



# A visual tool for monitoring and detecting anomalies in robot performance

Nuño Basurto<sup>1</sup> · Carlos Cambra<sup>1</sup> · Álvaro Herrero<sup>1</sup>

Received: 13 August 2021 / Accepted: 13 December 2021 / Published online: 24 January 2022  
© The Author(s) 2022

## Abstract

In robotic systems, both software and hardware components are equally important. However, scant attention has been devoted until now in order to detect anomalies/failures affecting the software component of robots while many proposals exist aimed at detecting physical anomalies. To bridge this gap, the present paper focuses on the study of anomalies affecting the software performance of a robot by using a novel visualization tool. Unsupervised visualization methods from the machine learning field are applied in order to upgrade the recently proposed Hybrid Unsupervised Exploratory Plots (HUEPs). Furthermore, Curvilinear Component Analysis and t-distributed stochastic neighbor embedding are added to the original HUEPs formulation and comprehensively compared. Furthermore, all the different combinations of HUEPs are validated in a real-life scenario. Thanks to this intelligent visualization of robot status, interesting conclusions can be obtained to improve anomaly detection in robot performance.

**Keywords** Smart robotics · Component-based robot software · Performance monitoring · Anomaly detection · Machine learning · Unsupervised visualization · Clustering · Exploratory projection pursuit

## 1 Introduction

The European Commission identified smart robotics as an innovation field that would benefit from the development of Key Enabling Technologies (KETs) [1]. Undoubtedly, to successfully deploy autonomous robotics systems, further innovative digital solutions must be conceived and validated in real scenarios. In the past years, plenty of attention has been devoted to deploy such robots, with advanced capabilities not only for autonomous operation but also for self-diagnosis. However, the demands of enhanced systems also lead

to a significant increase in the complexity of them, while reliability and robustness are also required. While operating in real-world environments, robots fail and analysing these failures is a keystone in the road to complete autonomy. Both the hardware and software components of robots suffer from failures. The former has been widely researched [2, 3] while little effort has been devoted to the latter so far [4]. In order to bridge this gap, a new tool is introduced in this novel work, based on previously proposed Hybrid Unsupervised Exploratory Plots (HUEPs) [5], to successfully monitor the performance of software components within a robot while supporting the detection of software anomalies.

Detection of anomalies can be defined as, once expected behavior is known, finding certain patterns in the data that do not conform to it [6]. In order to do that, many proposals based on supervised Machine Learning (ML) are being successfully applied to anomaly detection in a wide variety of industrial problems, ranging from service elevator [7] to solar panels [8] or Unmanned Aerial Vehicles [9], among others. From a complementary perspective, present work investigates the use of unsupervised ML techniques to exploratory study performance in order to know more about anomaly datasets. As a result of the obtained knowledge, the application of some other supervised ML methods may be

---

Carlos Cambra and Álvaro Herrero have contributed equally to this work.

---

✉ Nuño Basurto  
nbasurto@ubu.es

Carlos Cambra  
ccbbaseca@ubu.es

Álvaro Herrero  
ahcosio@ubu.es

<sup>1</sup> Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Ingeniería Informática, Escuela Politécnica Superior, University of Burgos, Av. Cantabria s/n, Burgos 09001, Spain

enhanced, being out of the scope of the present proposal. According to recent and comprehensive methodologies for cost-efficiently monitoring the Industry 4.0, “exploratory data analysis must be first done, comprising techniques such as descriptive statistics, dimensionality reduction and clustering, among others” [10]. Coherently, a projection extension of HUEPs is proposed as a novel combination of some of the above mentioned techniques; more precisely Curvilinear Component Analysis (CCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) are incorporated to HUEPs due to their non-linearity. Furthermore, such visualization techniques are applied for the first time to the monitoring of robot performance, up to the authors knowledge.

The rest of this paper is organized as follows: related work is discussed in Sect. 2, the extended HUEPs formulation is described in Sect. 3 while Sect. 4 presents the the real-life case and the associated data that are analysed. The results obtained by the proposed solution are presented in Sect. 5 and the main conclusions are discussed in Sect. 6.

## 2 Related work

The development that robotic systems are currently undergoing is huge, with different performance improvements, and several challenges related to people [11]. Likewise, this evolution is being carried out together with the robot software, where different lines of research are still open [12]. One of such unsolved problems is the detection of anomalies affecting the software of robots.

To contribute in this field, the first open dataset [13] was released and is presented in [14]. This dataset (described in the Sect. 4) contains data about performance indicators of a robotic system, comprising both “normal” and anomalous states. It is analyzed in the present work and has been previously studied from a supervised ML perspective. Support Vector Machines (SVMs) were applied by the dataset authors [15] in order to automatically detect the software anomalies. In the doctoral dissertation [16] associated to this dataset, that compiles results from all the previous publications by these authors, two different approaches are described. On the one hand, the previously-mentioned application of supervised ML for detecting anomalies and consequently activating automatic reactions in execution time, based on the use of component resources. On the other hand, a set of tools has been developed to understand and systematize resource control under the frame of the robotic system itself. Under the same perspective, supervised ML has been previously applied [17] by the present paper authors, trying to improve the classification results by dealing with missing values and applying data balancing techniques.

Differentiating from previous work, unsupervised methods are not proposed as an initial step for subsequent

supervised methods aimed at classifying data (normal/anomalous). The idea behind the present proposal is to apply unsupervised ML in isolation (visualization and clustering techniques are applied to the same data and then combined) to support the monitoring and study of robot performance status. Thanks to this improved visualization of high-dimensional data, deep knowledge can be gained about the structure of anomaly datasets.

In keeping with this idea, authors [18] of the present paper have previously explored the application of dimensionality-reduction techniques for the visualization of similar data. However, in this previous work exploratory techniques were not combined with any clustering method, as it is proposed in the present work.

As previously stated, scant attention has been devoted so far to the detection of anomalies affecting the software of robots. Furthermore, there is no previous work studying the application of unsupervised ML methods for visualising and monitoring such anomalies. The most similar previous work that is worth mentioning is [19]. Authors of this paper propose the use of unsupervised learning for the detection of anomalies in robots using a Long Short-Term Memory (LSTM) joint auto-encoder. Satisfactory results have been obtained when applying it to data coming from the sensors integrated in the robot. On the other hand, in [20] the use of sliding-window convolutional variational autoencoder is proposed to perform real-time anomaly detection in an industrial robot. It is applied as a tool that helps in the maintenance of robotics systems, by dealing with multivariate time series problems. A more recent research [21] proposes the application of Bayesian networks. As they allow to interrelate different heterogeneous data sources, are applied to perceive anomalies in mobile robots and to recover from them.

## 3 Novel visualization techniques for HUEPs

Hybrid Unsupervised Exploratory Plots (HUEPs) [5] have been recently proposed as a new visualization tool to combine the outputs of Exploratory Projection Pursuit (EPP) and Clustering methods in a novel and informative way. To address the well-know “curse of dimensionality” challenge and advancing in descriptive data analysis, both kind of methods are independently applied and their outputs combined in a new way. More precisely, 3 EPP methods (mainly based on Artificial Neural Networks) were proposed, namely Principal Component Analysis (PCA), Maximum Likelihood Hebbian Learning (MLHL), and Cooperative MLHL (CMLHL).

The output obtained by these 3 EPP methods is a bidimensional vector  $(y_1^{EPP}, y_2^{EPP})$  for each data in the original input space  $(x_n)$ . On the other hand, the output of the

clustering method is a scalar ( $y_c$ ). These three items are then combined in a 3D vector ( $y_1, y_2, y_3$ ) that is visualized in a scatter plot, together with some information for the glyph metaphor ( $x_i$ ).

