Available online at www.sciencedirect.com

ScienceDirect

Neurocomputing 00 (2024) 1–19

Neurocomputing

Improving the Detection of Robot Anomalies by Handling Data Irregularities

Nuño Basurto[0000−0001−7289−4689], Carlos Cambra[0000−0002−2444−5384], Álvaro Herrero[0000−0002−2444−5384]

Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Ingeniería Civil, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006, Burgos, Spain.

{*nbasurto, ccbaseca, ahcosio*}*@ubu.es*

Abstract

The ever-increasing complexity of robots causes failures of them as a side effect. Successful detection of anomalies in robotic systems is a key issue in order to improve their maintenance and consequently reducing economic costs and downtime. Going one step further in the detection of anomalies in robots, different mechanisms to deal with data irregularities are proposed and validated in present paper in order to increase detection rates. More precisely, strategies to overcome missing values and class imbalance are considered as complementary tools to get better one-class classification results. The effect of such strategies is evaluated through cross-validation when applying a standard supervised learning model, the Support Vector Machine. Experiments are run on an up-to-date and public dataset that contains some examples of different software anomalies that the middleware of the robot under analysis may experience.

Keywords: component-based robot, missing values, data balancing, anomaly detection, supervised learning, support vector machine

1. Introduction

It is widely acknowledged that in present fourth industrial revolution, knowledge extraction from large volumes of data is a crucial task. There are different facilitators [1] to support the transfer to Industry 4.0 [2], that include Artificial Intelligence in general and Machine Learning (ML) in particular. Among all the resources associated to the smart and future factories, robots play a key role [3]. There has been a 31% increase of industrial robots, reaching 384,000 units in the world in just one year, as reported by the Industrial Federation of Robotics [4]. Furthermore, the annual sales volume of industrial robots have been increasing from last 6 years (2013-2018). In parallel to the increase of sale figures, the complexity of robots is constantly growing over time. Additionally, the demands for robustness and reliability are increasing as well. However, as any cyber-physical system, robots suffer from failures and finding anomalies is required in order to allow recovery and continuous operation. Further effort must be devoted to anomaly detection in robots as little attention has been paid to it by the international research community until now [5].

Present paper addresses the detection of performance anomalies experienced by the software of a robotic system. Due to the widely acknowledged importance of data pre-processing, different such mechanisms (mainly data balancing and handling of missing values - MV) have been applied in order to better identify anomalies. Based on previous work on the same real-life dataset [6], [7], experiments are conducted by means of the One-Class Support Vector Machine (SVM), that is discussed in section 4.

The successful detection (and identification in multiclass cases) of anomalies/faults is a challenging task that does not only apply to robots [8], [9], [10]. For the benefit of industrial companies in general [11], and for the automatic anomaly detection in particular, ML techniques have been successfully applied in different fields [12] up to now.

Among the vast amount of classifiers that exist, SVM is one of the most widely applied ones for anomaly/fault detection as it has proved to be a successful model. In [13] it is applied to a multi-sensor motor after preprocessing data by means of the FShort-Time Fourier Transform. The aim is reducing maintenance costs of the electro-mechanical system of the motor. More recently, Zidi et al. [14] proposed the use of SVM in the Wireless Sensor Networks field where anomalies could come from different sources, such as software, hardware or the communication system. SVM was benchmarked against some other well-known classifiers such as Naive Bayes (NB) or Hidden Markov Models, obtaining positive results.

Within the stage of data preprocessing, necessary before the pattern recognition one, data irregularities are usually found and must be overcame [15]. Present paper focuses on two of these irregularities, namely MV [16] and data imbalance [17]. There have been previous proposals to deal with the MV in robot data, such as the one proposed by Twala [18], in which a probabilistic approach is used, based on the a-priori probability of each value determined from the instances in that node which have specified values. Robot failures are detected by applying a well-known classifier: a Decision Tree (DT). The classifier is applied to all available data collected from the sensors of the robot; it does not matter the type of attribute (whether numeric or nominal). The author applied and compared different imputation techniques for handling the MV. As opposed to this previous work, present paper deals with the software of a robot and strategies for discarding MV rather than imputing them, due to the induced error of imputation.

The imbalanced class distribution is another important issue that must be addressed before applying supervised learning techniques. Several approaches to deal with this problem [17], [19] have been proposed up to now. Dataspace weighting [20] was proposed in order to balance the classes by assigning different weights to instances of different classes. As a result, classes have the same total weigh, with a positive impact on classification rate. On the other hand, Cerqueira et al. [21] adopted an approach for dealing with MV similar to the one in present research: deleting them. Additionally, they used the Synthetic Minority Over-sampling Technique (SMOTE) to get a class-balanced distribution of data that improved the classification performance. The aim of such classification was carrying out a predictive maintenance (that is, detecting anomalies) on the air pressure system of heavy trucks. More recently, another study [22] has been published where SMOTE is applied for anomaly detection. In order to detect abnormal events in an assembly line, data are processed (to remove outliers) with DBSCAN and then SMOTE is applied for data balancing. Finally, Random Forest (RF) is used as the learning model for anomaly prediction. RF is also applied in [23] to detect and classify failures of a vehicle fleet. Additionally, a parameter tuning framework is proposed to overcome the class imbalance problem. Similarly, Luo et al. [24] considered the problem of imbalanced data and its implications in anomaly detection. In order to solve it, they generated new synthetic data samples by means of a technique called Triangle Syntethic Data, that is an extended version of SMOTE. They have used some standard classifiers, such as DT, Logistic Regression, SVM, and NB, so they can verify the universality of the algorithms. Rather than proposing the application of one balancing method, such as SMOTE, present paper is a comprehensive study of the application of different balancing methods to improve anomaly detection.

One-class classification has been also addressed before, together with data balancing techniques. In [25] authors analyze the effect that the imbalance of classes can have in seven one-class and two multiclass datasets. Six classification models, such as NB or SVM are compared when applying the Totem-Links undersampling technique. The model finally proposed by the authors, based on the SVM classifier, managed to improve the tendency of the minority class without affecting the majority class.

In the case of robotic systems, most previous work has been focused on the detection of hardware anomalies while few papers deal with software anomalies, which have been largely ignored. Software failures often occur in robotic systems and their automatic detection requires training data. The problem comes from the difficulty of obtaining the data either because of the lack of execution traces or because the existing registers do not refer to the exact moment in which they are produced. That is why it is difficult to find a dataset generated in a controlled environment where all the information is available. One of the pioneer works on the detection of software anomalies within the framework of component-based robots is [7]. In that paper, authors propose the only publicly-available dataset (further details in section 3) that gathers data from different performance indicators of a robot. The dataset [26] has been used in present paper as a benchmark dataset due to its interest and novelty. Authors of the dataset applied [27] One-Class SVM (OCSVM) in order to compare its performance with that obtained when using another model. Thus, present paper focuses on the effect of data irregularities on classification by OCSVM.

In the doctoral dissertation [28] associated to this dataset, compiling all the previous papers by these authors, they explored two alternatives. Firstly, they considered methods to understand and systematize resource control, for which a set of tools was developed. On the other hand, they studied the topic that leads to present work: the use of different ML techniques to detect anomalies and allow automatic reactions in execution time, based on the use of component resources.

The rest of this paper is organized as follows: the proposed framework for anomaly detection (comprising the applied classifier, the pre-processing strategies and the performance metrics) is described in section 2 while the case study and its associated dataset is described in section 3. The setup of performed experiments and the obtained results are presented in section 4. Finally, section 5 introduces the main conclusions derived from present research and points outs some proposals for future work.

2. Proposed Framework for Anomaly Detection

Detection of anomalies is known as the problem of finding certain patterns in the data that do not conform to a expected behavior [8]. This "anomalous" behaviour may be associated to failures or malfunctioning of any kind. Anomaly detection in the software of a robotic system is addressed in present paper by using the framework that is described in this section. The SVM classifier (see section 2.1) is used as the learning method to be trained on the analyzed dataset (described in section 3). The applied pre-processing techniques are then explained in section 2.2 and the different metrics that have been observed in order to compare the performance are described in section 2.3.

