Computing the degree of membership is based on the defined distance measure  $D$  which is given as follows:

$$D^2(x_i - v_k) = (x_i - v_k)^T A (x_i - v_k) \quad (4)$$

Including  $A$  (a positively defined matrix) in the distance metric allows to assign a proportional weight based on the statistical properties of each characteristic. According to Gath and Geva (1989), two alternative distance metrics can be defined and used at two different levels of the clustering process: (i) if  $A$  is equal to the identity matrix, the distance is Euclidean which refers to the fuzzy c-mean algorithm. (ii) For clusters of varying densities and unequal numbers of data points in each cluster, an exponential distance metric  $D_{exp}$ , derived from the MLE, is established and given as follows:

$$D_{exp}^2(x_i, v_k) = \frac{\sqrt{\det(F_k)}}{P_k} \exp \left[ \frac{1}{2} (x_i - v_k)^T F_k^{-1} (x_i - v_k) \right] \quad (5)$$

$F_k$  denotes the fuzzy covariance matrix of the  $k$ th cluster, and  $P_k$  represents the a priori probability of selecting the  $k$ th cluster which are computed as follows:

$$F_k = \frac{\sum_{i=1}^N h(k | x_i) (x_i - v_k)(x_i - v_k)^T}{\sum_{i=1}^N h(k | x_i)} \quad (6)$$

$$P_k = \frac{1}{N} \sum_{i=1}^N h(k | x_i) \quad (7)$$

We note that  $h(k | x_i)$  is the posterior probability for selecting the  $k$ th cluster considering the  $i$ th feature vector.

$$h(k | x_i) = \frac{1}{\sum_{j=1}^C \left( \frac{D_{exp}^2(x_i, v_k)}{D_{exp}^2(x_i, v_j)} \right)^{\frac{2}{m-1}}} \tag{8}$$

Fuzzy c-means is considered the simplest and the most flexible fuzzy clustering algorithm; however, it does not provide optimal partitioning in the case of clusters with varying and irregular shapes and densities. The fuzzy MLE algorithms have been introduced to solve this last problem, and they have provided a considerable improvement in clustering results by generating an optimal partition even with significant variability of cluster shapes and densities. All the same, this algorithm can converge to a local optimum in a relatively small region due to the incorporated exponential distance function; it is, therefore, less efficient and may be potentially unstable for unsupervised identification of classifier prototypes. Moreover, computing distance between items remains a serious challenge with the ever-growing amount of data to be processed and considered for the learning process.

To develop a new fuzzy clustering algorithm, which can overcome the limitations of both approaches mentioned above, we propose a technique that follows a novel strategy for defining the distance metric between an object and its appropriate cluster.

### 4 Fuzzy-PDM clustering algorithm

The adopted clustering algorithm is implemented following the same concept as a fuzzy MLE algorithm, and it uses the object-to-group probabilistic distance measure that considers the correlation between characteristics within each class and within each object independently. The distance metric is described as follows:

$$D_{og}^2 = w_1 \log \left[ \frac{f_i + \bar{f}_k}{2(|f_i| || \bar{f}_k|)^{\frac{1}{2}}} \right] + w_2(x_i - v_k)^T F_k^{-1}(x_i - v_k) \tag{9}$$

The first term consists of the difference between covariance matrices, where  $f_i$  represents the object-covariance matrix of the object  $x_i$ , and  $\bar{f}_k$  is the average of the covariance matrices of the objects in the  $k$ th cluster and is given below.

$$\bar{f}_k = \frac{1}{n_k} \sum_{l=1}^{n_k} f_l^k \tag{10}$$

Where  $n_k$  is the current number of elements in the  $k$ th cluster and is updated whenever a new object is added to the cluster.

It is inspired by the Bhattacharya distance, introduced by Bhattacharyya (1946). One reason to choose Bhattacharyya’s PDM is that our modeling of uncertain objects is based on multivariate normal pdfs. Moreover, this type of measure (Bhattacharyya or similar PDMs) can be calculated for all general probability distributions by numerical methods.

The second term of Eq. 9 is the Mahalanobis distance defined between an object  $x_i$  and the cluster prototype  $v_k$ .  $w_1$  and  $w_2$  correspond to the weights associated with the two terms; they allow to find the best compromise between the distance between the object to be added to the cluster and the center (prototype) of this cluster, and, on the other hand, the distance between the object and the cluster as a whole (given by the first term of Eq. 9.).

The distance between a given item picked and the cluster of leftover items can vary for every randomly sampled object in the probabilistic distance metric that we provide. A variety of weighting values may be appropriate for the newly formed object-group PDM. For simpler purposes, our paper considers equal weights for the two terms  $w_1$  and  $w_2$ . The implementation procedure of the proposed Fuzzy-PDM clustering algorithm is detailed in Algorithm 2.

