A Hybrid Machine Learning System to Impute and Classify a Component-Based Robot

Nuño Basurto[1][0000−0001−7289−4689], Ángel Arroyo[1][0000−0002−1614−9075], Carlos Cambra[1][0000−0001−5567−9194], and Álvaro Herrero[1][0000−0002−2444−5384]

Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006, Burgos, Spain. {aarroyop, nbasurto, ccbaseca, ahcosio}@ubu.es

Abstract. In the field of cybernetic systems and more specifically in robotics, one of the fundamental objectives is the detection of anomalies in order to minimize loss of time. Following this idea, this paper proposes the implementation of a Hybrid Intelligent System in four steps to impute the missing values, by combining clustering and regression techniques, followed by balancing and classification tasks. This system applies regressions models to each one of the clusters built on the instances of data set. Subsequently, a variety of balancing techniques are applied to improve the classifier's ability to discern whether it is in an error or a normal state. These techniques support to obtain better classification ratios in which a robot is close to error and allow us to bring the behavior back to a normal state. The experimentation is performed using a modern and public data set, which has been extracted from a component-based robotic system, in which different anomalies are induced by software in their components.

Keywords: Hybrid Artificial Intelligence System · Machine Learning · Clustering · Regression · Missing Values · Component-Based Robot.

1 Introduction

In recent years, the production systems have been including more and more robotic systems in production lines to automate processes and improve efficiency in productivity terms. Robotics has been expanding in a variety of ways, such as quality control, assembly or loading and unloading [19]. The robotic systems cover different disciplines like cinematic, mechatronics, electronics, and Artificial Intelligence (AI). Regarding the latter, the European Commission recently reported that 42% of enterprises use at least one technology related to this field [1]. This percentage will increase in the coming years, showing a clear need for companies to adopt these technologies, which are already consolidated in the market. However, everything depends on adapting current systems and hiring qualified personnel to carry out this work.

In the field of robotics, Machine Learning (ML) algorithms can be applied to detect errors in executions, in order to reduce the impact of these errors and return to normal production. To prevent them and reduce downtime, anomaly detection [19] is required, although not enough efforts are devoted to it from the scientific community so far [21].

The difficulty of processing real data that has been generated from sensors to detect anomalies is a challenge due to the presence of Missing Values (MV) [15], that must be overcame [10]. The need to impute values is enormous in order to minimize the loss of information when working with a robotic data set [27]. The relevance of having a complete data set to work with in order to obtain the best possible results is evident. In the case of not being able to impute the MV, it would be necessary to work with a less complete data set in terms of instances or attributes. In order to address this problem, a novel Hybrid Artificial Intelligence System (HAIS) is proposed.

[1] European Commission. European enterprise survey on the use of technologies based on artificial intelligence (July 2020). URL: https://ec.europa.eu/digital-single-market/en/news/european-enterprise-survey-use-technologies-based-artificial-intelligence

The paper is distributed as follows: the previous work is discussed in 2 while the novel HAIS is described in Section 3. The component-based robot dataset is described in Section 4 and the experimental results are presented in Section 5. Finally, Section 6 deals with the conclusions reached from this research and future lines of work to be pursued.

2 State of the art

To solve the MV problem, very different techniques have been applied up to now [27]. These Imputation Methods (IM) are classified as single imputation, where the method fills in one value for each missing one and multiple imputation where multiple values are tried at the same time. In this work, single imputation techniques have been applied to make the resulting imputation data set more easily usable for classification purposes. To achieve the best possible MV imputation a nonlinear regression technique together with an Artificial Neural Network (ANN), more precisely the Radial Basis Function Network (RBFN) taking advantage of their regression capability [16], are applied to fulfill the data set.

Other approaches that have been adopted concerning the presence of MV are carried out by Cerqueira et al. [7], who are committed to the elimination of MV. Likewise, these authors deal with the use of balancing techniques such as Synthetic Minority Oversampling Technique (SMOTE) to achieve a balanced distribution between classes. Following this line of use of balancing techniques, Syafrudin et al. [32] also relies on the use of SMOTE, in this case, the target is to detect possible anomalies in an assembly line. The approach used is to differentiate between two classes (one-class classification), one being normal and the other abnormal. Another recent study that combines balancing techniques with a one-class approach is the work related on Debarshhree et al. [13], where the authors investigated the impact that unbalancing can have on a variety of data sets. To test the effectiveness of the applied balancing techniques they use a variety of classifiers such as Extreme Learning Machines, Naive Bayes, or

Support Vector Machines (SVM). It was possible to improve the trends of the minority class for the majority class employing SVM.

In the case of the anomaly dataset that is analysed in the present paper [37], scant attention has been devoted to it by the research community, mainly due to its novelty. One of the very few papers addressing this topic is . [40]. The auhtors (those of the benchmark dataset itself) developed an error detection model, where they compare their own generated model by relying on a One-Class-SVM classifier, they act on the total set of anomalies, (detailed in Section 4). Their model obtained improved results in the vast majority of components, but not in the one analyzed in the present study. In a sequel paper [35], the authors discuss how to carry out individual performance tests on the different components of a robotic system, analyzing possible changes in the different revisions. Subsequently, Wienke et al. [38] further explored the idea of analyzing resource utilization in the different components, proposing a model that performs tests aimed at detecting regressions in resource utilization. The claims of this new model is to reduce the complexity in the creation of performance tests. Finally, Wienke published his thesis [36] in which he applied the methods seen in his earlier papers and extended them by emphasizing his component-based robotic data set and resource utilization improvement techniques.