2.1. Learning Method

The SVM [29], [30] is a widely-applied classifier that implements the Statistical Learning Theory. The purpose of this shallow ML model is to identify the hyperplane that maximizes the separation margin of data, according to the defined classes in the training dataset. For generalization purposes, it tries to universalize the archetype that will be used to classify the new data samples. This is the Structural Risk Minimization perspective, as opposed to the Empirical Risk Minimization one, that is implemented in other models such as neural networks. SVM for one-class classification (as the anomaly detection in present paper) is a learning model whose loss function is the Hinge function, defined as:

$$L[y, f(x)] = max[0, 1 - yf(x)] \tag{1}$$

Being x one of the observations taken from the input data, and y is the class x belongs to. *f(x)* is the output of the SVM itself. During training, the SVM identifies the support vectors that are those data samples that maximize the separation of data. Being S the set of support vectors, α the coefficients of the classifier, and β the coefficients of the predictor, once trained the SVM can be defined as in equation 2.

$$f(x) = \Sigma_{i\epsilon s}\alpha \cdot y_i \cdot \langle x_i, x\rangle + \beta_0 \tag{2}$$

In present paper, a SVM equipped with a sigmoidal kernel function has been used. This function is defined as:

$$k(x, y) = tanh(ax^T y + c) \tag{3}$$

2.2. Data Pre-processing

As previously stated, two data irregularities are addressed in present work. MV is an issue mainly when working in a field where sensors are involved (as present case study). It is even more important due to the fact that most supervised learning methods (SVM included) can not deal with MV. In present paper, MV are removed from the data in order to apply the mentioned classifier. There are mainly two ways of removing MV: deleting those data instances containing at least one of these values for any of the features or deleting those features containing at least one of these values for any of the data instances. The former causes a reduction in the number of data samples while the latter causes a reduction in the number of features, contributing the two of them to negative effects in the learning of the classifier. In order to find an equilibrium, present paper proposes establishing permissiveness ratios for MV when removing data features. That is, features containing a percentage of MV below the threshold are kept while all the others are removed. For those features that are below the threshold and contain any MV, data instances comprising such MV are subsequently removed. Consequently, by increasing the MV ratio more features are considered by the classifier but less instances and vice versa. To empirically set up such ratio, values of 0%, 10%, 25%, and 50% have been tested in the conducted experiments (see section 4).

On the other hand, the imbalance of classes often appears in datasets for anomaly detection. It results on the majority class ("normal" status of systems) getting a benefit from the classifier and being prejudicial to the results for the minority class (anomalies/failures). In order to deal with this problem, different solutions (known as balancing methods) have been proposed so far [19]. They are aimed at ensuring that the different classes have a similar number of instances. The different methods to get such a class balance can be classified in 3 main categories [19] by taking into account how do they get a similar number of instances: undersampling, oversampling, and hybrid methods. The methods belonging to each one of these categories that have been applied in present research are:

- **Undersampling methods**: their strategy to get a balanced number of instances per class is creating a new subset by removing some instances. Usually, the data instances to be removed are from the majority classes, so their prominence is reduced in favor of minority classes. The most common and widely used method of undersampling is known as Random Under Sampling (RUS). It is a simple non-heuristic method that gets a class-balanced subset by randomly selecting those instances to be deleted.

- **Oversampling methods**: their strategy to get a balanced number of instances per class is creating a superset by artificially generating data instances. Usually, these new instances are from the minority classes, so their prominence is increased. As in the case of undersampling, there is a common and widely used method of oversampling, known as Random Over Sampling (ROS) that randomly selects the data instances to be duplicated. A more advanced oversampling method is Synthetic Minority Oversampling TEchnique (SMOTE) [31]. It introduces synthetic data samples created by interpolating different minority-class instances. In order to select these reference instances, k-Nearest Neighbors (KNN) algorithm is applied, as graphically explained in Figure 1. SMOTE initially selects an x_i minority class instance as a basis for creating new instances of the minority class. By considering Euclidean distance, multiple data samples (Nearest Neighbors) of the minority class (points from x_{i1} to x_{i4}) are chosen from the dataset. Finally, an interpolation is performed in order to obtain new instances ranging from r_1 to r_4.

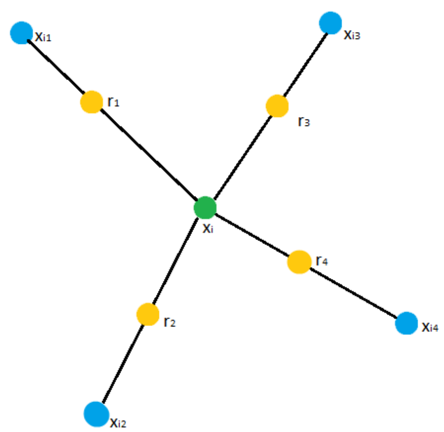

Figure 1. Syntethic data creation with SMOTE. Adapted from [19].

- **Hybrid methods**: they combine the use of oversampling and undersampling techniques in order to reduce the impact in only one of the classes that the single methods have. One of the hybrid methods that have been applied in present work is ROS + RUS, that combines the two simplest methods of data balancing, previously introduced. Additionally, in keeping with the idea in [31], SMOTE is combined with RUS, generating synthetic instances of the minority class while randomly eliminating instances of the majority one at the same time.

2.3. Performance Metrics

The performance of SVM when detecting the anomalies is validated through the standard metrics for supervised learning methods. These metrics are described in this subsection and are calculated from the figures associated to a one-class classification. These figures are the ones usually presented in a confusion matrix, that in anomaly detection are:

- False Positives (FP): normal data that are mistakenly classified as anomalous.
- False Negatives (FN): anomalies that are mistakenly classified as normal data.
- True Positives (TP): anomalies that are correctly classified as such.
- True Negatives (TN): normal data that are correctly classified as such.

Based on these basic statistics that have been previously defined, some useful metrics can be calculated:

2.3.1. Accuracy

It can be seen as the global hit ratio, without taking into account whether the data is anomalous or not. It is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

2.3.2. Precision

It is designed to reflect the proportion of data that the given classifier successfully labels as anomalous. This proportion is calculated by taking into account the total number of data labelled as anomalous (TP + FP). It is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

2.3.3. True Positive Rate (TPR)

This metric, also known as Recall, focuses on the relevant data of the problem that, in anomaly detection, are the anomalies. It is similar to Precision but the proportion is now calculated by taking into account the total number of truly anomalous data (TP + FN). It is defined as:

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

2.3.4. False Positive Rate (FPR)

This metric reflects the proportion of "normal" data that is mistakenly classified as anomalous. This proportion is calculated by taking into account the total number of "normal" data (FP + TN):

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

2.3.5. F_1 Score

As there are strong dependencies between some of the metrics that has been introduced so far, a new one was conceived in order to reflect a balance between the different aspects to be evaluated. This is the reason to introduce the F_1 Score, that is defined as:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

2.3.6. ROC Curve

The well-known Receiver Operating Characteristic (ROC) curve confronts TPR with FPR in a probabilistic way. It is used to depict the performance of a classifier in a 2D representation. Thus, it supports easily finding the best operating point in order to balance the two metrics (TPR and FPR). Based on this curve, the most important metric for present paper is calculated: the area under the curve (AUC) [32]. Although the previous metrics have also been calculated in all the experiments conducted in present study, it is AUC the top one as it is fair for evaluating classification results on imbalance datasets [17] [33] and it was the one used by authors of the dataset under analysis [27]. As AUC is calculated as a portion of the area of the perfect classification (unit square) it takes values in the range [0, 1]. As a result, the closer the AUC value is to 1, the better.

3. Real-life Case Study

As previously stated, present work addresses the detection of performance anomalies in the middleware of a component-based robot. It is done by analyzing a dataset [7] that was generated by researchers from the Bielefeld University (Germany) and is available at [26]. Data were recorded from the *ToBi* robot, whose base is PatrolBot, built upon the research platform GuiaBot, by MobileRobots. As a participant in the RoboCup@Home competition in 2015, its mission was to carry out different tasks related to a waiter's job, such as recognizing clients, asking them about the drink or serving. To complete these tasks, the robot has different components such as two RGBD cameras for person/object recognition, an arm for manipulating objects, and a speech recognition sensor, among others. Through a message-oriented, event-based middleware called Robotics Service Bus (RSB) [34], all the robot components are connected. Data from this RSB associated to different system executions have been captured at runtime thanks to a tool called *rsbag*.