**Algorithm 2** Fuzzy-PDM clustering algorithm**Input:**  $X$ ,  $m$ , initial  $C$ , and  $\epsilon$ 

- 1: Choose primary partition  $u_{ki}$
- 2: **do**
- 3:    Compute the centroids  $v_k$ , as well as the number of items  $n_k$

$$v_k^{(t)} = \frac{\sum_{i=1}^N u_{ki}^m x_i}{\sum_{i=1}^N u_{ki}^m} \quad (11)$$

- 4:    Compute the covariance matrix  $F_k$

$$F_k = \frac{\sum_{i=1}^N h(k | x_i)(x_i - v_k)(x_i - v_k)^T}{\sum_{i=1}^N h(k | x_i)} \quad (12)$$

With the a posterior probability  $h(k | x_i)$ 

$$h(k | x_i) = \frac{1}{\sum_{j=1}^C \left( \frac{D_{og}^2(x_i, v_k)}{D_{og}^2(x_i, v_j)} \right)^{\frac{2}{m-1}}} \quad (13)$$

- 5:    Update the membership degree of all feature vectors in all clusters.

$$u_{ki}^{(t)} = \frac{1}{\sum_{j=1}^C \left( \frac{D_{ob}^2(x_i, v_k)}{D_{ob}^2(x_i, v_j)} \right)^{\frac{2}{m-1}}} \quad (14)$$

- 6: **while** ( $\| u_{ki}^{(t)} - u_{ki}^{(t-1)} \| \geq \epsilon$ )

We note that the choice of the primary partition  $u_{ki}$  is important for an optimal performance of the algorithm. In this work, we initialize this parameter using fuzzy c-means algorithm.

## 5 Material and methods

### 5.1 Data extraction

Our work aims to recognize if any compute node accurately is anomalous (i.e., behaving abnormally) and to classify the anomaly type, regardless of which workload is running on the compute node. We exploit the periodic machine, energy, and workload data collected from an internal cluster to monitor the overall system's operational health.

The cluster has been serving as a facility for more than 400 University students and researchers. Within the time span of the dataset, they have submitted thousands of jobs, with the majority coming from the disciplines of Physics, Chemistry, Bioinformatics, Computer Science, and Machine Learning. The Jobs are exclusively per user; currently, no multi-user jobs or workflows exist. In addition, SLURM is used as the physical resource manager for multi-user, sequential and parallel machines to allow users to provide scheduling and job planning. The per-stakeholder FIFO method is used to schedule all work, with an equitable sharing among stakeholders. The datacenter provides co-allocation of jobs on the same node as well as exclusive usage of nodes through queuing.

In total, the cluster used for testing is composed of 19 nodes that offer the following capabilities: 760 cores (68 TFlops); 5.2 TB of Memory; 108 TB of Storage; and 2 GPU cards. These nodes are interconnected by a very low latency network (OPA) at 100 Gbps, which allows optimizing performance for parallel calculations.

### 5.2 Synthetic anomalies

To mimic anomalies observed in real systems, we perform synthesized anomalous events on a cluster to stress single node components (such as CPU or memory), thereby simulating an impairment or dysfunction occurring in the respective node component. We use the programs (described in Table 1) at random anomaly intensities. The resulting anomalies can arise from system or application failures. Typical cases of such abnormalities include the following:

- Orphan processes: processes that keep consuming system resources such as memory and CPU once a job is finished inappropriately (Brandt et al. 2010).
- Out-of-memory: when the total memory of the system is expended (typically due to memory leaks), the running application is interrupted by the Linux out-of-memory killer (Agelastos et al. 2015).
- Undisclosed Physical Assets: a drop in system performance caused by the auto-compensation functions used for hardware failures. A typical case of such problems occurred when some nodes experienced a performance degradation in which different CPUs run at lowered speed (Snir et al. 2014).

### 5.3 Data clean-up

After having inspected the data, we applied a clean-up process on (in)valid and missing data. We have created a clean-up script-based following different axes: (i) a node-level

**Table 1** The programs used to produce anomalous conditions in the cluster

Program	Description
Ddot	It assigns two equal-sized matrices of a dual-type and fills them using memalign. The scalar product of the two matrices is computed repetitively, resulting in a periodic change in the cache size of CPU varying between 0.9, 5 and 10. As a result, an interference gap between the CPU and the caches is caused; thus the CPU's performance is degraded.
Dcopy	It also allocates two equal-sized matrices as ddot, but it repeatedly maps one matrix to the other. Unlike ddot, it causes less CPU interference.
Leak	It allocates a 16 MB array and loads it with a bunch of characters. It is paused periodically within an internal loop for two seconds. Subsequently, the memory used is never released, leading to a memory leak.
Memeater	It involves allocating an array of integers of 36 MB and loading it with random entries. It increments the array size regularly via <i>realloc</i> and adds new values to the array. Subsequently, the system performance is decreased because of bandwidth saturation and a memory interference.