The problem associated in this paper deals with the imputation of values through the combined use of regression and clustering techniques. Failure detection has already been discussed previously on this dataset [4]. However, the previous work on this data proposed the elimination of MV [3] rather than their imputation. The problem addressed in this paper have already been dealt with in less depth [1], where the work focused only on cluster regression but without performing classification tests on the data resulting from the imputation. Likewise, the study was conducted only with the k-means clustering technique and in present work it has been extended with additional techniques.

3 Hybrid Intelligent System

The main goal of the hybrid system presented in this research is to establish a new type of MV imputation through the data set clustering, where the columns with MV have been previously extracted. Once the clusters have been obtained, within each one the columns with MV added again, the regression is applied on them to perform the imputation of the values later on. The Figure 1 shows the steps of each of the stages of this hybrid System.

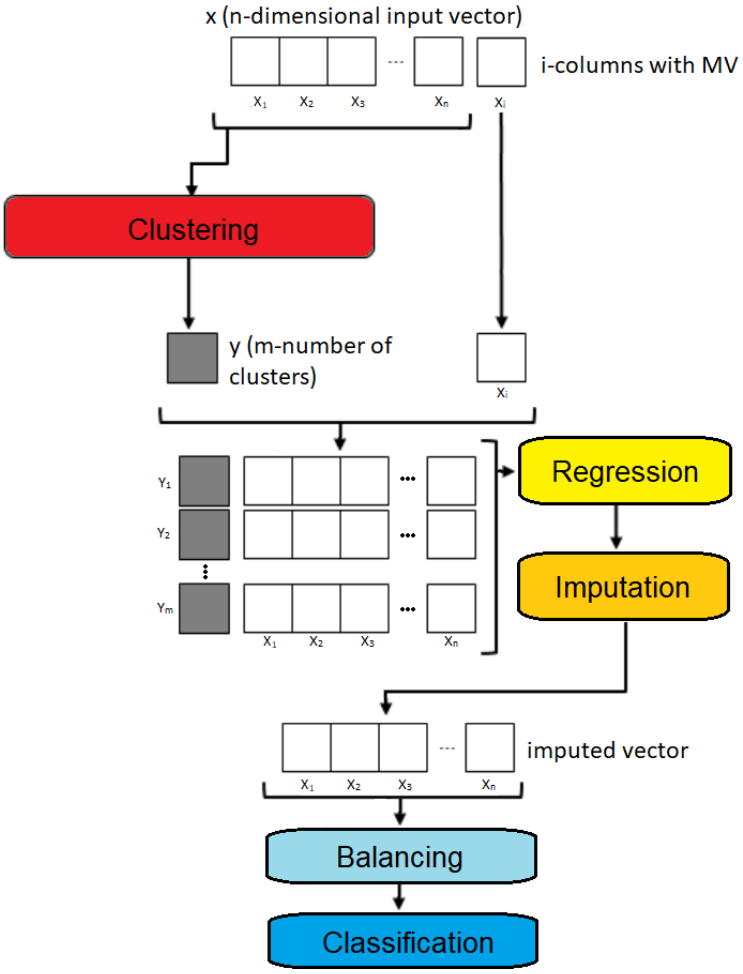

Fig. 1. Hybrid system novel formulation.

The hybrid ML solution proposed in the present work is divided into the following four steps:

1. Clustering: three clustering techniques are applied to the data set to obtain more homogeneous groups of data.
2. MV Imputation: two regression techniques are applied to each of the clusters generated in the previous step to impute the MV, therefore all the single attributes are valued.
3. Data Balancing: a series of balancing techniques are applied to already imputed data set to achieve a greater balance between the minority or anomalous class and the majority or normal class.
4. Classification: given the large number of sets obtained from the previous steps developed, the well-known Support Vector Machine (SVM) [31] classifier is used. For a clear interpretation of the results obtained by this process, several metrics are used to compare the results.

For the selection of different techniques in each of the steps of the present hybrid proposal, techniques previously applied to the data set have been used. For the selection of clustering techniques based on [1], only k-means and clustering techniques were used, although as detailed in the future work, the inclusion of hierarchical and density-based methods could be interesting. Secondly, for imputation techniques in [2] was concluded that, in general, the regression techniques that worked best were those used in the present investigation. Finally, for balancing techniques in the research carried out [3], all the techniques presented here were used with the exception of Borderline-SMOTE (BLSMOTE), whose adhesion is due to the fact that it is derived from SMOTE, a technique that stood out from the rest and could be of interest.