For the induction of anomalies, the authors of the dataset firstly surveyed researchers, university students, and workers through a questionnaire. As a result, the most usual software anomalies for the platform were identified. Then, the anomalies were induced in *ToBi* and were activated through RSB middleware in order to know the precise moment they were produced. 11 anomalies were induced and are present in the dataset, namely: armServerAlgo, legDetectorSkippable, objectBuilderSkippable, clafuSleep, pocketSphinxLeak, btlAngleAlgo, bonsaiParticipantLeak, bonsaiTalkTimeout, facerecSkippable, clockShift and SpreadLatency. Differentiating from previous work [6], where only one of them (armServerAlgo) was addressed, present paper addresses 9 of them. In order to set a common criteria, anomalies affecting more than one robot component (spreadLatency and clockshift) have been discarded for a fair comparison. Those anomalies analyzed in present work are shown in Table 1. For the sake of brevity and clarity, a code has been assigned to each one of them, as shown in the first column.

Table 1. Selected anomalies to be analyzed.

Code	Name	Description
A1	armServerAlgo	Certain movements of the arm are performed from known valid poses
A2	legDetectorSkippable	The ‘legdetector’ processed each scan multiple times
A3	objectbuilderSkippable	The person tracking performed transformations for each person multiple times
A4	clafuSleep	The results are returned only after a delay of 5 seconds
A5	pocketSphinxLeak	The speech recognition component accumulates memory for each sound
A6	btlAngleAlgo	Adds a mathematical error used to track people
A7	bonsaiParticipantLeak	Participants are not cleaned up properly
A8	bonsaiTalkTimeout	Configuring a wrong RSB scope for the text-to-speech engine
A9	fecrecSkippable	Temporarily removes a throttling of the main loop of the ‘facerec’ component

The dataset comprises 71 trials, being each one of them an attempt of the robot to perform some of the tasks. Not all trials are included in present work; analyzed trials are those that do not have undetected faults and that are considered valid by the dataset authors. Each trial comprises a file linked to each component of the robot and the different data associated to it. These data refer to the interaction of the robot with its environment during a given time interval and duration varies between trials. Each data instance from the dataset is a sampling of the different data sources at a certain time. Two data sources are used to get information about the performance of the robot software: Features and Counters. The authors of the dataset provided a third set, events, whose information is found in Features. It contains information about the relevant events that occurred in the component, including the size of sending and receiving information. On the other hand, Counters are the raw export of the performance counters for the component, whereas Features are a combination of performance counters and events with the timing of the counters. Further details on data sources are available at [26].

The dataset has the following structure, as described in Figure 2: the information of each one of the 71 trials is available. There is the information gathered from the different components and (in a faults file) all the induced anomalies, with the starting and ending time as well as its type. For each one of the components, the three sources of data (Features, Counters and Events) are available. Additionally, there is another file that indicates if there is an anomaly induced in this component and which is the affected time frame.

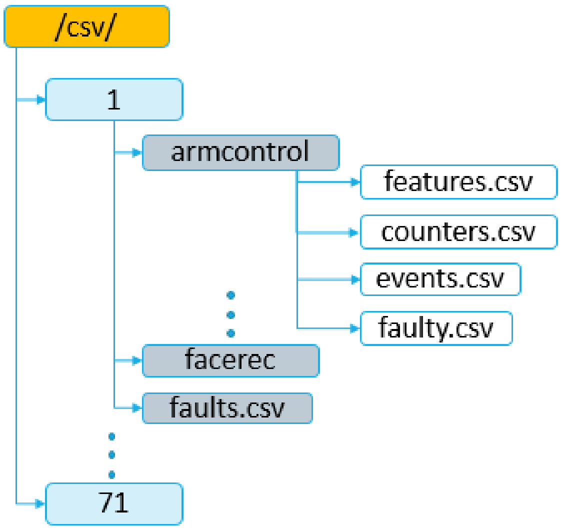

Figure 2. Structure of the dataset under analysis.

Table 2. Occurrences of each anomaly and distribution per trials. In bold, the trials selected for the one-trial experiments.

Anomaly	1 time	2 times	3 times	4 times
A1	28, 32, 36, 41, 45, 57, 65, 71	21	**23**	
A2	19, 21, 24, 31, 36, 55, 57, 64, 66, 68, 70		**71**	
A3	20, 38, 41, 42, 54, 55, 57, 64, 65, 66	18	**63**	
A4	21, 27, 32, 41, 42, 51, 55, 56	49, 63, **65**, 66		
A5	28, 29, 31, 32, 35, 37, 38, 41, 45, 68			**69**
A6	19, 21, 39, 70, 71	29, 37, **45**		
A7	19, 20, 32, 35, 51, 56, 64	**24**, 68, 70		
A8	23, 24, 28, 35, 36, 38, 39, 42, 45, 49, 56	**18**, 51		
A9	19, 20, 27, 29, 31, 32, 36, 42, 49, 51, **54**, 64			

In order to evaluate the impact of the proposed strategies, experiments were conducted comprising only one trial and all the trials containing examples of a certain anomaly. The underlying idea was to know if the performance of the applied classifier will significantly vary depending on the amount of trials to be considered and the balancing of them. The number of occurrences (from 1 to 4) of each anomaly and the related trials are shown in Table 2. In bold there can be identified those trials that have been selected for the one-trial experiments. Among all the trials containing examples of an anomaly, those that have a greatest relevance for each one of them were selected individually. The selection criteria of the trial has been firstly the one with the highest number of anomaly occurrences. Secondly, for those trials with the same number of occurrences, it has been selected the one with the longest period of time between each two anomalies. The main reason is to maximize the time between occurrences in order to let the robot recover from the first occurrence.

Table 3. Missing values in the dataset per anomaly and data source, with its percentage to total values.

Anomaly	Features	Counters	Features + Counters
A1	366166 (20.4%)	26025 (8.5%)	392191 (19.06%)
A2	17447(4.43%)	24163 (10.03%)	41610 (7.04%)
A3	16492 (3.59%)	23206 (11.78%)	39698 (6.48%)
A4	66308 (11.65%)	24630 (8.04%)	90938 (10.93%)
A5	67884 (6.74%)	25264 (8.25%)	93148 (7.33%)
A6	469662 (9.17%)	19308 (5.88%)	488970 (9.05%)
A7	469662 (9.17%)	19308 (5.88%)	488970 (9.05%)
A8	469662 (9.17%)	19308 (5.88%)	488970 (9.05%)
A9	175074 (21.62%)	48109 (14.65%)	223183 (20.39%)

Table 4. Class distribution of data per anomaly and trial in the dataset.

Anomaly	All Trials			One Trial		
	Normal Class	Anomaly Class	Anomaly Percentage	Normal Class	Anomaly Class	Anomaly Percentage
A1	20832	1055	5.06%	462	233	50.43%
A2	20765	1127	5.43%	439	206	46.92%
A3	20515	1375	6.70%	445	209	46.97%
A4	20547	1345	6.55%	427	160	37.47%
A5	20738	1147	5.53%	316	320	101.27%
A6	20934	951	4.54%	553	186	33.63%
A7	20837	1048	5.03%	522	160	30.65%
A8	20685	1200	5.80%	554	160	28.88%
A9	20847	1036	4.97%	500	88	17.60%

Since data irregularities are important in present work, some figures about them are provided. Firstly, the total amount of MV in all the trials affected by each anomaly are shown in Table 3 per anomaly and data source. As it can be seen from this table, the amount of MV significantly varies from one anomaly to the other ones. It is worth mentioning the case of A6, A7, and A8 anomalies as MV amount to 488,970 in all of them, when considering both Features and Counters. This is because they all affect the same robot component, namely statemachine. A study has been conducted to know whether the presence of MV adjusts to a given pattern, but no relevant evidences have been found.

Additionally, it can be observed in Table 4 the distribution of data in the two classes (normal/anomaly) for the different anomalies and the percentage of data from the minority class (anomalous). These figures are calculated by setting a MV ratio of zero and this is the reason why the number of instances is the highest one. In the right part of this Table 4, it can be seen the data distribution in classes for the individual trials. As indicated, these subsets of the data are much more balanced (higher percentage of anomalies) that in the case of the whole dataset (all trials). Furthermore, in the case of A5 anomaly, there are few more instances of the anomaly class than that of the normal class. Then, the effect of applying the pre-processing techniques is checked for similar data in slightly and strongly unbalanced datasets.