**Table 2** An overview of the machine dataset measurement

Dataset	Aspect	Value
Extracted dataset	Starting date	01-03-2019
	Ending date	30-07-2019
	Sampling rate (s)	30
	Max. samples per metric per node	629,320
	No. of measurements	16,653,474,810
Cleaned dataset	No. of valid nodes	14
	No. of valid measurements	13,984,822,911

clean-up, in which we consider 14 nodes used for computational purposes exclusively. The nodes collectively include 632 CPU cores, 90 TB of CPU space, and 2 GPUs. 12 nodes among the selected ones include only CPUs, yet the two come with GPUs onboard. (ii) A job-based clean-up in which we eliminate jobs based on the time they were initiated when they occurred outside the time range of the record set. (iii) A metric-based clean-up that consists of discarding the NaN values or zeroing them out, such as in summing during the numerical scans. The initial dataset collectively comprises more than 16 billion records, including nearly 0.8 billion NaN values (4.81%). We note that the data set shows missing data for certain measures in cases where the monitoring system was down. Eventually, the resulting cleaned data is presented in Table 2.

## 5.4 Feature selection

Datacenter monitoring frameworks typically capture hundreds of measures related to resource usage and system performance per compute node. To run abnormality detection using a reduced subset of such measurements and to compute only the statistical patterns that might be relevant for diagnosis can lead to considerable savings in terms of computational costs. Besides, the accuracy of clustering

algorithms may be greatly increased through a prior feature selection process.

To this end, we follow, in this paper, the feature selection technique provided in Genuer et al. (2010). For easy reference, we review briefly the basic concept. We select a feature  $X_j$  to obtain the maximum information gain from the data set  $X$  or to optimize the entropy. Simply put, we maximize  $I(X, X_j) = H(X) - H(X, X_j)$ .  $H(X)$  stands for the entropy of the data set prior to the selection of a feature; it is given by  $H(X) = -\sum_{i=1}^C p_i \log_2 p_i$ . The probability  $p_i$  of the  $i$ th-class is given by  $p_i = \frac{n_i}{N}$ , where  $n_i$  denotes the actual number of items in the  $i$ th class  $C$ ;  $i = (1, \dots, C)$  and  $N$  denotes the cardinality of  $X$ . Besides, the entropy after  $j$ th feature selection (so-called the conditional entropy) is expressed by  $H(X, X_j) = -\sum_x p(X_j = x) \sum_y p(X = y | X_j = x) \log_2 p(X = y | X_j = x)$ .

This method has proven to be more rapid and more selective than the commonly used random forest (RF) and genetic algorithm (GA) feature selection techniques (Tuncer et al. 2017). Namely, using this method, we selected only 8 out of 850 relevant features, as opposed to the GA-based method, which selected more features. This basically reflects the fact that the employed method is considered as a special purpose approach to estimating each feature's individual contributivity; However, GAs are non-special purpose optimizers driven by a fitness function.

## 6 Clustering results and discussion

In this section, the proposed algorithm results are compared with the results of three other clustering algorithms. The choice of these algorithms for the evaluation of our approach is due to our intention to compare our algorithm with alternatives that perform almost the same computation but are based on other strategies and other ways of distance calculation, including the Gath-Geva clustering algorithm; Gustafson-Kessel clustering; and the fuzzy neighborhood density-based spatial clustering with noise algorithm (FN-DBSCAN). For

more details about the implementation of the algorithms, the interested reader is referred to Gath and Geva (1989), Gustafson and Kessel (1978), Nasibov and Ulutagay (2009). Then, we measure the quality of the resulting partitions through the use of the external quality indices: Rand Index (RI) and Adjusted Rand Index (ARI) (Hubert and Arabie 1985).

We consider two main scenarios. The first scenario consists of inspecting the anomaly recognition case, i.e., our only interest is to check if an anomaly (regardless of its type) occurs. This is equivalent to a clustering run with two clusters  $C = 2$ . The second scenario concerns the case in which the number of clusters corresponds to the actual health conditions of the workloads. Moreover, the algorithms are independently executed 40 times given the same initial conditions (for each scenario). The goodness of the obtained partitions was then statistically assessed by non-parametric Friedman and Wilcoxon signed-rank tests (Hubert and Arabie 1985). We contemplate a 95% confidence interval represented by a significance level  $\alpha = 0.05$ . In other words, if the  $p$  value  $< 0.05$ , we can assume that there is a statistically significant difference between the investigated outcomes (rejection of the null hypothesis). Alternatively, no statistically significant difference among the results is reported.