3.1 Clustering

K-means. Cluster analysis [18] organizes data by grouping data samples according to a criterion distance. Two individuals in a valid group will be much more similar than those in distinct groups. The k-means clustering algorithm [23]

groups data samples into a predefined number of groups. It requires two input parameters: the number of clusters (k) and their initial centroids. Initially, each data sample is assigned to the cluster with the closest centroid. Once the clusters are defined, the centroids are recalculated and samples are reassigned. These steps are repeated until there is no further change in the centroids. The quality criterion to measure the grouping is the Sum of Squared Errors (SSE). The algorithm to minimize it can be defined as follows:

$$SSE = \sum_{j=1}^{k} \sum_{x \epsilon G_i} p(x_i, c_j)/n \tag{1}$$

where, k is the number of groups, p is the proximity function, c_j is the centroid of group j, and n is the number of samples. Different measures of distance have been tested to obtain the best results, with the Euclidean distance being the chosen one. In this distance, each centroid is the mean of the points in its cluster. Is defined as:

$$d_{st}^2 = (x_s - y_t)(x_s - y_t)' \tag{2}$$

where d is the distance from point x to centroid c. In the run experiments, the Means++ algorithm has been used for the initialization of centroids.

Hierarchical. Hierarchical clustering algorithms are top-down or bottom-up implementations. Bottom-up approaches treat each sample as a single cluster at the beginning, and then successively merge pairs of clusters until it merges all clusters into a single cluster containing all the samples. The bottom-up Hierarchical clustering, also called Hierarchical Agglomerative Cluster (HAC), generates a cluster tree or dendrogram by using heuristic techniques. A dendrogram [30] comprises many U-shaped lines connecting data points in a Hierarchical tree. The height of each U represents the distance between the two connected data points. The most popular algorithms that use merging to create the cluster tree are called agglomerative. There are many implementations of HAC [12]. Similar to the 3.1 Section, the Euclidean distance (equation 2) is chosen.

Density-based Spatial Clustering of Applications with Noise (DBSCAN). This clustering algorithm is based on density, i.e. it analyzes regions whose points have a higher density separated by others with a lower density [14]. In DBSCAN, each point sets a radius around itself, counting the number of points that fall within it. A minimum number of points that must be within this radius is established to know if they are part of the same group as the initial point. This algorithm does not follow centrality hypotheses as in the case of k-means, but produces complex groups. There are three types of points:

- Core points: the points that are in the interior of a group near the center.
- Border points: those located at the edge of the radius.
- Noise points: those located neither one nor the other and are not part of any group.

3.2 Regression Techniques

In the proposed ML hybrid system, once data are clustered, two regression techniques (statistic and ANN) are applied to the defined clusters to get more accurate results.

Regression attempts to model the relationship between two or more variables in the data set by fitting a linear equation to the input data. One or more of the variables are the predictor ones, and the other variable is considered the criterion variable [41]. The goal of multiple regressions [26] is to learn more about the relationship between the independent or predictor variables and a dependent or criterion variable(s). These relationships can be linear or non-linear.

Non-Linear Regression. Non-Linear Regression (N-LR) is a regression algorithm which models observational data by a function that is a non-linear combination of the input data and depends on one or more criterion variables [24]. The parameters can take the form of an exponential, trigonometric, power, or any type of non-linear function. To determine the non-linear parameter values,

an iterative algorithm is usually used. The model is defined as:

$$y = f(X, \beta) + \varepsilon \tag{3}$$

Where β is a nonlinear parameter estimates to be computed, X is the dependent variables and ϵ represents the error term.

Radial Basis Function Network. An ANN is a simplified model of natural neural systems. The neurons are connected by weights and output signals which are the sum of the inputs to the node modified by an activation function. Different ANN models have been tested to achieve the best imputation values, the one with the best results has been the Radial Basis Function Network (RBFN) which is defined as:

In the RBFN [22] each neuron in the hidden layer has its own n-dimensional centroid, and for each input vector $x = (x_1, x_2, \ldots, x_n)$, it computes the distance between x and the centroid of the network. A nonlinear Gaussian function distance is used to calculate the output of the neurons.

The overall output function has the form [25]:

$$\sum_{i=1}^{M} W_i * K\left(\frac{x - z_i}{\sigma_i}\right) = \sum_{i=1}^{M} W_i * g\left(\frac{||x - z_i||}{\sigma_i}\right) \tag{4}$$

Where x is the input vector,$\mathrm{W}_i \ \epsilon \mathbb{R}^m$ are the weights connecting the ith neuron in the hidden-layer to the output neurons, $M \epsilon \mathbb{N}$ is the number of hidden neurons, K is a radially symmetric kernel function of a unit in the hidden layer, z_i is the centroid and σ_i is the smoothing factor of the kernel node, g: $[0, \infty) \rightarrow \mathbb{R}$ is the activation function of the output neurons.

3.3 Balancing Techniques

Balancing techniques are a highly utilized resource when dealing with unbalanced data sets. When the classes of the data set are due to anomalous states, these data sets are highly unbalanced. The study of the data set, detailed in Section 4, shows that out of 21892 instances, only 1125 belong to the anomalous class, i.e.