Finally, figures about the size of the different datasets are provided in Table 5. The row-wise datasets are presented per anomaly and data-source and the number of both rows and columns are included. The applied MV ratio (varying from 0% to 50%) is indicated in the case of all trials and in the case of the one trial, there is only one value associated to the 0% MV ratio.

Table 5. Size of the different datasets per anomaly and data source.

		Rows					Columns				
		0	0.1	0.25	0.5	1 Trial	0	0.1	0.25	0.5	1 Trial
Features	A1	21887	20591	14429	7350	695	35	47	58	68	77
	A2	21892	21892	21892	21892	645	17	17	17	17	17
	A3	21890	21890	21890	21890	654	20	20	20	20	20
	A4	21892	21892	21892	21892	587	22	22	22	22	22
	A5	21885	21885	17175	17175	636	42	42	43	43	43
	A6	21885	18608	5772	-	739	175	189	207	-	213
	A7	21885	18608	5772	2583	682	175	189	207	212	214
	A8	21885	18608	5772	2583	714	175	189	207	212	214
	A9	21883	21883	21883	21883	588	28	28	28	28	29
		Rows					Columns				
		0	0.1	0.25	0.5	1 Trial	0	0.1	0.25	0.5	1 Trial
Counters	A1	21887	21887	18530	18350	695	11	11	13	13	11
	A2	21892	21892	18534	18534	645	8	8	10	10	10
	A3	21890	21890	18533	18533	654	6	6	8	8	8
	A4	21892	21892	18534	18534	587	11	11	13	13	13
	A5	21885	21885	17175	17175	636	12	12	13	13	13
	A6	21885	21885	21885	-	739	14	14	14	-	14
	A7	21885	21885	21885	21885	682	14	14	14	14	14
	A8	21885	21885	21885	21885	714	14	14	14	14	14
	A9	21883	21883	21883	21883	588	12	12	12	12	13
		Rows					Columns				
		0	0.1	0.25	0.5	1 Trial	0	0.1	0.25	0.5	1 Trial
Features + Counters	A1	21887	20591	11762	5974	695	44	56	69	79	86
	A2	21982	21892	18534	18534	645	23	23	25	25	25
	A3	21890	21890	18533	18533	654	24	24	26	26	26
	A4	21982	21892	18534	18534	587	31	31	33	33	33
	A5	21885	21885	17175	17175	636	52	52	54	54	54
	A6	21885	18608	5772	-	739	187	201	219	-	225
	A7	21885	18608	5772	2583	682	187	201	219	224	226
	A8	21885	18608	5772	2583	714	187	201	219	224	226
	A9	21883	21883	21883	21883	588	38	38	38	38	40

4. Experiments and Results

In this section, the results obtained after the execution of the different experiments are shown. 30 of them have been carried out for each one of the anomaly, which amount to 270 experiments in total (9 anomalies are studied). They have been conducted on different subsets of the original dataset (see section 3): on the one hand, the most significant trial for each one of the anomalies (see Table 2) has been analyzed (results in section 4.1) while all the trials have been also analyzed (results in section 4.2). The best results are presented in each case, regardless the data source (Features, Counters, and both of them). At the end of the section, results associated to the different data sources are presented (see Figure 5) and discussed.

For each one of these subsets of data, several experiments have been performed with different combinations of the data-preprocessing methods explained in section 2.2. With regard to the MV issue, different values of the previously explained MV ratio have been applied when analyzing all trials. Once MV had been removed, different experiments have been carried out with a great variety of data balancing methods, namely ROS, RUS, both at the same time (ROS + RUS), SMOTE, and SMOTE with RUS. All in all, one undersampling, two oversampling, and two hybrid methods have been applied for data balancing. Additionally, the obtained performance results are also compared with that for the originally imbalanced dataset without applying any of data-balancing method (referenced as “None”).

All the results presented in this section have been obtained by training a SVM (see section 2.1) on 75% of the available data while validating on the 25% remaining data. For the validation, the well-known technique of k-fold Cross Validation (with the value $k = 10$) has been applied. Additionally, 10 executions have been carried out per each experiment in order to obtain more statistically significant results. Average results for these 10 executions are shown in present section.

In order to validate the results obtained with the different models and datasets, the non-parametric Wilcoxon Signed-Ranks Test [35] [36] has been used, as it suits present work (compare different models on different dataset without a-priori assumptions). This test supports selecting the best methods on the varied situations under analysis (different datasets and combinations of methods).

4.1. One-Trial Experiments

As previously mentioned, experiments were carried out on one single trial per anomaly (that containing most examples of the anomaly as stated in Table 2). The same pre-processing methods have been applied in the one-trial and all-trial experiments. However, in the one-trial experiments only one MV ratio was tested: 0%. The reason for that is that none of the selected trials contains MV.

Obtained metrics, calculated on the different data and methods, are shown in the following tables. In Table 6 it is shown the obtained F_1 values, while the AUC ones are shown in Table 7.

Table 6. Obtained F_1 values per anomaly and data-balancing method in the one-trial experiments.

	None	ROS	SMOTE	RUS	ROS + RUS	SMOTE + RUS
A1	0.3194	0.4128	0.4140	0.4229	0.4135	**0.4242**
A2	0.9415	0.9387	0.9492	**0.9424**	0.9397	0.9362
A3	0.8187	0.8226	0.7918	**0.8324**	0.8194	0.8061
A4	0.5486	0.5373	**0.5755**	0.5529	0.5656	0.5595
A5	0.6888	0.7149	**0.7152**	0.6981	0.6907	0.6805
A6	0.4680	0.5995	0.5837	0.6388	0.6114	**0.6628**
A7	0.5853	**0.5881**	0.5532	0.5789	0.5796	0.5396
A8	0.1296	0.2985	0.2774	0.2954	0.2825	**0.3225**
A9	0.7932	**0.9034**	0.8320	0.8093	0.8339	0.8909

When analyzing the F_1-score metric (Table 6), it can be clearly seen that the application of balancing techniques has greatly improved the obtained values. SMOTE (rather applied in isolation or in connection with RUS) has got

Table 7. Obtained AUC values per anomaly and data-balancing method in the one-trial experiments.

	None	ROS	SMOTE	RUS	ROS + RUS	SMOTE + RUS
A1	0.5354	0.5592	0.5712	**0.5808**	0.5515	0.5737
A2	0.9588	0.9622	**0.9656**	0.9646	0.9582	0.9558
A3	0.8738	0.8796	0.8524	**0.8806**	0.8800	0.8656
A4	0.6926	0.6904	**0.7084**	0.6916	0.7040	0.7061
A5	0.7019	**0.7188**	0.7086	0.6969	0.6986	0.6857
A6	0.6491	0.7584	0.7390	0.7781	0.7571	**0.7903**
A7	0.7353	0.7464	0.7277	**0.7544**	0.7444	0.7188
A8	0.4432	0.5828	**0.5894**	0.5762	0.5885	0.5486
A9	0.8878	**0.9594**	0.9414	0.9546	0.9360	0.9540

the highest F_1 values in 5 cases (out of 9). On the other hand, none of the highest values has been obtained with the original dataset (no balancing method applied). Results are greatly improved in the case of the A8 anomaly, varying from 0.1296 (no balancing method) to 0.3225 (obtained with SMOTE + RUS).

The AUC scores (Table 7) are also provided. As it is the key metric (see section 2.3), a bar graph (see Figure 3) has been generated in order to ease the comparison of results. Thus, the best AUC scores for each one of the anomalies obtained by each one of the applied methods can be observed. Additionally, the results of statistical test for these two metrics are shown in Tables 8 and 9.