By performing the proposed algorithm on the 8 selected features, two reasonable partitions with  $C = 2, 4$  were discovered. Then, We run the four algorithms (DBSCAN, GG, GK and the new algorithm) 40 times for two and four clusters regarding the same experimental setup. Then, the quality of the different structural alternatives generated by the algorithms is assessed by the External Quality RI and ARI, which were statistically evaluated. For data visualization, the classical Sammon data dimensionality reduction method (Sammon 1969) is employed. Moreover, the hyper-parameters we adopted are outlined in Table 3. Without claiming the optimality of the hyper-parameters used (since there was no explicit approach to tuning the algorithms), FN-DBSCAN was the most daunting algorithm to configure.

### 6.1 The anomaly detection case

The first and basic case is the anomaly detection check, in which we are interested in whether the system shows abnormal behavior, regardless of the workload being executed, the virtual machine or the server used for execution. This case can be interpreted as having a number of clusters  $C = 2$ . A Sammon projection of the 8-dimensional feature space is exhibited in

Fig. 1. The anomalous patterns are represented by red dots (•).

From Fig. 1, we can distinguish small groups of red dots that appear in different regions of the projection space. This is explained by the fact that the proposed algorithm was able to detect anomalies. It interprets them as abnormal behaviors caused by the system or a system's component (hardware or software). These anomalies are due either to an excessive use of resources or to hardware failures.

Our algorithm is compared with three other algorithms (the most similar unsupervised approaches) through external validation, where the distribution of RI and ARI values is illustrated in Fig. 2a and b, respectively. From the analysis of the RI index (Fig. 2a), it was found that for anomaly detection case, a  $p_{Friedman} = 2981.8e^{-15}$  is obtained, which indicates that the algorithms' respective results differ statistically in a significant way in this case. A more refined examination reveals that for a number of clusters  $C = 2$ , the highest clustering quality is achieved by the Gath-Geva clustering algorithm, and the lowest quality is obtained by the FN-DBSCAN approach. Our algorithm, on its part, provides

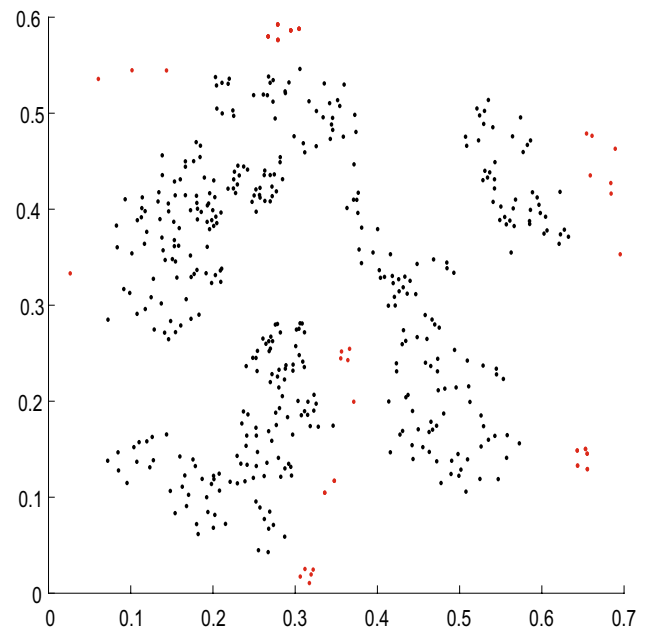


Fig. 1 The optimal clustering results provided by our algorithm as assessed by the RI for the anomaly detection case ( $C = 2$ )

**Table 3** The hyper-parameters used to benchmark the four algorithms

N of clusters	DBSCAN		GG		Ours		GK	
	$\epsilon$	$\nu$	$m$	$\rho_i$	$m$	$\epsilon$	$m$	$\rho_i$
2	0.9855	10	2	1.0	2	0.4	2	1.0
4	0.991	10	2	1.0	2	0.4	2	1.0



good partitioning. This confirms that the algorithm we propose is not only computationally lighter than the Gath-Geva method, but it also gives a clustering result almost equivalent to it Tables 4 and 5.

Moreover, the dispersion of the ARI index values (shown in Fig. 2b) corresponds to  $p_{Friedman} = 1.47e^{-16}$  indicates the significant statistical difference between the algorithms compared. In addition, the analysis of the resulting ARI index reveals that our algorithm provides a best performance in comparison with the other three clustering algorithms.