5% of the total number of instances. There are mainly three types of approaches used to carry out the balancing:

Oversampling. This strategy tries to achieve a similar number of instances of both classes by increasing the number of instances of the minority class. In this case, it generates new instances of the anomalous class to obtain a similarity between the number of instances of both classes. The simplest technique used is Random Over Sampling (ROS), which generates new instances by duplicating existing instances of the minority class. Another method widely used in this field, which is more advanced and has a higher complexity than ROS, is Synthetic Minority Oversampling TEchnique (SMOTE) [8], which generates new artificial instances from the existing ones. It achieves this by relying on the well-known k-Nearest Neighbors (KNN) algorithm, performing an interpolation of a minority instance with other neighboring instances. Finally, another technique used in this research is the Borderline-SMOTE (BLSMOTE) [17]. As its name suggests, it is based on the SMOTE method, in which an oversampling is performed only on those instances that are on the borderline.

Undersampling. The balancing algorithms that follow this strategy work in a completely opposite way to what was observed in the above mentioned methods. They try to reduce the instances of the majority class in order to achieve a number of similar instances. In the case addressed here, they eliminate the instances of the normal class. The algorithm used is Random Under Sampling (RUS), which selects completely randomly the instances of the majority class to be eliminated.

Hybrids. Hybrid techniques are those which use both, undersampling and oversampling algorithms at the same time. This reduces the impact of using only one of them. One of the methods presented in this research is ROS + RUS, that combines the two algorithms based on the random selection of instances above mentioned. Another technique used is SMOTE + RUS, which generates new

synthetic instances of the minority class and randomly eliminates those of the majority class.

3.4 Classifiers and metrics

The One-class SVM [9] [5] is one of the best known classifiers in general terms and specifically in the problems associated with one-class classification [29]. This classifier aims at identifying a hyperplane that maximizes the separation of the data instances sent to the algorithm from the training data set. In this way, once new data instances are used, it will be able to discern which class each one is, due to the universal archetype generated.

Working on unbalanced data sets, the well-known metric accuracy is not used on it's own, because it can lead to a high error in interpretation of its values, for example, the model may have the ability to distinguish only the majority class obtaining a good value, but on the other hand, the model does not have the capacity to detect the minority class. In this research, the results are shown with a wide variety of metrics, which are detailed below:

- Precision: shows the proportion of minority or anomalous class data that have been successfully labeled out of the total data labeled as anomalous.
- Recall: also known as True Positive Rate (TPR), shows the proportion of anomalous class well classified out of the real number of anomalous instances.
- F-Score: it is a metric that seeks a harmonic mean between Precision and Recall, given the difficulty in maximizing the values of both at the same time.
- AUC: this is the Area Under the Curve resulting from the visual ROC tool. It is used as an indicator of the model's ability to distinguish between classes.
- G-mean: it seeks to maximize the accuracy of both the minority and majority classes, while looking for a balance between them.

4 Component-based Robot

A data set of a component-based robotic system [37] is used for this experimentation. Component-based means that the robot is made up of a variety of components that may have been manufactured by different companies, but thanks to middle-ware they are interconnected and they can work as a unified system. The middle-ware used by this robotic system is RSB Middleware [34], which is an event-based system. Within the middle-ware, a tool called rsbag is located, which is in charge of gathering the information that circulates through the middle-ware. This tool is key for the data extraction used in this paper.

The data set has been created by researchers at the University of Bielefeld (Germany) and is available in the public domain [39]. The robot was developed for its participation in the RoboCup@Home competition in 2015, where it had to carry out different tasks similar to a waiter. Some of these tasks are greeting a customer or serving a glass at the table. These actions are developed by relying on different components, for example the action of leaving a glass on the table uses a robotic arm with a gripper to pick up a glass.

Trying to detect possible errors in the robot's behavior, the authors of this data set decided to induce anomalies by software, implying that these affect the system's performance counters without penalizing the task from being carried out. For example, in the case of the robotic arm mentioned above, the anomaly causes the arm to move several times instead of once, thus penalizing these counters.

Among the various available components and given the great complexity of the experimentation, the LegDetector component has been selected, with its associated anomaly LegDetectorSkippable. This component is in charge of detecting the legs of a person in front of the robot thanks to a laser sensor. The anomaly in this case affects the counters by performing the reading attempt a number of times.

The selection of this component is due to a current problem: the need for visual processing and object recognition [20]. As the good values that were orig-

inally obtained in the experiments carried out by the authors of the data set [40] and by us in [3].

Among the attributes that constitute this component, two of them contain MV:

- received_bytes: amount of data in unit of bytes that are hosted on the interface.
- sent_bytes: amount of data in unit of bytes that are dispatched by the interface.

5 Experiments and results

This section shows the results of applying the set of techniques discussed in Section 3 to the data set described in the Section 4.

The first step of the hybrid system described above is the application of three clustering techniques to the data set. To do this, the number of desired clusters (parameter k) must be provided. Estimating this value for k is not always straightforward, many techniques help us to estimate this parameter [6, 11, 28, 33]. All of them have been applied with different ranges of values for parameter k but no satisfactory results have been achieved, since the techniques return values that diverge from from each other.