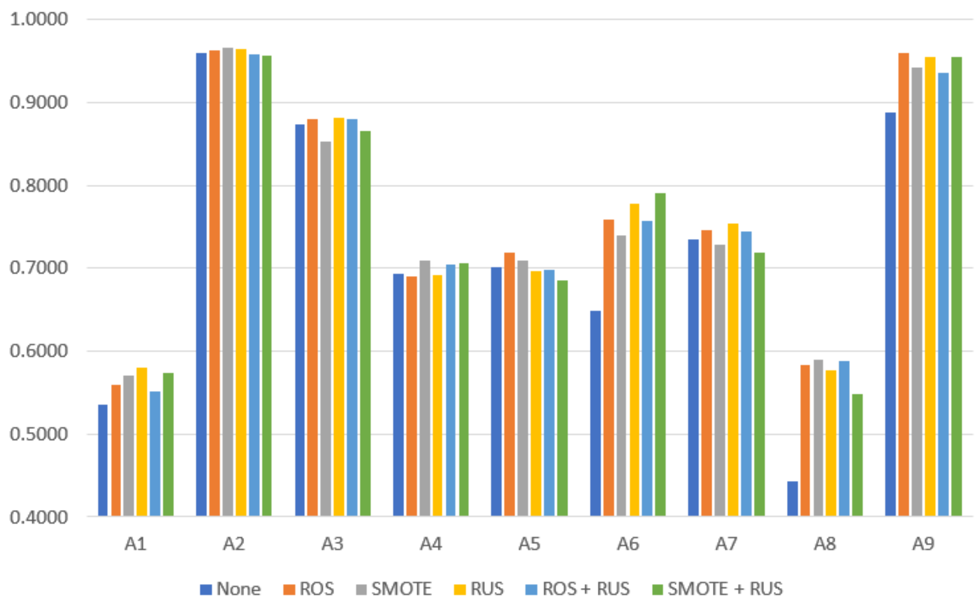

Figure 3. AUC values per anomaly in the one-trial experiments.

When considering these results, some facts are worth highlighting:

- Analysis by anomaly: AUC values strongly vary from some anomalies to the other ones. While in the case of A1 and A8 anomalies the obtained values are close to 0.5, for most of the anomalies they are in the range [0.6 - 0.9]. In the case of A2 and A9, values higher than 0.9 have been obtained. The case of the A5 anomaly is worth mentioning; it is the only balanced anomaly and as a result, AUC scores are not improved with the balancing techniques. Actually, worst results have been obtained with all the undersampling combinations (RUS, ROS + RUS, and SMOTE + RUS) than that obtained with the original data.

- Analysis by balancing method: when comparing the results obtained with the balancing methods and those obtained from the original dataset, similar values have been obtained except in the anomalies A6, A8, and A9. For these three anomalies, the AUC values from the original dataset are much lower than those obtained with any of the balancing methods. As it can be seen in the Table 4, these anomalies are three of the most unbalanced ones, except for the A7 anomaly. Going into more detail it has been observed that Features and Features + Counters follow this same pattern but it is not followed in the case of Counters itself. This trial has fewer instances than the other two anomalies with the same component, A6 and A8. In all anomalies, the best AUC scores are obtained with a balancing method and not with the original data. However, there is not a balancing method that gets better AUC values in all anomalies.

As previously stated, the Wilcoxon Signed-Ranked Test has been carried out to get more statistically significant conclusions from the one-trial experiments. It has been applied to the obtained values for the F_1 score (Table 8) and AUC (Table 9) metrics.

Table 8. p-values obtained by the non-parametric Wilcoxon Signed-Ranked Test pairwise on the one-trial experiments per balancing method for the F_1 values.

	None	ROS	SMOTE	RUS	ROS + RUS	SMOTE + RUS
None	-	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2
ROS	0.0546	-	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2
SMOTE	0.1132	≥ 0.2	-	≥ 0.2	≥ 0.2	≥ 0.2
RUS	0.0195	≥ 0.2	≥ 0.2	-	≥ 0.2	≥ 0.2
ROS + RUS	0.0546	≥ 0.2	≥ 0.2	≥ 0.2	-	≥ 0.2
SMOTE + RUS	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	-

Table 9. p-values obtained by the non-parametric Wilcoxon Signed-Ranked Test pairwise on the one-trial experiments per balancing method for the AUC values.

	None	ROS	SMOTE	RUS	ROS + RUS	SMOTE + RUS
None	-	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2
ROS	0.0078	-	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2
SMOTE	0.0976	≥ 0.2	-	≥ 0.2	≥ 0.2	≥ 0.2
RUS	0.0195	≥ 0.2	≥ 0.2	-	≥ 0.2	≥ 0.2
ROS + RUS	0.0195	≥ 0.2	≥ 0.2	≥ 0.2	-	≥ 0.2
SMOTE + RUS	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	-

According to the values in the Table 8 (F_1 values) and the values of R^+ and R^-, H_0 for None is not rejected just in the cases of SMOTE and SMOTE + RUS. For all the other balancing methods, it can be said that SMOTE and SMOTE + RUS do not outperform None. The same happens when considering both the R^+ and R^- values and p-values (Table 9) related to AUC. As a result, it can be concluded that in the case of one-trial experiments, ROS, RUS, and ROS + RUS outperform all the other methods (None included) but any of them can be identified as the best one in all cases.

4.2. All-Trials Experiments

As it has been previously explained, same experiments have been run on data subsets containing all the trials for each one of the anomalies. That is, data from all the trials listed in each row of Table 2 have been merged. As a result, highly unbalanced datasets have been generated, as can be seen in Table 4.

Differentiating from the experiments on the one-trial datasets, for the all-trials ones 4 different MV ratios have been considered: 0%, 10%, 25%, and 50%. Obviously, with a 0% MV rate, the datasets with the highest number of instances and the lowest number of features (the ones in Table 4) have been obtained. For the sake of brevity, in the case of all-trials datasets, only best AUC scores for each combination of parameters are shown. Obtained values are compiled in Table 10 for the MV ratios of 0%, 10%, 25%, and 50% respectively, and all data sources.

From the obtained results, it can be stated that the worst performance has been obtained with the original datasets (no balancing method) for all the anomalies. On the other hand, from these results no clear conclusion can be drawn as to which balancing method works best when stating a 0% MV ratio. With this MV threshold, SMOTE and RUS stand out, being the best ones for 3 of the anomalies each, while the combination of them (SMOTE + RUS) has not obtained the best result for any of the anomalies.

When increasing the MV ratio to 10% (see second section of Table 10), it can be observed with greater clarity that oversampling methods (ROS and SMOTE) stand out from the rest of balancing methods and the original datasets. Actually, the highest AUC value (0.9822) of all the performed experiments in present work has been obtained with the 10% MV ratio and ROS balancing method for the A2 anomaly. The AUC values for the A9 anomaly are very similar to those with the 0% MV ratio (they only vary for the undersampling methods) because there is no change in the number of MV and hence instances. That is, there is not any original feature containing less than 10% MV. In the case of the 10% MV ratio, it is ROS that stands out (best one for 4 anomalies). Once again, the combination of SMOTE and RUS (SMOTE + RUS) has never obtained the best results, as it has happened with the original data.

From the results with a 25% MV ratio (see third section Table 10), as opposed to what was pointed our for the 10% MV ratio, the RUS method stands out from the other ones (best results in 5 out of 9 anomalies). As an exception, it has got the worst result of a balancing method in the case of the A6 anomaly. When combined with the SMOTE method (SMOTE + RUS) it has also obtained the highest value for the A1 anomaly. For this anomaly, this combination is the only method that has improved the AUC values when compared with that obtained from the original (unbalanced) data. SMOTE is the second best method, obtaining the highest AUC scores in 2 anomalies. The hybridization of ROS + RUS has not obtained the best value for any anomaly.

Finally, results obtained when applying a 50% MV ratio (see last section Table 10) are discussed. No results are available for the A6 anomaly because when pre-processing it with that threshold of MV, many instances are eliminated. It causes that none from the anomaly class is kept and hence all the ones remaining in the dataset belong are from the normal class. For the other anomalies, it can be pointed out that, as it happened for the 25% MV ratio, RUS outperforms the other methods, being the best one 4 times. SMOTE has obtained the best results 3 times and the hybrid methods have not obtained the highest AUC value for any anomaly.

Obtained p-values when applying the Wilcoxon Signed-Ranked Test are calculated per balancing method (Table 11) and MV ratio (Table 12). In the case of the balancing methods, the null hypothesis is rejected in all cases for None. Thus, we can conclude that all classifiers obtain a better rank than None. On the other hand, when comparing all the balancing methods, none of them can be designated as the best one for all the datasets in present study as there are not statistical differences.

When analyzing the results per MV ratio, in none of the cases the null hypothesis is rejected, either because of the R or p-value scores. As a result, it can be said that there are no significant statistical differences between the different MV ratios.