### 6.2 The multi-fault clustering case

A more detailed perspective is achieved by using the proposed algorithm to determine the type of anomaly detected. The Sammon projection of the 8-dimensional feature space into the plan is exhibited in Fig. 3. From the analysis of this figure, it follows that the proposed algorithm allows defining four different clusters as the best partitioning alternative. The anomaly clusters defined in the case of identification of abnormal behavior have a more meaningful meaning with this representation. For example, the anomaly group associated with the blue cluster represents the type of anomaly corresponding to the leak program that consists of allocating a 16 MB array and loading it with characters. The two-second periodic pause within an internal loop causes an eventual

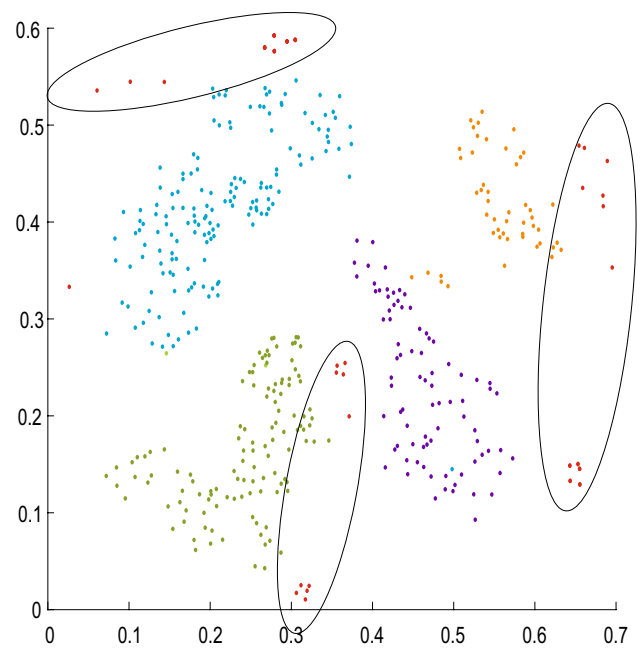


Fig. 3 The optimal clustering results provided by our algorithm as assessed by the RI for the multi-fault classification case ( $C = 4$ )

anomaly (memory leak) represented by the red dots shown on the top of the cluster.

Curiously enough, the subgroup of anomalies that lies between the green cluster (representing ddot) and the purple cluster (representing dcopy, respectively) is misleading. It can refer to anomalies due to the execution of ddot, i. e., it is a degradation of CPU performance due to the interference between CPU and caches, whereas it can be understood as a low CPU interference due to the repetition of the matrices maps. Eventually, the interpretation of the results, in this

Table 4 RI and ARI values achieved by the proposed algorithm over 40 independent runs for 2 clusters

	FN-DBSCAN	GG	Ours	GK
RI	0.512	0.625	0.608	0.578
ARI	0.145	0.391	0.838	0.478

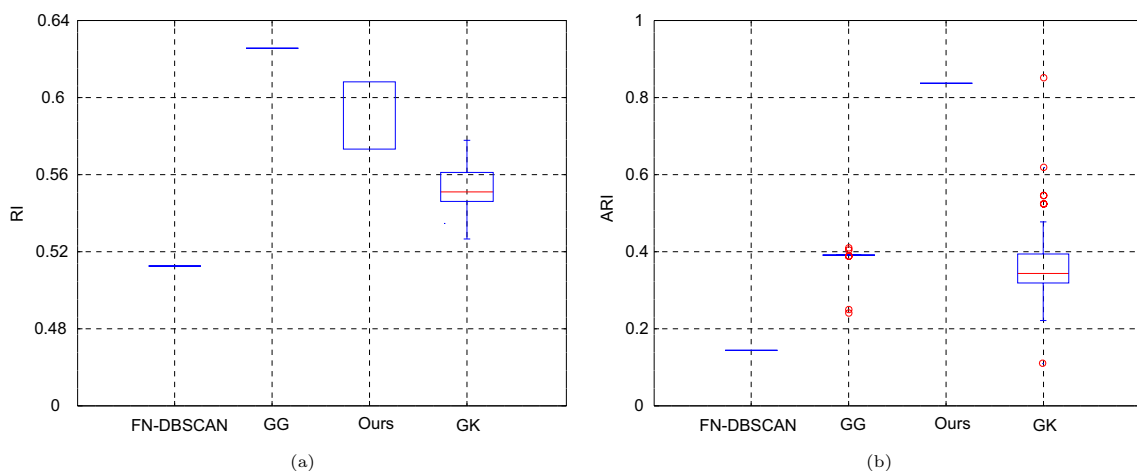


Fig. 2 Boxplots indicating the scattered a RI and b ARI values achieved by the proposed algorithm over 40 independent runs for 2 clusters

**Table 5** RI and ARI values achieved by the proposed algorithm over 40 independent runs for 4 clusters

	FN-DBSCAN	GG	Ours	GK
RI	0.857	0.898	0.973	0.905
ARI	0.462	0.465	0.901	0.757

case, returns to the domain expert who is entrusted with the analysis of the clusters.

For further analysis, again, the algorithm proposed in this work is benchmarked against three other algorithms of its type using external validation, and the observed dispersions of the RI and ARI values are reported in Fig. 4. From the analysis of Fig. 4a, it follows that we obtain a  $p_{Friedman} = 1.24e^{-17}$ , which implies that a statistically significant difference does exist among the results of the algorithms. In terms of partitioning quality, we can see that our algorithm provides the best distribution; however, FN-DBSCAN is the worst.