Another option to estimate this value for the parameter k is the use of dendrograms. The Figure 2 shows the dendrogram for the original normalised data set.

Figure 2 depicts two clusters of data, one on the right-hand side with a few samples and the other on the left-hand side which groups most of the samples. This graphical result induces as most approximate values for the parameter k = (2, 3). The value of 3 because in the majority group on the right two subsets of data are distinguished.

Once the number of clusters has been selected, the different algorithms described in the Section 3.1 are applied. The clustering algorithms are applied in

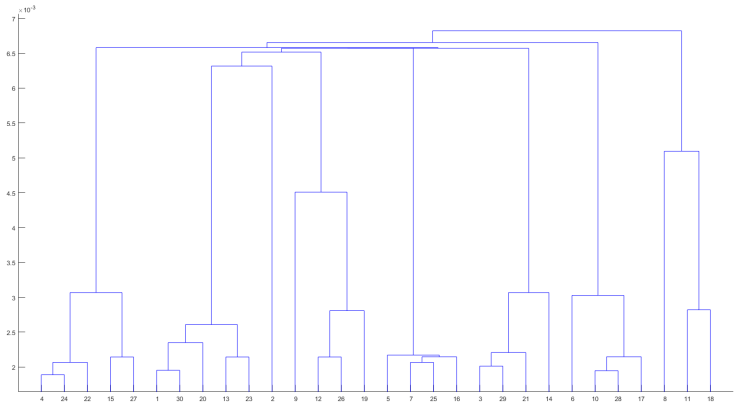

Fig. 2. Dendrogram with 30 leaf nodes ('Euclidean' distance, 'Complete' linkage method).

the data set omitting the attributes where the MV were located and described in Section 4. From these generated clusters, the attributes with MV are added again and the regressions are carried out with the methods detailed in 3.2. The imputation of MV is done in each cluster. After the classification is carried out in order to compare which combination of regression and clustering algorithms works best.

For a better understanding of the tests performed, for each of the two regression techniques, two different clustering distributions (2 and 3) have been used, and these distributions have been obtained from 3 clustering algorithms, giving a total of 12 different runs. In each of these runs, 6 balancing algorithms and the execution of the data set without any type of treatment have been used, this last one has been denominated as "None".

1. In Section 5.1 the results are analyzed from the perspective of the regression algorithms used.
2. In Section 5.2 from the perspective of the regression methods used.
3. In Section 5.3 from the balancing techniques used.

5.1 Regression methods approach

An interesting comparison is the one between the two regression techniques used in this research (N-LR and RBFN) in each of the three clustering methods applied (K-means, HAC and DBSCAN). The Figure 3 shows a comparison of the different values achieved by each regression method in metrics. The first Figure shows the F-Score (a), where the general trend is that N-LR performs better than RBFN in most of the generated models, but taking special attention to the value achieved in DBSCAN with 3 clusters where it is observed that RBFN is slightly better. In the case of AUC (b), a generalized growth of the values is observed, with very good results in general. The values of both regression techniques reach similar values, making it difficult to conclude the best approach. Finally, the g-mean metric again shows greater variability in the results, where the general trend is that N-LR performs better in general terms, highlighting the values obtained by DBSCAN.

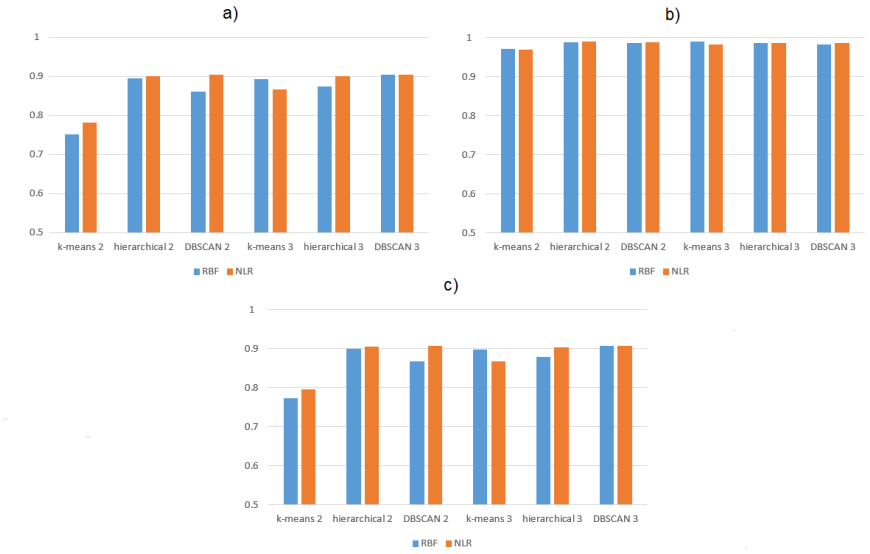

Fig. 3. Bar plot showing the differences between RBFN and N-LR in the different metrics. a) F-Score, b) AUC and c) g-mean.