Going one step further, a visualization extension is proposed in the present paper, to improve the original HUEP formulation. Initially, HUEPs were conceived as a new way of intuitively visualizing data by applying one partitional (k-means) or one hierarchical (agglomerative) clustering method together with one EPP method. Advancing this initial proposal, complementary visualization methods are proposed in the present work, as shown in Fig. 1 (original formulation in grey and present visualization extension in blue).

Visualization is a challenging task, specially when analyzing high-dimensional and real-life data as the one in present research. To address this issue, the proposed visualization extension comprises the following methods, that are applied and validated under the frame of HUEPs for the first time: t-SNE and CCA. These methods are now considered for this extension as they generate non-linear visualizations of data. They are briefly introduced in the following subsections and further details about them can be found in the given references.

### 3.1 Curvilinear Component Analysis

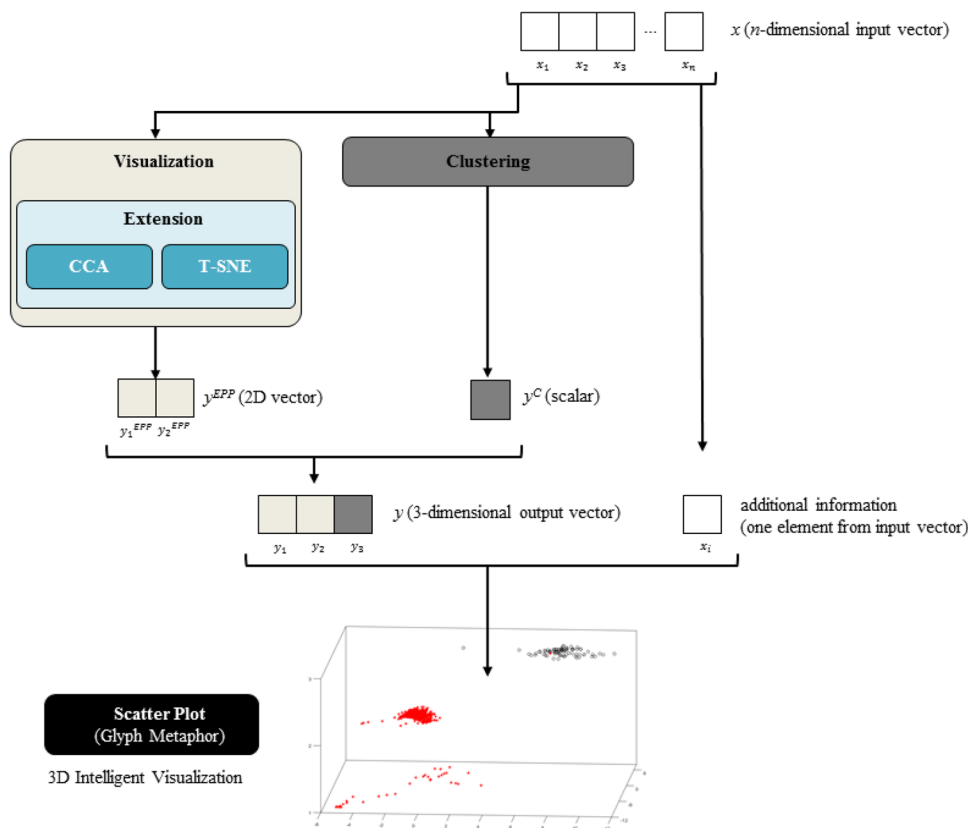
Curvilinear Component Analysis (CCA) was proposed by Demartines and Herault [22] as a self-organizing neural network to find a representation of multidimensional datasets by reducing their dimensionality. To do so, an H-dimensional dataset is projected in an R-dimensional map. CCA is similar to other nonlinear mapping projection techniques such as Sammon’s nonlinear mapping [23]. But it mainly differs from these other methods in the use of a new cost function and a greater courage when it comes to represent. It is proposed as an improvement of Sammon’s mapping because the latter cannot reproduce all distances. CCA does this by reproducing first the nearest and then the farthest distances. The error function used by CCA is the following one.

$$E_{CCA} = \sum_{i,j=1}^N (d_{ij}^n - (d_{ij}^p)^2 F_{\lambda}(d_{ij}^p)) \tag{1}$$

### 3.2 t-distributed stochastic neighbor embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) was proposed by Van der Maaten and Hinton [24]. It is a variation of Stochastic Neighbor Embedding [25] producing a better visualization of high-dimensional data while reducing

**Fig. 1** HUEP novel formulation comprising the proposed Visualization Extension. Adapted from [5]



the concentration of points at the same place in the map. This visualization method is able to take into account the concentration of global structures, revealing at the same time the presence of different clusters. One of the tricks used by t-SNE is the “early compression”, that is aimed at keeping the points together at the beginning of the compression, as shorter distance between the points would ease for the clusters to differentiate between them. It tries to reduce the divergence between two distributions, the first of which measures the similarities between the input objects in pairs, while the second measures the similarities in a low dimensionality in pairs of the points. It differs from PCA in that this visualization technique is non-linear, as CCA. Its operation can be summarized as follows; first it creates a probability distribution between the different points and their neighbors. Then, t-SNE creates a visualization with a lower dimensionality as a result of that distribution.

#### 4 Analysing performance anomalies in robots

As previously mentioned, the present paper proposes HUEPs to study performance anomalies in the middleware of a component-based robot. This kind of anomalies affecting robot software can be divided into several categories. Some of them may not cause the robot stop working, but lead to performance deterioration and cause delays.

To address this problem, an open dataset [13] is analysed in the present paper. This dataset was gathered by different researchers at the University of Bielefeld (Germany), for which different metrics of a robot were recorded during a test run on several times. This robot is based on a model developed by Omron Adept Technologies (former MobileRobots).

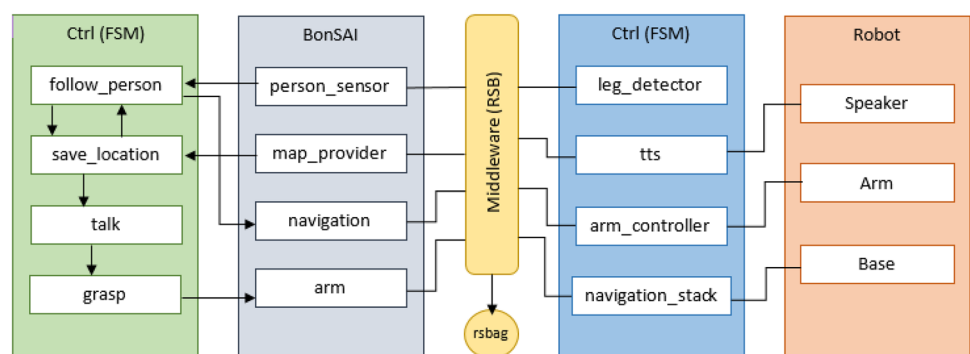
The robotic system under study has different components added to the base, which is why it is called a component-based robot. These components may have been developed by different companies and integrated, forming the robotic system. Some of the different components are: sensors and cameras capable of performing object recognition, voice

recognition or person tracking, actuators responsible for navigation or translation of sentences from text to voice, and an arm that performs human-like movements, along with a grip capable of lifting objects. All the information obtained from the different sensors together with the actions to be performed by the actuators are integrated thanks to the RSB Middleware [26] that provides communication among the components. The coordination of the system is carried out by the BonSAI Framework allowing the system to behave as a finite state machine. A graphical summary of the robot structure, comprising these elements, can be seen in Fig. 2.

The authors of the dataset provide documents in which they indicate what are the moments in which the anomalies are induced and on which components. As a result, the data instances are correctly and completely labelled. They also comment on the time instants in which unplanned anomalies occur in the system.