As in the case of the one-trial experiments, a bar plot has been generated showing the obtained AUC values for each anomaly, balancing method and MV ratio. In order to also contribute to easily interpret the obtained AUC results, several boxplots have been generated. They are presented, summarizing information according to different criteria: balancing method and MV ratio (Figure 4), data sources (Figure 5 a), b), and c)), and anomalies (Figure 5 d)).

From all these results and figures, it can be seen that the variance of AUC values is pretty similar for the results obtained with all the balancing methods. On the other hand, the median values are similar as well; those obtained by SMOTE and RUS are the two highest ones. Paradoxically, it is the combination of these two the one that has obtained the lowest variance.

Table 10. Obtained AUC values per anomaly and data-balancing method. All-trial experiments with 0%, 10%, 25%, and 50% MV ratio.

		None	ROS	SMOTE	RUS	ROS + RUS	SMOTE + RUS
0%	A1	0.5255	0.5747	**0.5841**	0.5639	0.5798	0.5574
	A2	0.8855	0.9805	0.9797	0.9785	**0.9821**	0.9798
	A3	0.7268	0.7842	0.7765	**0.7919**	0.7700	0.7786
	A4	0.5067	0.5704	0.5667	0.5603	**0.5722**	0.5697
	A5	0.4993	0.6945	**0.7053**	0.6994	0.6968	0.6857
	A6	0.5041	**0.5354**	0.5149	0.5144	0.5202	0.5263
	A7	0.5946	0.6770	0.6837	**0.6871**	0.6842	0.6709
	A8	0.4879	0.5225	0.5424	**0.5479**	0.5319	0.5358
	A9	0.6557	0.9059	**0.9110**	0.9098	0.9054	0.9097
10%	A1	0.5155	**0.5531**	0.5368	0.5480	0.5415	0.5225
	A2	0.8855	**0.9822**	0.9797	0.9788	0.9810	0.9814
	A3	0.7268	0.7774	0.7765	**0.7947**	0.7852	0.7719
	A4	0.5067	**0.5775**	0.5667	0.5636	0.5683	0.5740
	A5	0.4993	0.6945	**0.7053**	0.7050	0.6927	0.7013
	A6	0.5067	0.5158	0.5226	0.5183	**0.5260**	0.5243
	A7	0.5701	**0.6806**	0.6712	0.6803	0.6751	0.6796
	A8	0.4884	0.5196	**0.5344**	0.5334	0.5329	0.5315
	A9	0.6557	0.9059	0.9110	**0.9210**	0.9104	0.9086
25%	A1	0.5423	0.5309	0.5289	0.5322	0.5118	**0.5524**
	A2	0.8719	0.9717	0.9702	**0.9745**	0.9702	0.9709
	A3	0.7056	0.7726	0.7711	**0.7928**	0.7835	0.7646
	A4	0.4874	**0.5765**	0.5701	0.5700	0.5693	0.5744
	A5	0.5082	0.6917	**0.7079**	0.6993	0.6891	0.6924
	A6	0.5063	0.7891	**0.8010**	0.7213	0.7704	0.7866
	A7	0.4962	0.6776	0.6744	**0.6837**	0.6823	0.6811
	A8	0.4726	0.5257	0.5386	**0.5825**	0.5438	0.5344
	A9	0.6557	0.9059	0.9051	**0.9210**	0.9104	0.9086
50%	A1	0.5379	0.5670	**0.5699**	0.5646	0.5479	0.5516
	A2	0.8719	0.9717	0.9702	**0.9745**	0.9702	0.9709
	A3	0.7056	0.7726	0.7711	**0.7928**	0.7835	0.7646
	A4	0.4874	**0.5765**	0.5701	0.5700	0.5693	0.5744
	A5	0.5082	0.6917	**0.7079**	0.6993	0.6891	0.6924
	A6	-	-	-	-	-	-
	A7	0.4967	0.9090	**0.9178**	0.7015	0.8458	0.8859
	A8	0.4784	0.6417	0.6472	**0.6671**	0.6439	0.6453
	A9	0.6557	0.9083	0.9095	**0.9161**	0.9159	0.9085

In Figure 4 it can be seen the effect of modifying the MV ratio. The median value is higher with a 50% Missing Values ratio. On this same boxplot it is seen as the third and the first quartile are considerably higher than on the rest, especially in the case of the third quartile. The general trend is that at a higher percentage of Missing Values in the data set both the first, the third quartile and the median increase. However, in the case of 10% of Missing Values the third quartile and the median reach values similar to 0% and on the other hand the first quartile gets a slightly smaller value.

Table 11. p-values obtained by the non-parametric Wilcoxon Signed-Ranked Test pairwise on the all-trials experiments per balancing method for the AUC values.

	None	ROS	SMOTE	RUS	ROS + RUS	SMOTE + RUS
None	-	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2
ROS	0.0039	-	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2
SMOTE	0.0039	0.1641	-	≥ 0.2	≥ 0.2	≥ 0.2
RUS	0.0039	≥ 0.2	≥ 0.2	-	≥ 0.2	≥ 0.2
ROS + RUS	0.0039	≥ 0.2	≥ 0.2	≥ 0.2	-	≥ 0.2
SMOTE + RUS	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	≥ 0.2	-

Table 12. p-values obtained by the non-parametric Wilcoxon Signed-Ranked Test pairwise on the all-trials experiments per MV ratio for the AUC values.

	0%	10%	25%	50%
0%	-	≥ 0.2	≥ 0.2	≥ 0.2
10%	≥ 0.2	-	≥ 0.2	≥ 0.2
25%	≥ 0.2	≥ 0.2	-	≥ 0.2
50%	≥ 0.2	≥ 0.2	≥ 0.2	-

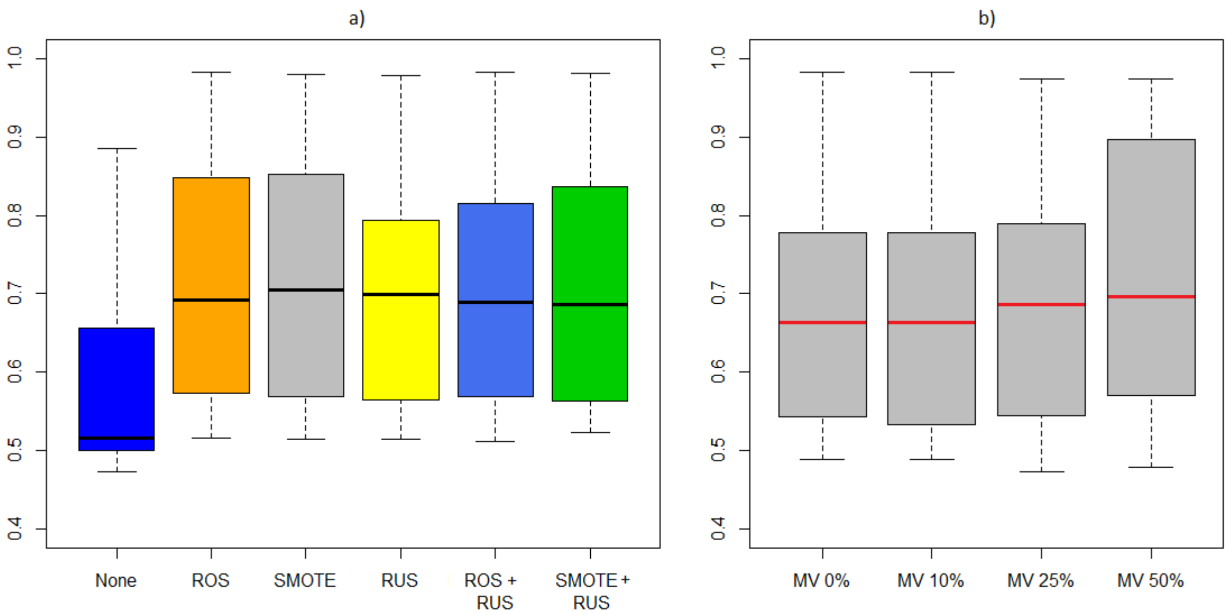

Figure 4. Boxplot of the obtained AUC values in the all-trials experiments: a) per balancing method and b) per MV ratio.