5.2 Clustering techniques approach

A more global approach is chosen for the analysis of the clustering results, considering a wider range of metrics. It has been subdivided according to the number of clusters (k= 2, 3). Following this general approach, the results are displayed with radar plots 4, by this, the differences in the metrics can be recognized more intuitively. The Figure 4 for the two clusters(a) shows how in the general trend, Hierarchical and DBSCAN algorithms perform better than K-means, especially outperforming in g-mean and F-Score. Finally, the similarities achieved in AUC are remarkable. On the other hand, in the values achieved with 3 clusters(b), these are much more similar highlighting K-means standing out slightly in Recall.

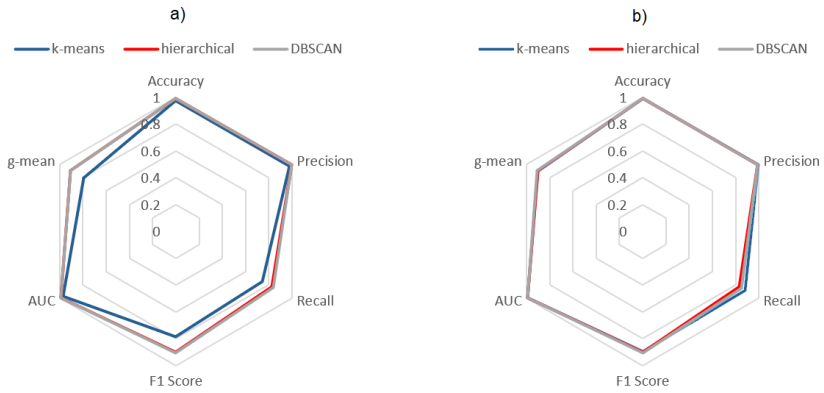

Fig. 4. Radar diagrams are obtained from the results for each clustering technique with its different algorithms. Each one shows the results with different clusters a) k=2 and b) k=3.

5.3 Balancing methods approach

The last approach adopted to analyze this novel hybrid system focuses on the study of the results obtained by each of the balancing methods applied. Table 1

shows the best values obtained for each balancing method. The values obtained by the Accuracy and Precision metrics are very good, with the peculiarity of two techniques coinciding in value, ROS + RUS in both cases. This trend continues with F-Score and g-mean where it stands out again over the rest, although BLSMOTE results are also very good. Unexpectedly, Recall technique overcomes the other ones, where the non-application of any kind of balancing technique stands out from the rest so far.

Table 1. Metrics values for each of the balancing methods.

	None	ROS	SMOTE	BLSMOTE	RUS	ROS + RUS	SMOTE + RUS
Accuracy	0.9867	0.9846	0.9889	**0.9896**	0.9889	**0.9896**	0.9881
Precision	0.9751	**0.9964**	0.9893	0.9893	0.9929	**0.9964**	0.9929
Recall	**0.8824**	0.7790	0.8293	0.8478	0.8254	0.8457	0.8176
F-Score	0.8680	0.8696	0.9008	0.9055	0.9015	**0.9058**	0.8953
AUC	0.9694	0.9902	0.9874	0.9860	**0.9908**	0.9884	0.9903
g-mean	0.8681	0.8767	0.9042	0.9076	0.9053	**0.9081**	0.8994

After analyzing the different possibilities offered by this data set, there is a general tendency that the data sets treated by the DBSCAN clustering algorithm provide better results than the other techniques, see Sections 5.1 and 5.2. The difference between RBFN and N-LR has not been very marked, although, as mentioned above, N-LR has generally performed better than RBFN. In terms of balancing techniques, the hybrid ROS+RUS technique performed better than the rest, followed closely by the BLSMOTE oversampling technique.

6 Conclusions and Future Work

In this paper, a novel alternative in data imputation has been addressed, together with a set of techniques which seek to perform a complete data treatment. The proposed ML hybrid system has been validated on a data set of a component-based robotic system. The experimentation has been divided into different stages and the results obtained have been analyzed in Section 5.

The data set offered by the robot with anomaly information had some MV in two of their attributes (received_bytes and sent_bytes). In order to estimate these MV and have available a complete data set, the ML Hybrid System proposed in Section 3 was applied. A detailed analysis of each of the techniques of this hybrid system has led to the following conclusions. The three clustering techniques described offered good results, but DBSCAN stands out on the positive side and the Hierarchical technique on the negative side due to its slowness. Regarding to regression, has been performed on each of the clusters, both N-LR and RBFN obtained very good results with really low MSE values, highlighting slightly N-LR. These good results in the regression have allow to achieve a very reliable imputation on the MV and have at its disposal higher quality data set. In terms of balancing techniques, whose have been applied to the new dataset, the good performance of the hybrid ROS+RUS technique stands out overall.

The main objective of this paper was to demonstrate a novel system and the different alternatives with which to execute it, in terms of combining techniques of Machine Learning. Satisfactory results have been achieved in the implementation of the proposed hybrid model, which leads us to conclude that that the Hybrid Machine Learning System is a valid alternative for future researchers in this topic.