There are several anomalies that have been induced and affect one or several components. *ArmServerAlgo*, *legDetectorSkippable*, *objectBuilderSkippable*, *clafuSleep*, *pocketSphinxLeak*, *btlAngleAlgo*, *bonsaiParticipantLeak*, *bonsaiTalkTimeout*, and *facerecSkippable* just affect one component. *ClockShift* affects four components and *Spread-Latency* affects all components. In the present research two anomalies have been chosen (*armSeverAlgo* and *legDetectorSkippable*), according to the investigations of the creators of the dataset [15]. The first one (*armSeverAlgo*) got the worst results when evaluating the dataset by means of SVM, while for the second one (*legDetectorSkippable*), very accurate classification values were obtained. *ArmServerAlgo* (hereinafter referred as A1) affects the *armcontrol* component, that supervises the movement of the mechanical arm as well as the movement of the gripper. The anomaly causes the arm to make a series of extra moves to carry out an action, which penalises the performance but does not prevent the task from being carried out successfully. The anomaly *legDetectorSkippable* (hereinafter referred as A2) affects the *legdetector* component, that is in charge of detecting people’s legs for recognizing the presence of human beings. Similarly to what happened with the *armServerAlgo* anomaly, several unnecessary actions were performed, thus

**Fig. 2** Overview of the robot system architecture. Adapted from [14]



affecting the performance but not preventing the task to be completed.

When information was gathered from the robot, it was performing different tasks in a certain order. The different tasks involved the components discussed above. Experiments were repeated a total of 71 times; that is the reason why the dataset comprises 71 trials. Not all of them are used in the present experimentation because some of them have undesired anomalies, whose origin is not known.

For each one of these 71 trials, all the information gathered from each one of the components is available. The dataset contains the information about when and which anomalies were induced together with data from several data sources. The first one of these data sources are the *counters*, that export the performance counters of each one of the robot components on a regular basis (less than a second). The information from this source includes data such as the number of active processor threads or the amount of information sent and received.

The second source are the *events*, those that were sent between the system bus and the user code. This information tells which are the relevant events in the component, including the sending and receiving address or the information size. For every second of execution of the trial there can be multiple instances that appear in this dataset, being approximately 60 times larger than those observed in counters. This is why the third of the data sources (*features*) is required. It combines events with performance counters in the counters time instants. The events presented in features are the last ones received as well as the averages for different temporal moments. This causes the features to be the dataset with the highest dimensionality. The authors published more detailed information on the dataset as well as the source files in [13]. From these three sources of data, features and counters are analyzed in the present work as in previous experiments they were identified as those that implied best results [27].

Due to length limitation, results from all the anomalies and trials can not be included in the present paper. In order to evaluate the proposed HUEPs extension under different circumstances, experiments were conducted including, for each one of the two analysed anomalies (previously discussed) two variants: only one trial and all the trials. The motivation for that is to check whether the HUEP visualizations are equally useful depending on the amount of trials (and data instances) to be depicted. Hence, trials for these 2 anomalies had to be selected. Among all the trials containing examples of the A1 and A2 anomalies, only one was chosen for each case. Firstly, the smallest trial (lowest number of data instances) was selected. Secondly, for those trials with the same number of occurrences, the one with lowest balance ratio between the normal and the anomaly class (the most unbalanced) was selected. As a result, the proposed HUEPs extension is validated both on small and unbalanced

datasets. One-trial datasets have been selected in the present paper to check the ability of extended HUEPs to visually depict the structure of small datasets. On the contrary, all-trial datasets have been selected as they contain a big amount of data.

As previously studied [27], the trials within this dataset contain missing values (MV), that must be pre-processed before applying most ML methods. In the present paper, MV were removed according to a 0% rate. Accordingly, all the features containing any MV were removed from the dataset. As a result, datasets to be analysed in the present paper have a different dimensionality.

Figures about some characteristics of the employed datasets (one and all trials) are shown in Table 1.

In addition to the visualization extension that is above described, the application of HUEPs to the present problem unveils another novelty. The additional information that is provided to the HUEPs consist now on the class information for each data. Consequently, the glyph metaphor is used to depict the data according to the class (normal versus anomalous states) it belongs to. This way, hybridization in HUEPs is maximized as on the top of the combination of unsupervised methods, class information is used to better understand the patterns associated to anomalous states of the robot software. Thanks to it, the structure of the dataset and the results of other ML techniques can be easily understood.

## 5 Experiments and obtained results

In this section, the advanced visualizations obtained by applying HUEPs (original and extended formulations) are shown. For the visualization and clustering methods, parameters have been tuned according to previous recommendations [28, 29]. As previously mentioned, the glyph metaphor is applied by using the class (normal versus anomalous) information, once the data are located in the 3D output space. Accordingly, normal data are depicted as red stars while anomalous data are depicted as black circles. On the other hand, figures in this section show 3D visualizations;  $y1$ ,  $y2$  and  $y3$ , (discussed in Sect. 3 and observable in

**Table 1** Characteristics of the datasets analyzed in present paper

| Dataset     | Trials | Dimensionality | Normal data  | Anomalous data | Total data |
|-------------|--------|----------------|--------------|----------------|------------|
| A1Trial41   | 1      | 87             | 500 (86.2%)  | 80 (13.8%)     | 580        |
| A2Trial36   | 1      | 23             | 638 (88.9%)  | 80 (11.1%)     | 718        |
| A1AllTrials | 10     | 42             | 5773 (84.2%) | 1032 (15.2%)   | 6805       |
| A2AllTrials | 12     | 21             | 6628 (14.5%) | 1128 (85.5%)   | 7756       |

image 1), are represented in the following figures. The first two are 2D coordinates of each data sample, while  $y_3$  is the assigned cluster.

As there are many different visualizations obtained by combining all the EPP (PCA, MLHL, CMLHL, CCA, and t-SNE) and clustering (k-means and Agglomerative) methods for each one of the datasets and parameter values, only some of them can be included in the present paper. Visualizations have been selected according to different comparison criterion and are shown in the following subsections. Firstly, results for the A1 anomaly are shown in Sect. 5.1 while results for the A2 anomaly are shown in Sect. 5.2, comprising visualizations of both one and all-trial datasets each.