Likewise, a  $p_{Friedman} = 1.85e^{-12}$  is reached in Fig. 4b, which confirms the existing difference given by the RI. Compared to the ground truth, the partition provided by our clustering algorithm has an  $RI \approx 0.97$ . Typically,  $RI > 0.7$  reflects an appropriate clustering alternative since such values may not be consistent with random cross-partition agreements.

Eventually, it may be assumed that the distance computation technique on which our algorithm is based has a significant impact in multivariate feature spaces, as our benchmark that needs more complicated approaches to understand the data extracted from a computational cluster and identify the potential abnormal behaviors of workloads that run over different computational nodes.

## 7 Conclusion

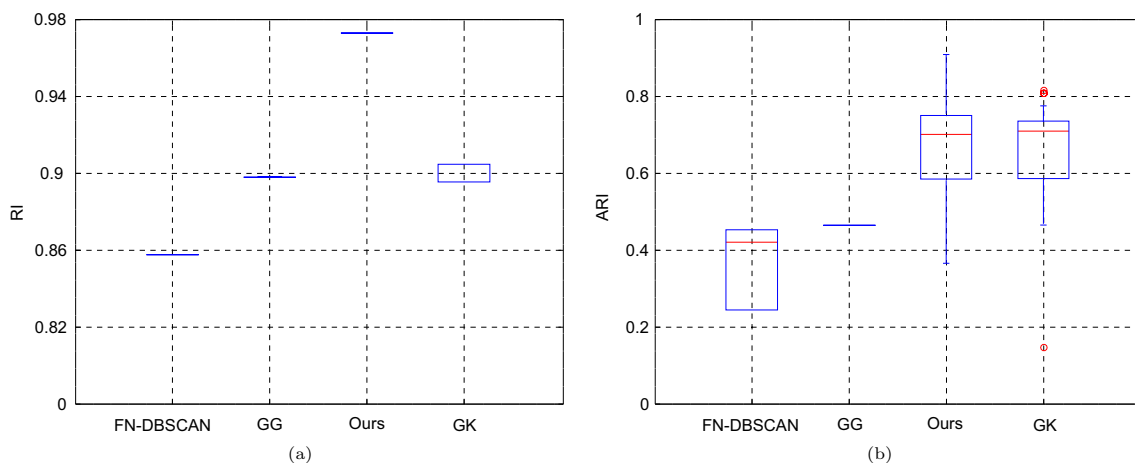
In this paper, we focused on anomaly detection problems in computing clusters in a datacenter, aiming to enable solutions that improve monitoring and control with a new automated method that identifies abnormal behavior of computing nodes while running both known and unknown workloads.

Previous proposals to address this problem exploited classification, which is supervised learning. While this approach is viable for identifying anomalies with high accuracy, it requires prior knowledge of anomalous behavior in the form of labelled data; thus, it is difficult to apply in datacenter infrastructures where physical hosts are shared by many users and simultaneously run hundreds or thousands of applications per hour. This limitation motivates our proposal of a new unsupervised learning technique that adopts the recently introduced distance measure object-to-group of a particular uncertain object with a set of uncertain objects.

The proposed clustering algorithm in this paper has been designed to overcome the limitation imposed by clustering algorithms based on the centroid scheme, but it maintains the simplicity of the partition extraction process of these algorithms.

To provide the most realistic operating conditions possible, we evaluate our algorithm using exclusively real-world data from a benchmark dataset of computational nodes extracted and cleaned by our research team, where both faults interference and diverse workload requirements can be explored.

The performance evaluation has accounted for the quality of the resulting partitions, as measured by the external validation indices RI and ARI. Statistical hypothesis tests have shown that, for our setup, where eight features were

**Fig. 4** Boxplots indicating the scattered **a** RI and **b** ARI values achieved by the proposed algorithm over 40 independent runs for 4 clusters

selected as the most relevant to characterize the state of the nodes, our algorithm and the GG algorithm showed the highest performance in the detection case (2 clusters) and for the multi-fault identification case (4 clusters), respectively.

This paper is a follow-up work in a developing line of research that focuses on applying clustering techniques to anomaly detection problems in computational clusters. Future research will be centred on further investigation of the suggested models, with supplementary vulnerability analyses of the model settings, the proposal of additional unsupervised models, and the application of new distance metrics to improve the quality of partitioning in cases where the data are multi-variate or uncertain.

## Appendix: p-Friedman test

The Friedman test is an extension of the Wilcoxon Signed Rank Test and the non-parametric equivalent of the 1-factor analysis of variance with repeated measures (Hubert and Arabie 1985). The Friedman test assumes the null hypothesis that  $k$  dependent variables belong to the same population. For the position parameter of a sample  $i$  by  $M_i$ , we denote the null hypothesis by  $H_0$  and the alternative hypothesis by  $H_a$  by the following:

$$H_0 : M_1 = M_2 = \dots = M_k$$

$$H_a : M_i \neq M_j \text{ for at least one } (i, j)$$

pair of the Friedman test.