As future work, it would be interesting to combine new regression and clustering methods, but without losing the target of continuing with this modern hybrid system. The application of this approach to more data sets is undoubtedly an option to be considered.

References

1. Arroyo, Á., Basurto, N., Cambra, C., Herrero, Á.: Clustering and Regression to Impute Missing Values of Robot Performance. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 12344 LNAI, pp. 86–94. Springer Science and Business Media Deutschland GmbH (nov 2020). https://doi.org/10.1007/978-3-030-61705-9_8, https://link.springer.com/chapter/10.1007/978-3-030-61705-9_8
2. Basurto, N., Arroyo, Á., Cambra, C., Herrero, Á.: Imputation of Missing Values Affecting the Software Performance of Component-based Robots. Computers and Electrical Engineering **87**, 106766 (oct 2020). https://doi.org/10.1016/j.compeleceng.2020.106766
3. Basurto, N., Cambra, C., Herrero, Á.: Improving the detection of robot anomalies by handling data irregularities. Neurocomputing **Under review** (2020)
4. Basurto, N., Herrero, Á.: Data selection to improve anomaly detection in a component-based robot. In: Martínez Álvarez, F., Troncoso Lora, A., Sáez Muñoz, J.A., Quintián, H., Corchado, E. (eds.) 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019). pp. 241–250. Springer International Publishing, Cham (2020)
5. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. pp. 144–152. COLT '92, ACM, New York, NY, USA (1992). https://doi.org/10.1145/130385.130401
6. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods **3**(1), 1–27 (1974)
7. Cerqueira, V., Pinto, F., Sá, C., Soares, C.: Combining boosted trees with metafeature engineering for predictive maintenance. In: Boström, H., Knobbe, A., Soares, C., Papapetrou, P. (eds.) Advances in Intelligent Data Analysis XV. pp. 393–397. Springer International Publishing, Cham (2016)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
9. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018

10. Das, S., Datta, S., Chaudhuri, B.B.: Handling data irregularities in classification: Foundations, trends, and future challenges. Pattern Recognition **81**, 674 – 693 (2018). https://doi.org/10.1016/j.patcog.2018.03.008
11. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence **PAMI-1**(2), 224–227 (1979)
12. Day, W.H., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. Journal of classification **1**(1), 7–24 (1984)
13. Devi, D., Biswas, S.K., Purkayastha, B.: Learning in presence of class imbalance and class overlapping by using one-class svm and undersampling technique. Connection Science **31**(2), 105–142 (2019). https://doi.org/10.1080/09540091.2018.1560394
14. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD-96: Proceedings. pp. 226 –231 (1996)
15. García-Laencina, P.J., Sancho-Gómez, J.L., Figueiras-Vidal, A.R.: Pattern classification with missing data: a review. Neural Computing and Applications **19**(2), 263–282 (Mar 2010)
16. García-Laencina, P.J., Sancho-Gómez, J.L., Figueiras-Vidal, A.R.: Pattern classification with missing data: a review. Neural Computing and Applications **19**(2), 263–282 (2010)
17. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Lecture Notes in Computer Science. vol. 3644, pp. 878–887. Springer Verlag (2005). https://doi.org/10.1007/11538059_91, https://link.springer.com/chapter/10.1007/11538059_91
18. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Comput. Surv. **31**(3), 264–323 (Sep 1999). https://doi.org/10.1145/331499.331504, https://doi.org/10.1145/331499.331504
19. Jove, E., Casteleiro-Roca, J.L., Quintián, H., Simić, D., Méndez-Pérez, J.A., Luis Calvo-Rolle, J.: Anomaly detection based on one-class intelligent techniques over a control level plant. Logic Journal of the IGPL (01 2020). https://doi.org/10.1093/jigpal/jzz057, https://doi.org/10.1093/jigpal/jzz057, jzz057

20. Kasaei, S.H., Oliveira, M., Lim, G.H., Lopes, L.S., Tome, A.M.: Towards lifelong assistive robotics: A tight coupling between object perception and manipulation. Neurocomputing **291**, 151 – 166 (2018). https://doi.org/10.1016/j.neucom.2018.02.066, https://doi.org/10.1016/j.neucom.2018.02.066
21. Khalastchi, E., Kalech, M.: On fault detection and diagnosis in robotic systems. ACM Comput. Surv. **51**(1), 1–24 (Jan 2018). https://doi.org/10.1145/3146389
22. Lippmann, R.P.: Pattern classification using neural networks. IEEE Communications Magazine **27**(11), 47–50 (Nov 1989). https://doi.org/10.1109/35.41401, https://doi.org/10.1109/35.41401
23. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA (1967)
24. Neter, J., Kutner, M.H., Nachtsheim, C.J., Wasserman, W.: Applied linear statistical models, vol. 4. Irwin Chicago (1996)
25. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Computation **3**(2), 246–257 (June 1991). https://doi.org/10.1162/neco.1991.3.2.246, https://doi.org/10.1162/neco.1991.3.2.246
26. Pearson, K., Lee, A.: On the generalised probable error in multiple normal correlation. Biometrika **6**(1), 59–68 (1908), http://www.jstor.org/stable/2331556
27. Pigott, T.D.: A review of methods for missing data. Educational research and evaluation **7**(4), 353–383 (2001)
28. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
29. Shin, H.J., Eom, D.H., Kim, S.S.: One-class support vector machines - An application in machine fault detection and classification. Computers and Industrial Engineering **48**(2), 395–408 (mar 2005). https://doi.org/10.1016/j.cie.2005.01.009
30. Sokal, R.R., Rohlf, F.J.: The comparison of dendrograms by objective methods. Taxon **11**(2), 33–40 (1962)
31. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural processing letters **9**(3), 293–300 (1999)