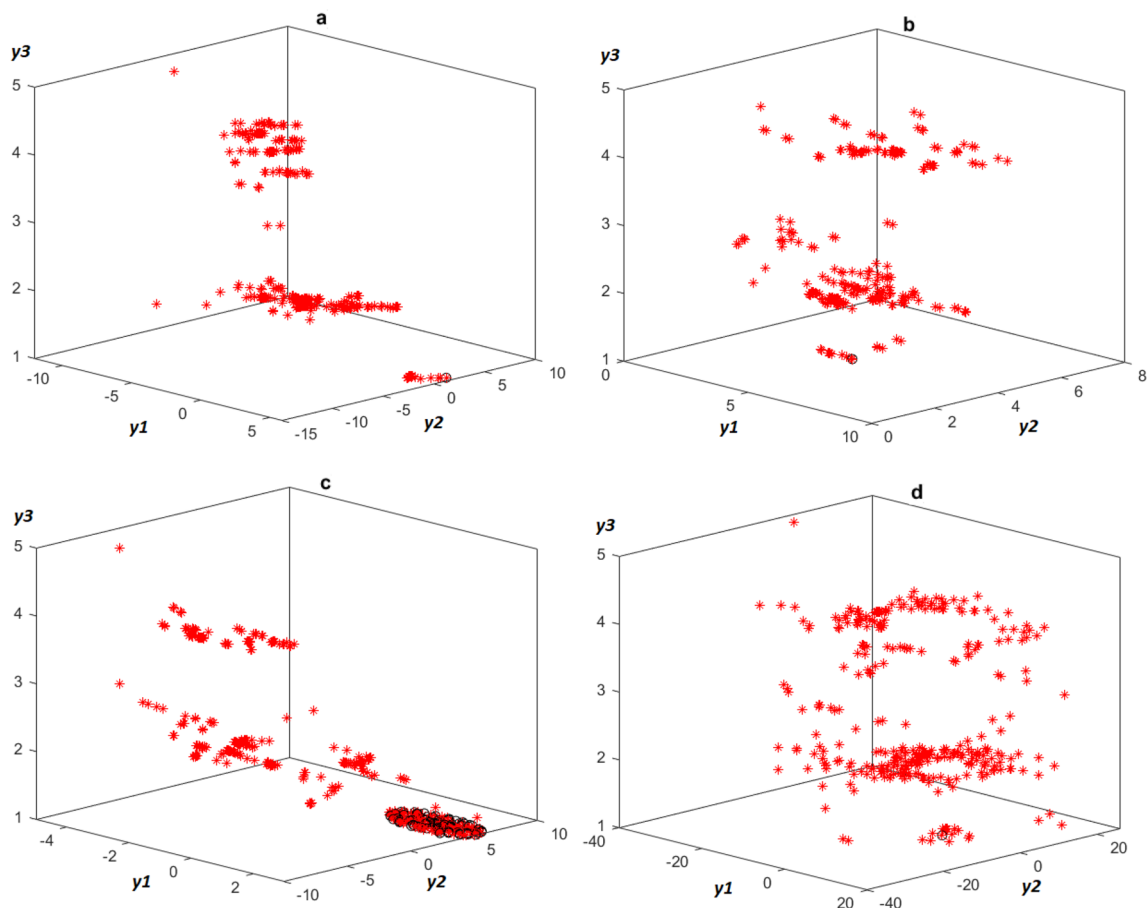
### 5.1 A1 anomaly results

In order to compare the visualizations for the A1 anomaly (A1Trial41 and A1AllTrials datasets) discussed above, only some of them are shown in the present subsection for each one of the datasets. These HUEPs consist of the combination of the same clustering method (Agglomerative) and cluster

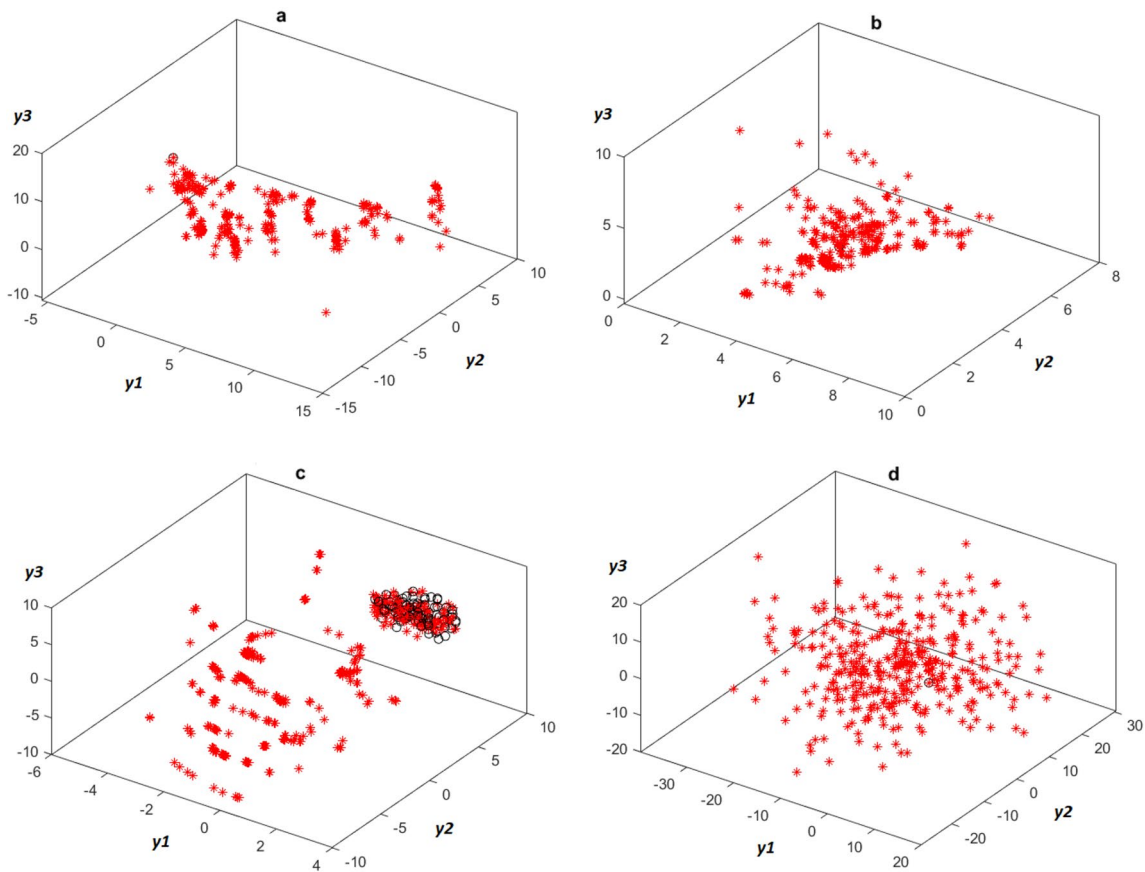
number (5), together with 4 different visualization methods, namely PCA, MLHL, t-SNE, and CCA.

HUEPs visualizations of the one-trial (A1Trial41) dataset are shown in Fig. 3. Before analyzing these visualizations, it is worth mentioning that in this dataset all the anomalous instances (80) are the same. That is, for all the features in the dataset, all of them took the same value at the different times when data were captured. This phenomenon can be observed in Fig. 3 as for most of the visualization methods (PCA, MLHL, and CCA) anomalous instances overlap and are depicted as a single data, containing not only the anomalies but also normal data. However, it is not the same in the case of the t-SNE visualization, where anomalies do not overlap. Additionally, all these anomalous instances are grouped together. They are depicted as a group in the right-bottom corner of Fig. 3c). This way, it is easily seen that the group comprises both anomalous and normal data.

For comparison purposes, 3D visualizations obtained by the same techniques are shown in Fig. 4. It means that the three components of each data sample are only calculated by the visualization technique. That it, no clustering results are



**Fig. 3** HUEP visualizations for A1Trial41 dataset. Agglomerative with 5 clusters + projection techniques. **a** PCA, **b** MLHL, **c** t-SNE, and **d** CCA



**Fig. 4** 3D visualizations for A1Trial41 dataset: **a** PCA, **b** MLHL, **c** t-SNE, and **d** CCA

combined but the glyph metaphor is applied (differentiating between normal and anomalous data).

The 3D visualizations are quite similar to those obtained by HUEPs. Anomalous instances overlap except for the t-SNE visualization. Furthermore, data are not clearly grouped in the case of the 3D PCA and t-SNE visualizations. It is worth noting that in the case of CCA, the 3D visualization is almost useless as there is only one group containing all the data and the structure of the dataset is not revealed at all. In the 3D visualizations, data groups are less clearly defined than in the corresponding HUEPs. This way, it is proved one of the contributions of HUEPs, that is reinforced thanks to the novel visualization techniques (t-SNE in this dataset): the representation capability of the visualization techniques is improved by adding the clustering information.