Given the Friedman null hypothesis, the expected summed ranks of each group are equal to  $\frac{n(k+1)}{2}$ . The Friedman test statistic is expressed as follows:

$$Q = \frac{12}{nk(k+1)} \sum_{j=1}^k \left\{ R_j - \frac{n(k+1)}{2} \right\}^2 \quad (15)$$

where  $R_i$  is the sum of the ranks for the sample  $i$ .

**Acknowledgements** The experimental work was developed at the HPC-MARWAN computing cluster of the Mohammed V University In Rabat, Morocco.

**Author contributions** Conceptualization: SEM; Formal analysis and implementation: SEM; Writing—original draft preparation: SEM; Writing—review and editing: SEM and BH; Supervision: AY.

**Funding** The authors of this paper have not received any financial support for research, authorship and/or publication of this article.

**Data availability** The data that support the findings of this paper are available from the corresponding author, Saloua El Motaki, upon reasonable request.

## Declarations

**Conflict of interest** The authors of this paper declare that they have no significant competing financial, professional, or personal interests that might have influenced the performance or presentation of this work.

## References

- Abdelsalam M, Krishnan R, Sandhu R (2017) Clustering-based iaas cloud monitoring. In: 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), pages 672–679. <https://doi.org/10.1109/CLOUD.2017.90>
- Anthony A, Benjamin A, Jim B, Ann G, Sophia L, Steve M, Jeff O, Mahesh R, Joel S (2015) Toward rapid understanding of production hpc applications and systems. In: 2015 IEEE International Conference on Cluster Computing, pages 464–473. <https://doi.org/10.1109/CLUSTER.2015.71>
- Amruthnath N, Gupta T (2018) A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In: 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), pages 355–361. <https://doi.org/10.1109/IEA.2018.8387124>
- Bari MF, Boutaba R, Esteves R, Granville LZ, Podlesny M, Rabbani MG, Zhang Q, Zhani MF (2013) Data center network virtualization: a survey. *IEEE Commun. Surv. Tutor.* 15(2):909–928
- Bashir M, Irfan A, Hassan U, Muhammad Y (2019) Failure prediction using machine learning in a virtualised hpc system and application. *Clust. Comput.* 22:471–485. <https://doi.org/10.1007/s10586-019-02917-1> (ISSN 1573-7543)
- Bhatele A, Mohror K, Langer SH, Isaacs KE (2013) There goes the neighborhood: performance degradation due to nearby jobs. In: SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pages 1–12. <https://doi.org/10.1145/2503210.2503247>
- Bhattacharyya A (1946) On a measure of divergence between two multinomial populations. *Sankhya: The Indian Journal of Statistics (1933-1960)*, 7(4):401–406, ISSN 00364452. URL <http://www.jstor.org/stable/25047882>
- Bi J, Yuan H, Zhang LB, Zhang J (2019) Sgw-scnc: an integrated machine learning approach for workload forecasting in geodistributed cloud data centers. *Information Sciences*, 481:57–68, ISSN 0020-0255. <https://doi.org/10.1016/j.ins.2018.12.027>. URL <https://www.sciencedirect.com/science/article/pii/S0020025518309642>
- Brandt J, Chen F, De Sapio V, Gentile A, Mayo J, Pèbay P, Roe Di, Thompson D, Wong M (2010) Quantifying effectiveness of failure prediction and response in hpc systems: methodology and example. In: 2010 International Conference on Dependable Systems and Networks Workshops (DSN-W), pages 2–7. <https://doi.org/10.1109/DSNW.2010.5542629>
- Daradkeh T, Agarwal A, Zaman M, Goel N (2020) Dynamic k-means clustering of workload and cloud resource configuration for cloud elastic model. *IEEE Access* 8:219430–219446. <https://doi.org/10.1109/ACCESS.2020.3042716>
- Egele M, Woo M, Chapman P, Brumley D (2014) Blanket execution: dynamic similarity testing for program binaries and components. In: 23rd {USENIX} Security Symposium ({USENIX} Security 14), pages 303–317
- El Motaki S, Yahyaouy A, Gualous H, Sabor J (2019) Gath-geva clustering algorithm for high performance computing (hpc) monitoring. In: 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), pages 1–6