A continuity in the values obtained in the different data sources is observed in Figure 5, where the most different one from the other two is Counters. In the case of this data source, the variance is higher for the majority of anomalies. Additionally, the case of A6 anomaly is worth mentioning as the variance is drastically reduced for the Counters when compared to the other two data sources (Features and Features + Counters). After a thorough analysis, it has been

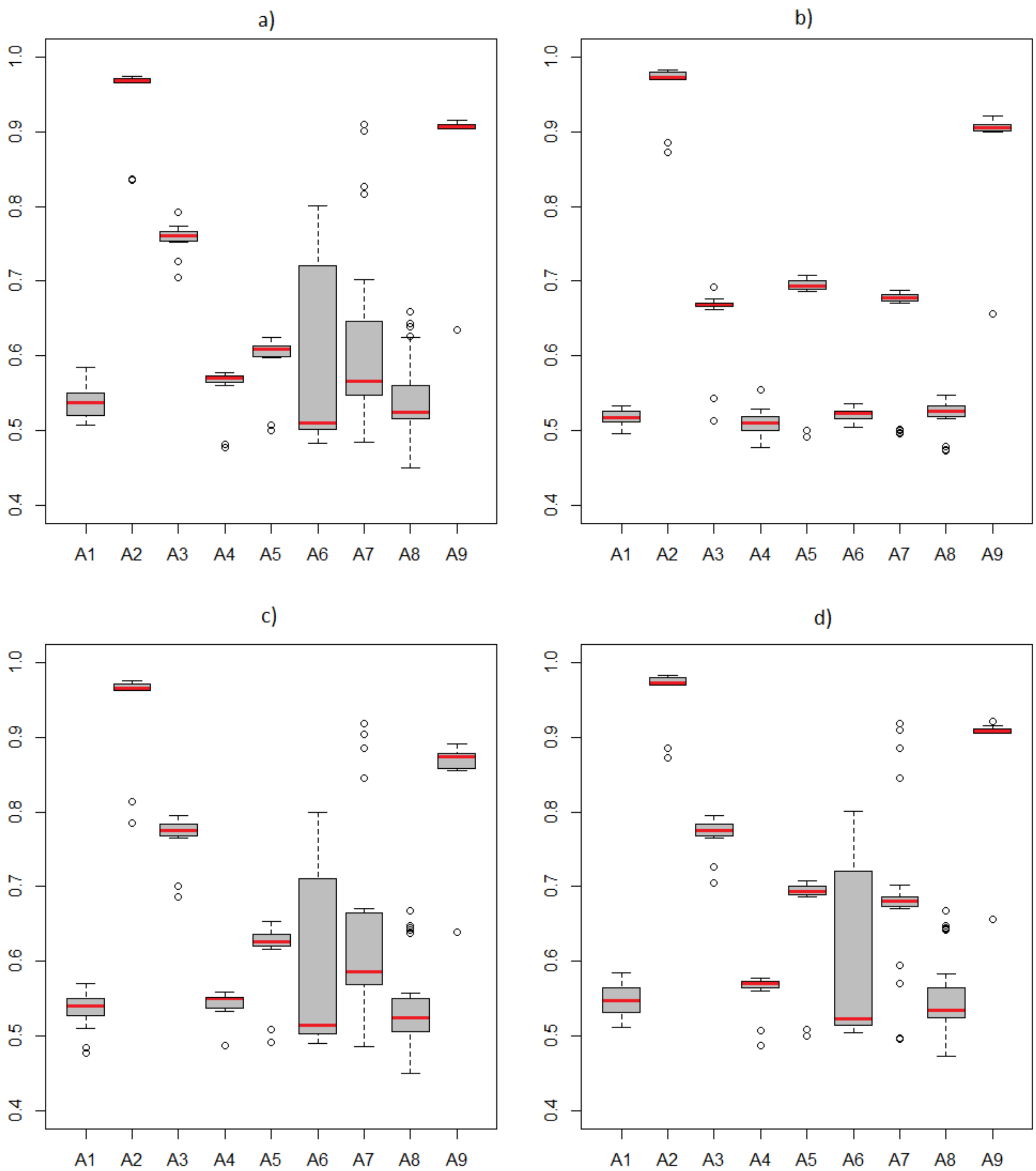

Figure 5. Boxplot of the obtained AUC values per anomaly in the all-trials experiments in each data source. a) Features, b) Counters, c) Features + Counters, and d) All data sources.

identified the results causing this phenomenon, that have been obtained with the 25% MV ratio (note that no results are available for this anomaly when applying the 50% MV ratio). It can be seen in the Table 5 that there is a big difference in the number of features between the 10% and 25% MV ratios in the case of the A6 anomaly for the Features and Features + Counters data sources.

Results are also discussed per anomaly, summarizing all the figures for the different MV ratios and balancing methods (boxplot d) in Figure 5). Very concentrated AUC values (comprising few "abnormal" ones) have been obtained for the A2, A3, A4, A5, A7, and A9 anomalies. Results associated to the A1 and A8 anomalies have a greater variance and lower median. The A6 anomaly is the one with the lowest median, very distant from the third quartile and close to the first quartile. This is mainly due to those results that has been previously highlighted from the last section of Table 10: with a 50% MV ratio, no result can be obtained given the absence of instances of the anomalous class.

When analyzing results obtained from the different data sources, the following ideas can be observed:

- Features: the most prominent algorithm is RUS (best one for 4 of the anomalies) followed by ROS and SMOTE (for 2 anomalies), whereas SMOTE + RUS ha attained the best results for 1 anomaly. With regard to the MV ratio, it should be noted that the same results for the A2 and A3 anomalies have been obtained with 50% and 25% values as the size of the dataset persists (see Table 5). The 50% MV ratio is the one associated to most best results (5 out of 9 anomalies).

- Counters: the method of balancing with best results has been ROS (for 4 anomalies), followed by RUS (for 3 anomalies) and SMOTE (for 2 anomalies). There were 3 anomalies attaining the same best results with 2 different MV ratios: A3 with 0% and 10%, A5 with 25% and 50%, and A9 with 10% and 25%. As previously mentioned in the case of Features, the datasets are the same in all cases. What is different in this case is that the 0% MV ratio is associated to best results for 6 of the anomalies.

- Features + Counters: when combining the two data sources, the methods obtaining best results are RUS and SMOTE (for 4 anomalies each) while the combination of them (SMOTE + RUS) is the best one for an anomaly. In this case, a MV ratio is not clearly the best one for any anomaly: 50% for 3 anomalies, 25% for 2 anomalies, 10% for 3 anomalies, and 0% for 1 anomaly.

To sum up, a brief summary of the individual results is presented: the SMOTE algorithm has outperformed all the other ones for 4 anomalies, RUS for 2 and ROS for 2 as well. It is worth highlighting the fact that none of the hybrid balancing techniques have achieved the best AUC result for any anomaly. When taking into account the MV ratio, the 0% value is associated to the best results in 1 occasion, 10% and 25% in 3 occasions and 50% in 2 occasions. Finally, each one of the data sources is associated to the best results for 3 of the anomalies. On the other hand, the worst results (from those presented in previous tables) have always been associated to the A8 anomaly, being the lowest one that with a 0.5 MV ratio and Features + Counters as data sources.

5. Conclusions and Future Work

In present paper, different alternatives for data preprocessing (management of MV and class-balancing methods) have been validated for the detection of anomalies in a component-based robotic system. Obtained results when training and testing the same learning model (SVM) are presented and compared in section 4 to validate the effect of pre-processing. All these figures have been obtained on a real-life and brand new dataset.

From the one-trial experiments it can be concluded that, as expected, the more balanced datasets are, the higher AUC values are obtained for the great majority of anomalies. However, in a real-life setting, anomalies are not frequent and unbalanced datasets are usual. Thus, experiments on more unbalanced and hence more real datasets have been also conducted. When analyzing the balancing methods by means of the statistical test, ROS and RUS, together with its combination are the ones that outperforms the no-balancing alternative. However, none of them can be clearly identified as the best one.

From the all-trials experiments when considering the balancing method, it must be highlighted the results obtained with SMOTE (grey box in Figure 4.a), which has stood out from the rest of methods. When individually analyzing the results (Table 10), RUS is the balancing technique obtaining the highest AUC rates in most occasions (40%). Complementary, from the point of view of the the MV ratio, it can be concluded that the least restrictive value (a 50% ratio) means better AUC values in general terms (box in the right side of Figure 4.b). As previously mentioned in section 4.2 when discussing the results in Table 4, the best results for the anomalies with a lower percentage of anomalous data (except the case of the A9) have been obtained with a high MV ratio (25% or 50%). For the other anomalies, best results have been obtained with the lowest ratios (0% and 10%).