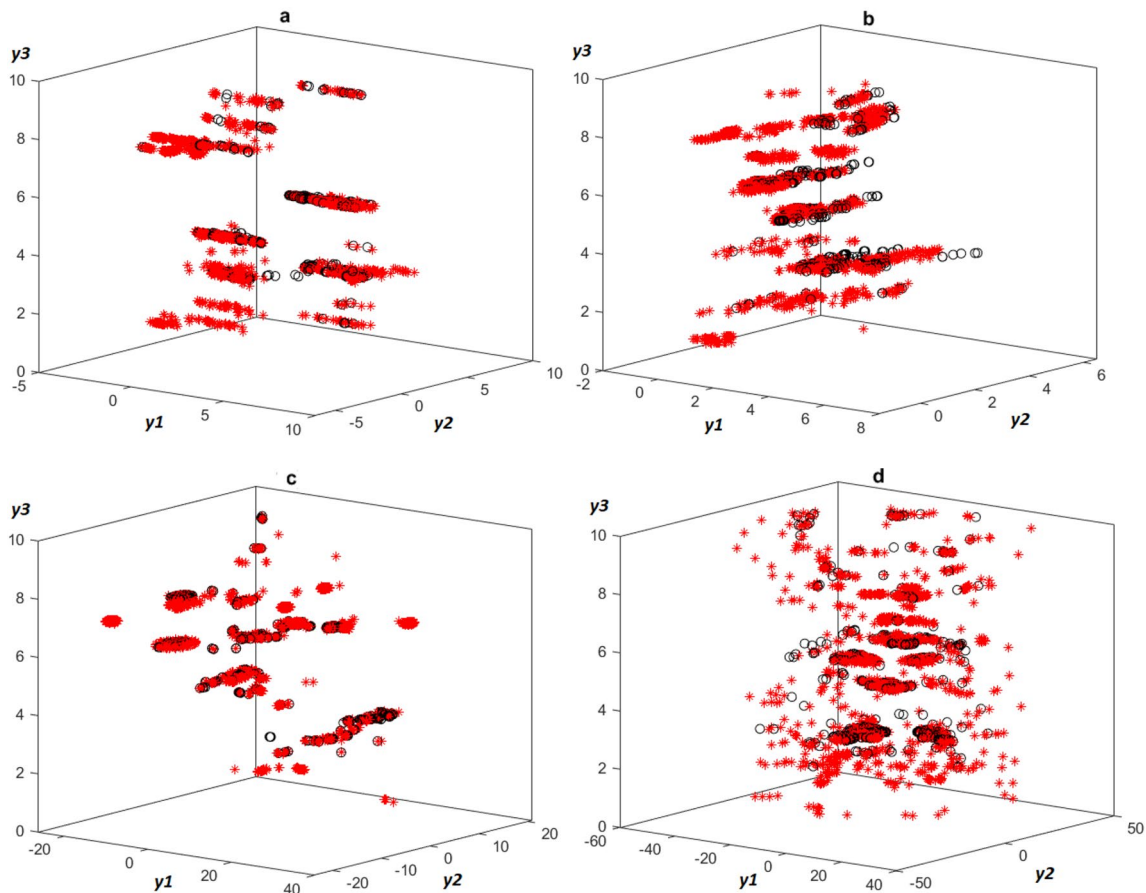
Similarly to the one-trial dataset, some visualizations of HUEPs are shown for the all-trial (A1AllTrials) dataset in Fig. 5. For a fair comparison with the one-trial visualizations, the same clustering method (Agglomerative) is applied and together with 4 different visualization methods, namely PCA, CMLHL, t-SNE, and CCA. In this case, a higher number of clusters (10) is set in order to check the ability of HUEPs to depict such results.

It can be seen in Fig. 5 that a high number of groups in the clustering causes poor visualizations where groups can be hardly identified and normal/anomalous data are not clearly separated. The worst visualization is obtained by CCA as there is no data grouping at all.

### 5.2 A2 anomaly results

As it has been previously explained, HUEPs are also applied to the A2 anomaly (A2Trial36 and A2AllTrials) datasets. Firstly, HUEP visualizations for the one-trial dataset are shown. Each one of these figures shows the HUEPs obtained by combining the output of the new visualization techniques proposed in this research (CCA and t-SNE), together with the clustering techniques previously mentioned: k-means and Agglomerative. This way, a direct comparison between the two novel techniques is provided for this dataset. Their visualization performance is also validated in conjunction with the clustering techniques by using a different number of clusters.

In order to compare the performance of HUEPs, executions have been performed with different number of clusters: 3 and 9. 3 was chosen as a relatively small number of



**Fig. 5** HUEP visualizations for A1AllTrials dataset. Agglomerative with 10 clusters + projection techniques. **a** PCA, **b** CMLHL, **c** t-SNE, and **d** CCA

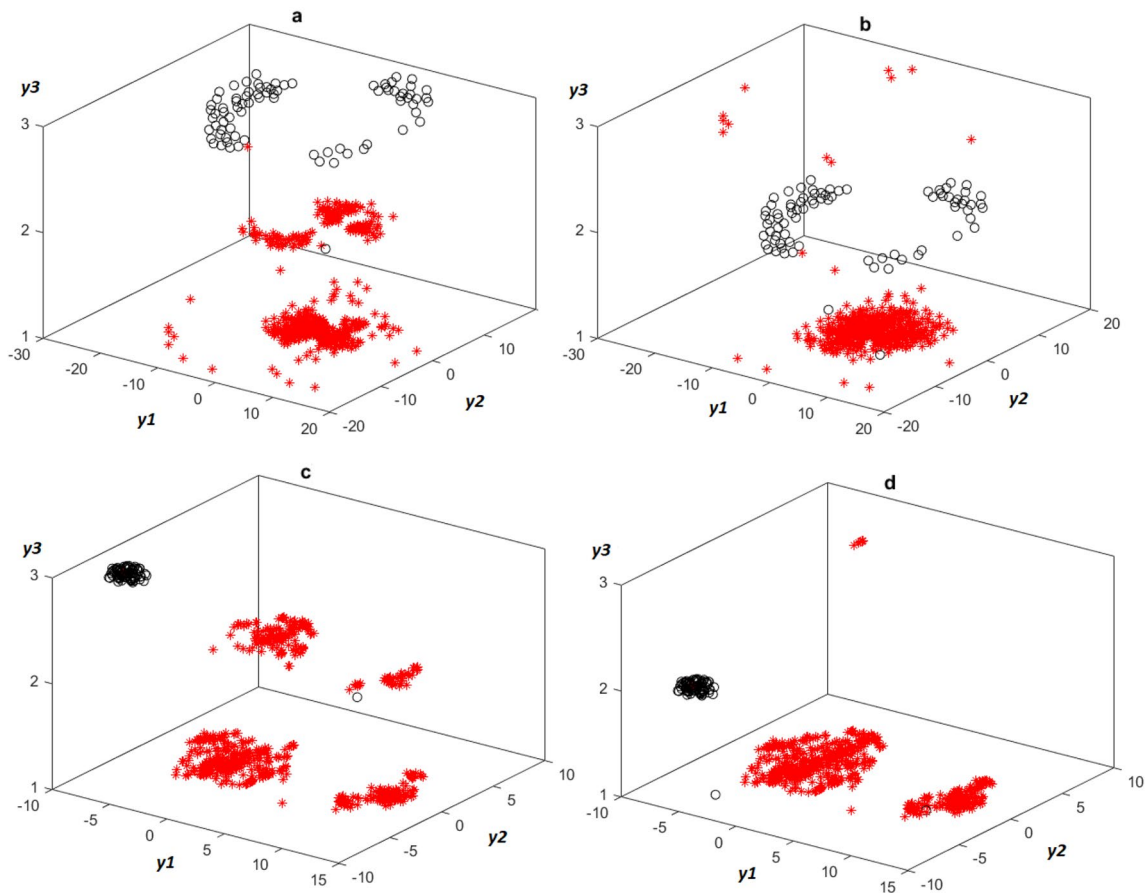
clusters while 9 was chosen as a large one, although a larger value (10) has been used to obtain the visualization of the all-trials dataset. Thanks to it, the visualization ability of extended HUEPs is validated for clustering results comprising both low and high numbers of clusters.