- El Motaki S, Yahyaouy A, Gualous H, Sabor J (2021) A new weighted fuzzy c-means clustering for workload monitoring in cloud data-center platforms. *Clust. Comput.* 24(4):3367–3379. <https://doi.org/10.1007/s10586-021-03331-2> (ISSN 1573-7543)
- Gath I, Geva AB (1989) Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern. Anal. Mach. Intell.* 11(7):773–780
- Genuer R, Poggi J-M, Tuleau-Malot C (2010) Variable selection using random forests. *Pattern Recognit. Lett.* 31(14):2225–2236
- Gustafson D, Kessel W (1978) Fuzzy clustering with a fuzzy covariance matrix. 1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes, pages 761–766
- Hubert L, Arabie P (1985) Comparing partitions. *J. Classif.* 2(1):193–218. <https://doi.org/10.1007/BF01908075> (ISSN 1432-1343)
- Hui Y (2018) A virtual machine anomaly detection system for cloud computing infrastructure. *J. Supercomput.* 21:6126–6134. <https://doi.org/10.1007/s11227-018-2518-z>
- Ismaeel S, Miri A, Al-Khazraji A (2016) Energy-consumption clustering in cloud data centre. In: 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), pages 1–6
- Khan A, Yan X, Tao S, Anerousis N (2012) Workload characterization and prediction in the cloud: A multiple time series approach. In: 2012 IEEE Network Operations and Management Symposium, pages 1287–1294
- Lorido-Botran T, Huerta S, Tomás L, Tordsson J, Sanz B (2017) An unsupervised approach to online noisy-neighbor detection in cloud data centers. *Expert Systems with Applications*, 89:188–204, ISSN 0957-4174. <https://doi.org/10.1016/j.eswa.2017.07.038>. <https://www.sciencedirect.com/science/article/pii/S0957417417305158>. Accessed 6 June 2022
- Nasibov EN, Ulutagay G (2009) Robustness of density-based clustering methods with various neighborhood relations. *Fuzzy Sets Syst.* 160(24):3601–3615
- Pandeeswari N, Kumar G (2016) Anomaly detection system in cloud environment using fuzzy clustering based ann. *Mob. Netw. Appl.* 21:494–505. <https://doi.org/10.1007/s11036-015-0644-x>
- Rugwiro U, Chunhua G (2017) Customization of virtual machine allocation policy using k-means clustering algorithm to minimize power consumption in data centers. In: Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing, New York, NY, USA, Association for Computing Machinery. ISBN 9781450347747. <https://doi.org/10.1145/3018896.3018947>
- Sammon JW (1969) A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* 100(5):401–409
- Sauvanaud C, Kaâniche M, Kanoun K, Lazri K, Da Silva SG (2018) Anomaly detection and diagnosis for cloud services: Practical experiments and lessons learned. *Journal of Systems and Software*, 139:84–106, ISSN 0164-1212. <https://doi.org/10.1016/j.jss.2018.01.039>. <https://www.sciencedirect.com/science/article/pii/S0164121218300256>. Accessed 2 June 2022
- Shirazi N, Simpson S, Marnierides AK, Watson M, Mauthe A, Hutchison D (2014) Assessing the impact of intra-cloud live migration on anomaly detection. In: 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), pages 52–57. <https://doi.org/10.1109/CloudNet.2014.6968968>
- Snir M, Wisniewski R W, Abraham JA, Adve SV, Saurabh B, Pavan B, Jim B, Pradip B, Franck C, Bill C, Chien AA, Paul C, Debardeleben NA, Diniz PC, Christian E, Mattan E, Saverio F, Al G, Rinku G, Fred J, Sriram K, Sven L, Dean L, Subhasish M, Todd M, Rob S, Jon S, Eric Van H (2014) Addressing failures in exascale computing. *Int. J. High Perform. Comput. Appl.* 28(2):129–173. <https://doi.org/10.1177/1094342014522573> (ISSN 1094-3420)
- Tavakkol B, Jeong Myong K, Albin Susan L (2017) Object-to-group probabilistic distance measure for uncertain data classification. *Neurocomputing* 230:143–151. <https://doi.org/10.1016/j.neucom.2016.12.007> (ISSN 0925-2312)
- Tuncer O, Ates EC, Zhang Y, Turk A, Brandt JM, Leung VJ, Egele M, Coskun AK (2017) Diagnosing performance variations in hpc applications using machine learning. In: ISC. [https://doi.org/10.1007/978-3-319-58667-0\\_19](https://doi.org/10.1007/978-3-319-58667-0_19)
- Xiao X, Sun J, Yang J (2021) Operation and maintenance(o & m) for data center: an intelligent anomaly detection approach. *Computer Communications*, 178:141–152. ISSN 0140-3664. <https://doi.org/10.1016/j.comcom.2021.06.030>. <https://www.sciencedirect.com/science/article/pii/S0140366421002541>. Accessed 2 June 2022
- Zhang X, Meng F, Chen P, Xu J (2016) Taskinsight: A fine-grained performance anomaly detection and problem locating system. In: 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), pages 917–920. <https://doi.org/10.1109/CLOUD.2016.0136>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.