As far as data sources are concerned, generally the best values are obtained with the combination of both (Features + Counters), whereas the highest median of values is the one associated to Counters. Results with Features greatly vary, depending on the MV ratio, while those with Counters are more constant.

All in all, it can be concluded that the proposed data-preprocessing techniques greatly contribute to increase the detection rate of anomalies, outperforming previous work [27]. However, the balancing method, MV ratio and data source must be carefully selected in each case as there is not a combination of them that outperforms the other ones in all cases, according to statistical tests.

Further work will be focused on covering all the anomalies and therefore analyzing those affecting more than one component. Additionally, due to the high number of features (columns) in the dataset, the application of feature selection techniques will also be explored. An alternative way of dealing with MV such as data imputation will also be considered, as well as benchmarking with additional learning models. The effect of the different strategies for handling data irregularities will also be considered for other supervised learning models in addition to SVM. Finally, some other alternatives for handling data imbalance (such as class weighting) will also be studied.

References

[1] F. Shrouf, J. Ordieres, G. Miragliotta, Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the internet of things paradigm, in: 2014 IEEE International Conference on Industrial Engineering and Engineering Management, 2014, pp. 697–701. doi:10.1109/IEEM.2014.7058728.

[2] P. K. Muhuri, A. K. Shukla, A. Abraham, Industry 4.0: A bibliometric analysis and detailed overview, Engineering Applications of Artificial Intelligence 78 (2019) 218 – 235. doi:10.1016/j.engappai.2018.11.007.

[3] M. Aiman Kamarul Bahrin, F. Othman, N. Hayati Nor Azli, M. Farihin Talib, Industry 4.0: A review on industrial automation and robotic, Jurnal Teknologi 78 (2016) 137–143. doi:10.11113/jt.v78.9285.

[4] IFR, Summary - OUTLOOK on World Robotics Report 2019 by IFR.
URL `https://ifr.org/ifr-press-releases/news/summary-outlook-on-world-robotics-report-2019-by-ifr`

[5] E. Khalastchi, M. Kalech, On fault detection and diagnosis in robotic systems, ACM Comput. Surv. 51 (1) (2018) 1–24. doi:10.1145/3146389.

[6] N. Basurto, Á. Herrero, Data selection to improve anomaly detection in a component-based robot, in: F. Martínez Álvarez, A. Troncoso Lora, J. A. Sáez Muñoz, H. Quintián, E. Corchado (Eds.), 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019), Springer International Publishing, Cham, 2020, pp. 241–250.

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

[8] X. Xu, H. Liu, M. Yao, Recent progress of anomaly detection, Complexity 2019. doi:10.1155/2019/2686378.

[9] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, N. F. Samatova, Anomaly detection in dynamic networks: a survey, Wiley Interdisciplinary Reviews: Computational Statistics 7 (3) (2015) 223–247. doi:10.1002/wics.1347.

[10] E. Jove, J.-L. Casteleiro-Roca, H. Quintián, J. A. Méndez-Pérez, J. L. Calvo-Rolle, A fault detection system based on unsupervised techniques for industrial control loops, Expert Systems 0 (0) (2019) e12395. doi:10.1111/exsy.12395.

[11] Á. Herrero, A. Jiménez, Improving the management of industrial and environmental enterprises by means of soft computing, Cybernetics and Systems 50 (1) (2019) 1–2.

[12] R. Malhotra, A systematic review of machine learning techniques for software fault prediction, Applied Soft Computing 27 (2015) 504 – 518. doi:10.1016/j.asoc.2014.11.023.

[13] T. P. Banerjee, S. Das, Multi-sensor data fusion using support vector machine for motor fault detection, Information Sciences 217 (2012) 96 – 107. doi:10.1016/j.ins.2012.06.016.

[14] S. Zidi, T. Moulahi, B. Alaya, Fault detection in wireless sensor networks through svm classifier, IEEE Sensors Journal 18 (1) (2018) 340–347. doi:10.1109/JSEN.2017.2771226.

[15] S. Das, S. Datta, B. B. Chaudhuri, Handling data irregularities in classification: Foundations, trends, and future challenges, Pattern Recognition 81 (2018) 674 – 693. doi:10.1016/j.patcog.2018.03.008.

[16] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, Pattern classification with missing data: a review, Neural Computing and Applications 19 (2) (2010) 263–282.

[17] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Information Sciences 250 (2013) 113 – 141. doi:10.1016/j.ins.2013.07.007.

[18] B. Twala, Robot execution failure prediction using incomplete data, in: 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2009, pp. 1518–1523. doi:10.1109/ROBIO.2009.5420900.

[19] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera, Data Level Preprocessing Methods, Springer International Publishing, Cham, 2018, Ch. 5, pp. 79–221.

[20] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284. doi:10.1109/TKDE.2008.239.

[21] V. Cerqueira, F. Pinto, C. Sá, C. Soares, Combining boosted trees with metafeature engineering for predictive maintenance, in: H. Boström, A. Knobbe, C. Soares, P. Papapetrou (Eds.), Advances in Intelligent Data Analysis XV, Springer International Publishing, Cham, 2016, pp. 393–397.

[22] M. Syafrudin, N. L. Fitriyani, G. Alfian, J. Rhee, An affordable fast early warning system for edge computing in assembly line, Applied Sciences 9 (1) (2018) 84–102. doi:10.3390/app9010084.
[23] P. Bergmeir, C. Nitsche, J. Nonnast, M. Bargende, Classifying component failures of a hybrid electric vehicle fleet based on load spectrum data, Neural Computing and Applications 27 (8) (2016) 2289–2304. doi:10.1007/s00521-015-2065-y.
[24] M. Luo, K. Wang, Z. Cai, A. Liu, Y. Li, C. F. Cheang, Using imbalanced triangle synthetic data for machine learning anomaly detection, Computers, Materials & Continua 58 (1) (2019) 15–26. doi:10.32604/cmc.2019.03708.
[25] D. Devi, S. K. Biswas, B. Purkayastha, Learning in presence of class imbalance and class overlapping by using one-class svm and undersampling technique, Connection Science 31 (2) (2019) 105–142. doi:10.1080/09540091.2018.1560394.
[26] J. Wienke, S. Wrede, A Fault Detection Data Set for Performance Bugs in Component-Based Robotic Systems. doi:10.4119/unibi/2900911. URL `https://doi.org/10.4119/unibi/2900911`
[27] J. Wienke, S. Wrede, Autonomous fault detection for performance bugs in component-based robotic systems, in: Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, IEEE, 2016, pp. 3291–3297. doi:0.1109/IROS.2016.7759507.
[28] J. Wienke, Framework-level resouce awareness in robotics and intelligent systems, PhD dissertation, Bielefeld University (2018). doi:10.4119/unibi/2932136.
[29] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297. doi:10.1007/BF00994018.
[30] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, ACM, New York, NY, USA, 1992, pp. 144–152. doi:10.1145/130385.130401.
[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357. doi:10.1613/jair.953.
[32] Jin Huang, C. X. Ling, Using auc and accuracy in evaluating learning algorithms, IEEE Transactions on Knowledge and Data Engineering 17 (3) (2005) 299–310. doi:10.1109/TKDE.2005.50.
[33] S. S. Mullick, S. Datta, S. G. Dhekane, S. Das, Appropriateness of performance indices for imbalanced data classification: An analysis, Pattern Recognition 102 (2020) 107197. doi:10.1016/J.PATCOG.2020.107197.
[34] J. Wienke, S. Wrede, A middleware for collaborative research in experimental robotics, in: 2011 IEEE/SICE International Symposium on System Integration (SII), 2011, pp. 1183–1190. doi:10.1109/SII.2011.6147617.
[35] E. A. Gehan, A generalized Wilcoxon test for comparing arbitrarily singly-censored samples*, Biometrika 52 (1-2) (1965) 203–224. doi:10.1093/biomet/52.1-2.203.
URL `https://doi.org/10.1093/biomet/52.1-2.203`
[36] G. Santafe, I. Inza, J. A. Lozano, Dealing with the evaluation of supervised classification algorithms, Artificial Intelligence Review 44 (1965) 467–508. doi:10.1007/s10462-015-9433-y.
URL `https://doi.org/10.1007/s10462-015-9433-y`