HUEP visualizations of the one-trial (A2Trial36) dataset with 3 clusters are shown in Fig. 6, where the dataset structure is clearly depicted in all cases. The group containing the anomalous data is more sparsely depicted by CCA while t-SNE tends to concentrate the data in a group of small size. However, it is worth highlighting that in the case of t-SNE (c and d), the anomalous data is almost completely isolated from the normal one. Similarly, the clustering techniques are not able to split all normal and anomalous data: in the case of the Agglomerative clustering technique (b and d), there are two anomalous instances grouped with the normal ones. In the case of CCA (a and

b) these instances are closer to the rest of the anomalous instances. Thanks to this visualization, these anomalous data have been further studied. In a more detailed analysis, it has been discovered that they are data instances associated to those moments of time in which the induction of anomalies begins and ends. Thanks to the HUEP visualization, such situations can be identified, leading to a subsequent improvement on the anomaly detection. All in all, it can be said that HUEPs extended by the new visualization techniques are able to depict this dataset in a way that its structure is clearly revealed. As a result, anomalous data can be clearly identified and split from normal data.

In the HUEP obtained with 9 clusters (Fig. 7), it can be seen how this visualization is less clear than that with a smaller number of clusters (Fig. 6). As in this previous figure, it can be observed that t-SNE (c and d) more clearly reveals the structure of the dataset as groups are





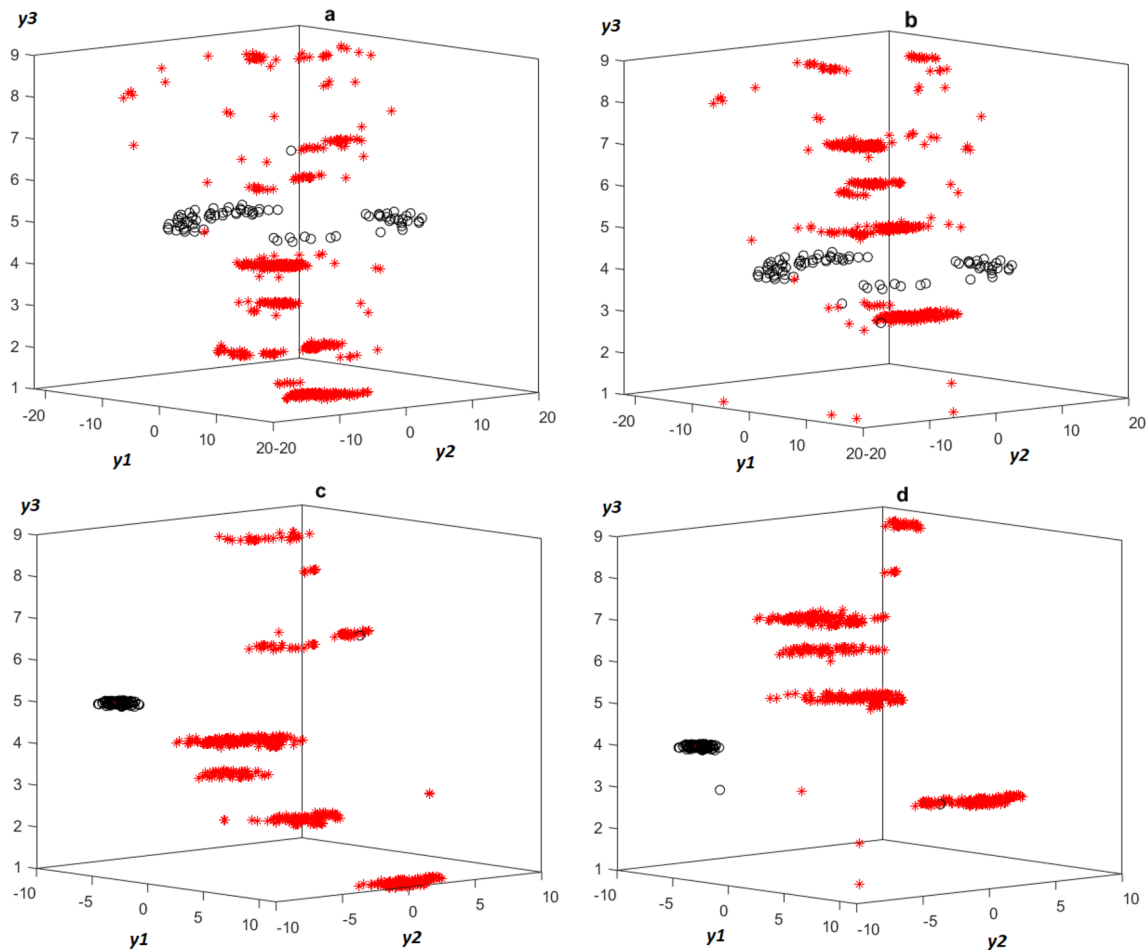
**Fig. 6** HUEP visualizations for A2Trial36 dataset. K-means and Agglomerative with 3 clusters + projection techniques. **a** CCA + k-means, **b** CCA + Agglomerative, **c** t-SNE + k-means, and **d** t-SNE + Agglomerative

shown in a compact way. CCA (a and b) does not generate visualizations as compact as those obtained for a reduced number (3) of clusters. When visualizing same data with a higher number of clusters there is a more confusing separation between data classes. One common characteristic of visualizations in Fig. 7 is that the normal class is split in a greater number of clusters while the anomalous data are kept in only one group. This group is much more concentrated in the case of t-SNE. By taking this into account, it can be concluded that increasing the number of clusters does not always mean a better visualization.

Finally, HUEPs are shown for all the trials of A2 anomaly (A2AllTrials dataset) in Fig. 8. These visualizations have been obtained by combining the Agglomerative clustering with the PCA, MLHL, t-SNE, and CCA visualization

methods. In this case, a reduced number of clusters (4) has been used.

In this case, the visualizations are very similar to those obtained for the one-trial dataset of the same anomaly (A2Trial36). However, Fig. 8 shows the visualizations obtained by two other techniques: MLHL (b) and PCA (a). Although they have achieved a good separation of clusters, in the case of MLHL the instances of the different classes overlap, that does not happen with any of the other visualization techniques. As for PCA, it achieves a very good separation although slightly worse than that obtained by t-SNE (c). Finally, CCA (d) generates a good separation of clusters, similar to the one previously observed in Figs. 6 and 7 but slightly worse. All in all, it can be said that the separation by clusters is quite good, as it happened with the dataset of



**Fig. 7** HUEP visualizations for A2Trial36 dataset. K-means and agglomerative with 9 clusters + projection techniques. **a** CCA + k-means, **b** CCA + Agglomerative, **c** t-SNE + k-means, and **d** t-SNE + Agglomerative

a single trial (A2Trial36). It can be said that the extended HUEPs can be successfully applied to a large dataset (A2A1-Trials dataset is the largest one in the present study).