32. Syafrudin, M., Fitriyani, N.L., Alfian, G., Rhee, J.: An affordable fast early warning system for edge computing in assembly line. Applied Sciences **9**(1), 84–102 (2018). https://doi.org/10.3390/app9010084

33. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63**(2), 411–423 (2001)

34. Wienke, J., Wrede, S.: A middleware for collaborative research in experimental robotics. In: 2011 IEEE/SICE International Symposium on System Integration (SII). pp. 1183–1190 (Dec 2011). https://doi.org/10.1109/SII.2011.6147617, https://doi.org/10.1109/SII.2011.6147617

35. Wienke, J., Wrede, S.: Performance regression testing and run-time verification of components in robotics systems. Advanced Robotics **31**(22), 1177–1192 (nov 2017). https://doi.org/10.1080/01691864.2017.1395360, https://www.tandfonline.com/action/journalInformation?journalCode=tadr20

36. Wienke, J.: Framework-level resouce awareness in robotics and intelligent systems. PhD dissertation, Bielefeld University (2018). https://doi.org/10.4119/unibi/2932136, https://doi.org/10.4119/unibi/2932136

37. Wienke, J., Meyer zu Borgsen, S., Wrede, S.: A data set for fault detection research on component-based robotic systems. In: Alboul, L., Damian, D., Aitken, J.M. (eds.) Towards Autonomous Robotic Systems. vol. 9716, pp. 339–350. Springer International Publishing, Cham (2016)

38. Wienke, J., Wigand, D., Köster, N., Wrede, S.: Model-based performance testing for robotics software components. In: Proceedings - 2nd IEEE International Conference on Robotic Computing, IRC 2018. vol. 2018-January, pp. 25–32. Institute of Electrical and Electronics Engineers Inc. (apr 2018). https://doi.org/10.1109/IRC.2018.00013

39. Wienke, J., Wrede, S.: A Fault Detection Data Set for Performance Bugs in Component-Based Robotic Systems. https://doi.org/10.4119/unibi/2900911, https://doi.org/10.4119/unibi/2900911

40. Wienke, J., Wrede, S.: Autonomous fault detection for performance bugs in component-based robotic systems. In: Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. pp. 3291–3297. IEEE (2016). https://doi.org/0.1109/IROS.2016.7759507

41. of Yale, U.: Linear Regression (2017), http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

Title: "A Hybrid Machine Learning System to Impute and Classify a Component-Based Robot"

Authors: Nuño Basurto, Ángel Arroyo, Carlos Cambra, Álvaro Herrero

Manuscript number: HAIS2020-IGPL-9

First of all, we would like to thank the reviewers once again for their valuable comments and corrections that have let us improve our contribution further. The individual answers to each of the reviewer comments are set out below.

Reviewer 1

Comment 1.1	A detailed state of the art section can be introduced after Section 1. The authors can include here part of the description already available provided in the Introduction section. This explanation must be extended with a detailed description of the main pros and cons of the authors' proposal with regard related work.
Answer 1.1	Authors than the reviewer for his/her suggestion. Accordingly, Section 2 (State of the art) has been added to the manuscript.
C 1.2	In Section 2 the authors must justify the reasons the described set of specific clustering, regression and balancing techniques have been selected and no other. The explanation in this section can be substantially reduced.
A 1.2	The justification for the use of the different techniques has been included in Section 3 before the individualized presentation of each technique.
C 1.3	The authors must describe in Sections 3 and 4 the evaluations completed with other authors using the same reference corpora in order to provide a comparative assessment with the authors' proposal.
A 1.3	Although very few authors have analysed the benchmark dataset so far, further information about them has been added to Section 2. The only previous publication on this very same problem, analysing the same data is: Clustering and Regression to Impute Missing Values of Robot Performance This is the HAIS 2020 conference paper (by the same authors of the present paper) that has been invited to submit an extended version to the present special issue.
C 1.4	There is not text in the point number 4 of the enumeration in the conclusions section. These points can be better introduced in the explanation without using this enumeration.
A 1.4	The points have been deleted and each one of the evaluations carried out have been adequately explained.

Reviewer 2

Comment 2.1	The abstract is well structured, the introduction section states the problem clearly, although it should be separated into ‘Introduction’ and ‘Related Works’.
Answer 2.1	As previously stated (Answer 1.1), Section 2 (State of the art) has been added.
C 2.2	There are some minor issues with the use of English
A 2.2	All minor errors have been fixed and the whole text has been proofread.