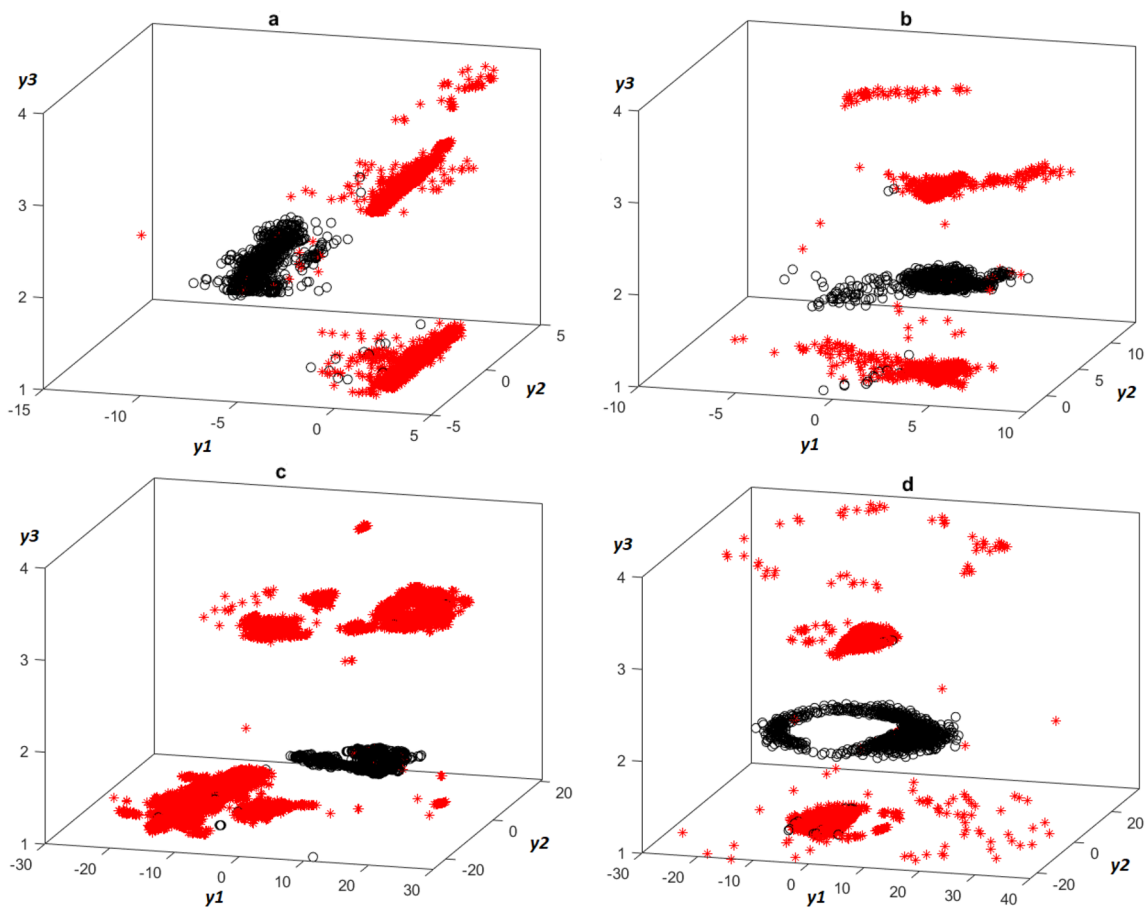
As in the case of the previous anomaly, the 3D visualizations of A2 by the different techniques are shown in Fig. 9. They are slightly different from the HUEPs, in a different way to what happened with the A1 dataset (see Sect. 5.1). The 3D PCA (a) visualization is quite similar to what can be observed in the corresponding HUEP (Fig. 8a), where the two classes are separated but closely located in the output space. Visualization is also quite similar in the case of t-SNE (c), showing a good separation of classes. However, groups containing normal data are more clearly separated in the case of the HUEP visualization. The biggest differences can be observed in the cases of MLHL and CCA. The 3D

visualizations are not revealing; groups can not be identified and classes (normal/anomalous) are mixed up.

It can be concluded that HUEPs provide with visualizations where the structure of datasets can be observed in a more clear way than in the case of visualizations obtained by other methods. By analysing the results of the present study, classes have been more clearly grouped in the case of the A2 anomaly than in the case of the A1 one.

## 6 Conclusions and future work

To monitor the performance of the software components within a robot, a visualization extension of HUEPs is proposed and validated in the present paper. More precisely,



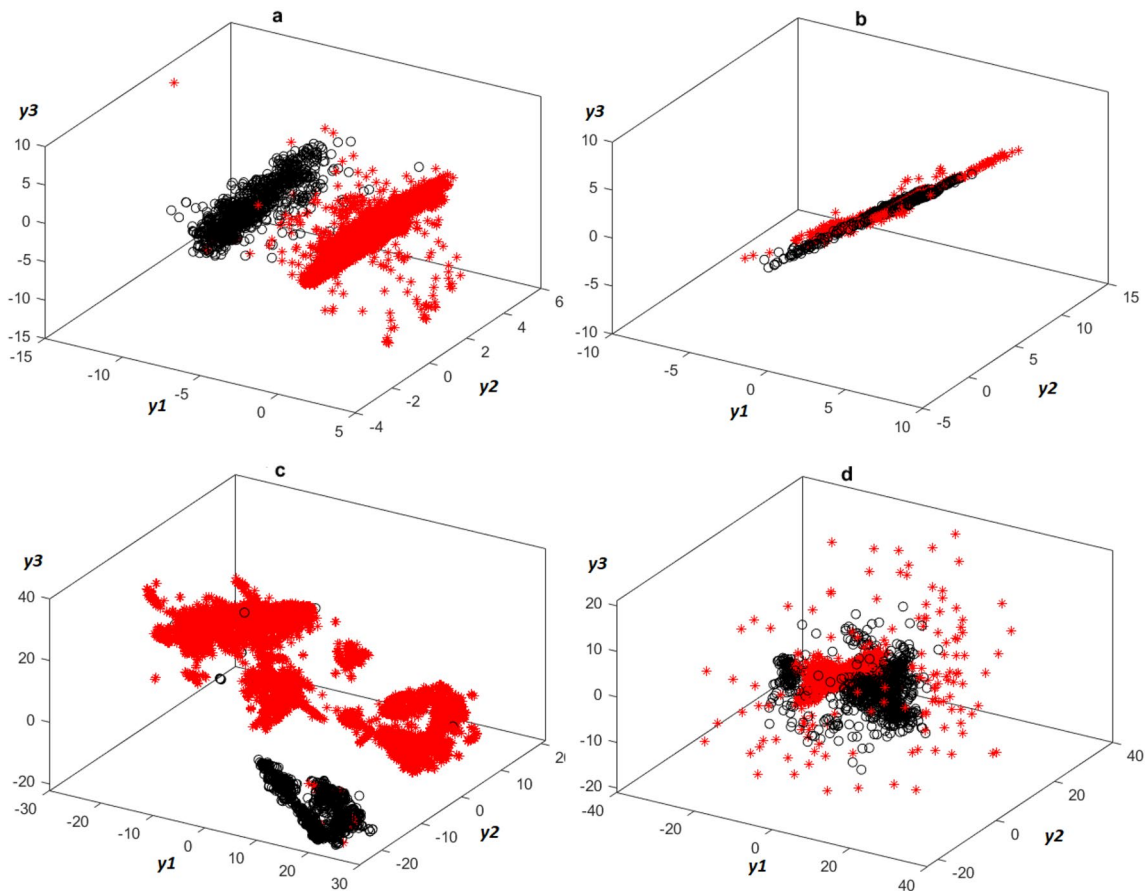
**Fig. 8** HUEP visualizations for A2AllTrials dataset. Agglomerative with 4 clusters + visualization techniques. **a** PCA, **b** MLHL, **c** t-SNE, and **d** CCA

CCA and t-SNE are introduced in the HUEPs formulation to benchmark such methods against the originally proposed ones (PCA, MLHL, and CMLHL). A comprehensive experimental study has been carried out to validate the proposed extension, comprising experiments on all the different methods, parameters and datasets. As a result, the proposed visualization methods have been applied to the different (small/large and balanced/unbalanced) datasets when applying different visualization techniques as well as clustering ones. Furthermore, results are compared with a varying number of clusters.

From a general perspective, the experimental results show that HUEPs generated by the new visualization methods can be considered more useful than those obtained by

the original methods. It is worth noting that among all the methods that are applied for the first time, the t-SNE outperforms the other one, contributing to more informative and intuitive visualizations of normal/anomalous states. This is mainly given to the ability of such method to isolate data instances from one class. It must be said that, in general terms, in order to select the most appropriate visualizations technique, as well as the optimal number of clusters, a benchmark study must be conducted on each dataset. There is not a visualization technique that clearly outperforms the other one in all cases.

Regarding the dataset size, it can be said that HUEP visualizations are better for smaller datasets (one-trial). In the visualizations of these datasets data are more clearly



**Fig. 9** 3D visualizations for A1Trial41 dataset. **a** PCA, **b** MLHL, **c** t-SNE, and **d** CCA

grouped and projections are more sparse. In an opposite way, datasets with a larger size (all-trials) are visualized in a less informative way, although good results have also been obtained for such datasets.

As a follow-up of this research line, authors aim at testing new clustering methods to be applied under the frame of HUEPs. Similarly, new ways of combining different sources of information in the same display could also be investigated in order to improve HUEP visualizations.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not

permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Commission E (2014) Study on cross-cutting KETs (Ro-cKETs) . [https://ec.europa.eu/growth/industry/policy/key-enabling-technologies/eu-actions/ro-ckets\\_en](https://ec.europa.eu/growth/industry/policy/key-enabling-technologies/eu-actions/ro-ckets_en)
2. Khaldi B, Harrou F, Cherif F, Sun Y (2017) Monitoring a robot swarm using a data-driven fault detection approach. *Robot Autom Syst* 97:193–203. <https://doi.org/10.1016/j.robot.2017.06.002>
3. Park D, Kim H, Kemp CC (2019) Multimodal anomaly detection for assistive robots. *Autonom Robots* 43(3):611–629. <https://doi.org/10.1007/s10514-018-9733-6>
4. Khalastchi E, Kalech M (2018) On fault detection and diagnosis in robotic systems. *ACM Comput Surv* 51(1):1–24. <https://doi.org/10.1145/3146389>
5. Herrero A, Jimenez A, Bayraktar S (2019) Hybrid unsupervised exploratory plots: a case study of analysing foreign direct investment. *Complexity*. <https://doi.org/10.1155/2019/6271017>
6. Xu X, Liu H, Yao M (2019) Recent progress of anomaly detection. *Complexity*. <https://doi.org/10.1155/2019/2686378>

7. Canizo M, Triguero I, Conde A, Onieva E (2019) Multi-head cnn-rnn for multi-time series anomaly detection: an industrial case study. *Neurocomputing* 363:246–260. <https://doi.org/10.1016/j.neucom.2019.07.034>
8. Murtada WA, Omran EA (2019) Robust anomaly identification algorithm for noisy signals: spacecraft solar panels model. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-019-04407-2>
9. Khalastchi E, Kalech M (2018) A sensor-based approach for fault detection and diagnosis for robotic systems. *Autonom Robots* 42(6):1231–1248. <https://doi.org/10.1007/s10514-017-9688-z>
10. Para J, Del Ser J, Nebro AJ, Zurutuza U, Herrera F (2019) Analyze, sense, preprocess, predict, implement, and deploy (asp-id): an incremental methodology based on data analytics for cost-efficiently monitoring the industry 4.0. *Eng Appl Artif Intell* 82:30–43. <https://doi.org/10.1016/j.engappai.2019.03.022>
11. Roldan-Gomez JJ, de Leon J, Garcia-Aunon P, Barrientos A (2020) A review on multi-robot systems: current challenges for operators and new developments of interfaces. *Revista Iberoamericana de Automática e Informática Industrial* 17:294–305. <https://doi.org/10.4995/riai.2020.13100>
12. Mao X, Huang H, Wang S (2020) Software engineering for autonomous robot: challenges, progresses and opportunities. In: 2020 27th Asia-Pacific software engineering conference (APSEC), pp 100–108. <https://doi.org/10.1109/APSEC51365.2020.00018>
13. Wienke J, Wrede S (2016). A fault detection data set for performance bugs in component-based robotic systems. <https://doi.org/10.4119/unibi/2900911>
14. Wienke J, Meyer zu Borgsen S, Wrede S (2016) A data set for fault detection research on component-based robotic systems. In: Alboul L, Damian D, Aitken JM (eds) *Towards autonomous robotic systems*, vol 9716. Springer, Cham, pp 339–350. [https://doi.org/10.1007/978-3-319-40379-3\\_35](https://doi.org/10.1007/978-3-319-40379-3_35)
15. Wienke J, Wrede S (2016) Autonomous fault detection for performance bugs in component-based robotic systems. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). <https://doi.org/10.1109/IROS.2016.7759507>. IEEE, pp 3291–3297
16. Wienke J (2018) Framework-level resource awareness in robotics and intelligent systems. Phd dissertation, Bielefeld University. <https://doi.org/10.4119/unibi/2932136>
17. Basurto N, Cambra C (2020) Álvaro Herrero: improving the detection of robot anomalies by handling data irregularities. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2020.05.101>
18. Basurto N, Cambra C, Herrero A (2020) Ai-driven visualizations for performance monitoring and anomaly detection in robots. In: 2020 IEEE/ACS 17th international conference on computer systems and applications (AICCSA). IEEE Computer Society, Los Alamitos, CA, USA, pp 1–6. <https://doi.org/10.1109/AICCSA50499.2020.9316513>
19. Wen X, Chen H (2020) Heterogeneous connection and process anomaly detection of industrial robot in intelligent factory. <https://doi.org/10.1142/S0218001420590417>
20. Chen T, Liu X, Xia B, Wang W, Lai Y (2020) Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder. *IEEE Access* 8:47072–47081. <https://doi.org/10.1109/ACCESS.2020.2977892>
21. Castellano-Quero M, Fernández-Madriral JA, García-Cerezo A (2021) Improving Bayesian inference efficiency for sensory anomaly detection and recovery in mobile robots. *Expert Syst Appl* 163:113755. <https://doi.org/10.1016/j.eswa.2020.113755>
22. Demartines P, Hérault J (1997) Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Trans Neural Netw* 8(1):148–154. <https://doi.org/10.1109/72.554199>
23. Sammon JW (1969) A nonlinear mapping for data structure analysis. *IEEE Trans Comput C* 18(5):401–409. <https://doi.org/10.1109/T-C.1969.222678>
24. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. *J Mach Learn Res* 9:2579–2605
25. van der Maaten L, Hinton G (2002) Stochastic neighbor embedding. *Adv Neural Inf Process Syst* 15:833–840
26. Wienke J, Wrede S (2011) A middleware for collaborative research in experimental robotics. In: 2011 IEEE/SICE international symposium on system integration (SII), pp 1183–1190. <https://doi.org/10.1109/SII.2011.6147617>
27. Basurto N, Herrero Á (2020) Data selection to improve anomaly detection in a component-based robot. In: Martínez Álvarez F, Troncoso Lora A, Sáez Muñoz JA, Quintián H, Corchado E (eds) 14th International conference on soft computing models in industrial and environmental applications (SOCO 2019). Springer, Cham, pp 241–250
28. Schubert E, Sander J, Ester M, Kriegel HP, Xu X (2017) Dbscan revisited: why and how you should (still) use dbscan. *ACM Trans Database Syst* 42(3):19–11921. <https://doi.org/10.1145/3068335>
29. Arroyo Á, Herrero Á, Tricio V, Corchado E (2017) Analysis of meteorological conditions in Spain by means of clustering techniques. *J Appl Logic* 24:76–89. <https://doi.org/10.1016/j.jal.2016.11.026>

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