A Multi-phase Iterative Approach for Anomaly Detection and its Agnostic Evaluation

Kévin Ducharlet, Louise Travé-Massuyès, Marie-Véronique Le Lann, Youssef Miloudi

A Multi-phase Iterative Approach for Anomaly Detection and its Agnostic Evaluation*

Kévin Ducharlet[1], Louise Travé-Massuyès[2], Marie-Véronique Le Lann[2], and Youssef Miloudi[1]

[1] CARL Software - Berger-Levrault, Limonest, France
[firstname.lastname]@carl.eu
[2] LAAS-CNRS, University of Toulouse, CNRS, INSA, Toulouse, France
{louise,mvlelann}@laas.fr

Abstract. Data generated by sets of sensors can be used to perform predictive maintenance on industrial systems. However, these sensors may suffer faults that corrupt the data. Because the knowledge of sensor faults is usually not available for training, it is necessary to develop an agnostic method to learn and detect these faults. According to these industrial requirements, the contribution of this paper is twofold: 1) an unsupervised method based on the successive application of specialized anomaly detection methods; 2) an agnostic evaluation method using a supervised model, where the data labels come from the unsupervised process. This approach is demonstrated on two public datasets and on a real industrial dataset.

Keywords: Anomaly detection · Machine learning · Agnostic evaluation · Industrial applications.

1 Introduction

Nowadays, industrial systems are equipped with sets of sensors that perform different physical measures on the parts of the system. These measures are linked to the instant they are observed and form time series that can be used in various ways and give precious information about the behaviour of the system over time. An interesting use of these data is predictive maintenance that aims to anticipate breakdowns with the study of irregularities in data [1]. However, faulty sensors may generate irrelevant irregularities that disturb the analysis of data, leading to misinterpretations. For this reason, sensor faults must be detected so that the impact of faulty data on the predictive maintenance process is limited.

We chose to train anomaly detection models on selected training datasets. Yet, available datasets are usually not labelled, which means that we do not know if an observation is normal or generated by a faulty sensor. On top of that, for the same company, there are different systems with different operating contexts. Thus, we wish to build a solution that can be generalized to most industrial systems.

* This project is supported by ANITI through the French "Investing for the Future – PIA3" program under the Grant agreement noANR-19-PI3A-0004.

The contribution of this paper is twofold. We propose: 1) an unsupervised method, named SuMeRI (Successive Methods Run Iteratively), that performs anomaly detection on unlabelled datasets. It is based on applying successively different anomaly detection methods that detect different kinds of anomalies. Each method is applied iteratively on data where anomalies are removed at each iteration; 2) an agnostic evaluation method based on consistency checking, named CC-Eval (Consistency Checking Evaluation), that measures the similarity of the results of the unsupervised learned model with those of a supervised model learned with the training dataset labelled according to the unsupervised model (cf. Figure 1).

The paper is organized as follows. Section 2 provides a state of the art of the field of anomaly detection and positions our contribution. Section 3 presents the two methods, SuMeRI and CC-Eval. In Section 4, some results obtained with public datasets and real case study datasets are provided to validate our approach. To conclude, Section 5 includes a discussion of the results and directions for future works.

2 Related work on Anomaly Detection

Various communities have been interested in anomaly detection, starting with statisticians in the end of the 19^{th} century with the works of Edgeworth [2], followed by control scientists interested in fault detection and isolation, i.e. the FDI community [3], and later by artificial intelligence scientists that proposed paradigms for diagnosis reasoning, i.e. the DX community [4]. Many of these latter works are based on the existence of a model built from physical knowledge [5][6][7]. However, the complexity of today's systems which makes it difficult to build models and the growing interest in data-based methods have led to the development of many machine learning methods. This paper is interested in these methods.

A commonly accepted definition of anomaly, or *outlier*, is the one from Hawkins that defines an anomaly as "an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism" [8]. A survey published by Chandola et al. in 2009 [9] acts as a reference in the domain and provides all the required elements to approach the field. This survey introduces three kinds of anomalies:

- *point anomalies* defined as pointwise observations that appear abnormal compared to the rest of the dataset,
- *contextual anomalies* defined as pointwise observations that appear abnormal with respect to the context in which they occur,
- *collective anomalies* defined as collections of observations that are solely abnormal if studied together.

While point anomalies are the most frequent and the most trivial to detect, collective anomalies are nonetheless common in time series and this makes their detection important in industrial context with sensor data.

The survey by Chandola [9] also divides the methods according to the availability of data labels during the training phase. Learning methods using entirely labelled datasets, for which labels indicate for each instance if it is normal or abnormal, are called *supervised* and usually refer to the classification problem. But the cases where the knowledge about the labels is available are quite uncommon, and this is why semi-supervised methods are preferred. In this later case, models are learned from training datasets in which only one specific class is present, usually normal data. The methods of the last type are known as unsupervised and they do not use any knowledge a priori. Because of this, unsupervised techniques are more generalizable than supervised or semi-supervised. However, unsupervised methods often make the assumption that abnormal data are very few in the training set and that it is possible to apply a sufficiently robust semi-supervised method without being disturbed by outliers.

There are two types of outputs for anomaly detection methods: a decision or a score. The decision is mostly used in a supervised context and only gives the predicted class of each instance. The score is defined as the degree of abnormality (or, in some cases, normality) of an instance, in a range that is defined by the method itself. Then, one usually defines a threshold on this score to make a decision.

As described in the introduction, the detection of faulty sensors needs to be done in an unsupervised context. Studies about unsupervised anomaly detection are common because of the wide applicability of such methods, especially those based on the famous nearest neighbor method [10][11][12]. Because of the large amount of methods that claim to be the best in response to specific needs and the lack of labelled data to evaluate the learned models, numerous comparative studies have been done to help selecting the one that fits the best a defined context [13][14][15].

However, authors of these studies hardly give the datasets that are used to get their results nor they indicate how they parameterize the algorithms that they compare. Even if they provide such information, their analyzes focus solely on a specific experimental context or dataset, and the results can not be generalized to other contexts, as proved by the differences in the results of the different works. Some comparative works try to be more generalizable using a wide variety of datasets and evaluation metrics [16], but it is still difficult to know if their results can be applied in a general context, in particular for the detection of sensors faults. Eventually, it is not possible to compare the proposed algorithms on our real case study datasets because the metrics computed to perform the comparison require the availability of validation data.

For this reason, we developed SuMeRI to apply different methods successively and CC-Eval to give a first sight on an evaluation of the model in a fully unsupervised way, bypassing the lack of validation data. Applying different methods offers the opportunity to detect different kinds of anomalies (point, contextual and collective), the only requirement to use a method in SuMeRI being the availability of a score output. Interestingly, SuMeRI can be used in an univariate or a multivariate setup, which is important to distinguish an anomaly caused by a faulty sensor from an anomaly affecting the system components. To make it

more robust to anomalies in the training set, SuMeRI runs each method iteratively, so the training set is cleaned from the most outlying instances at each iteration until a fixed stopping point. It returns an unsupervised model able to compute an anomaly score on data and decide which instances are abnormal, this decision being based on a learned threshold on the score. To evaluate the model returned by SuMeRI, we propose CC-Eval that uses a supervised method on the dataset labelled by the unsupervised SuMeRI model. CC-Eval measures the consistency of the SuMeRI model by comparing its results to those of the supervised model learned on the training dataset labelled according to the SuMeRI model. The rational under this idea is that the greater the consistency, the more likely it is that SuMeRI's results follow a well-founded rule. The whole process is represented in Figure 1.

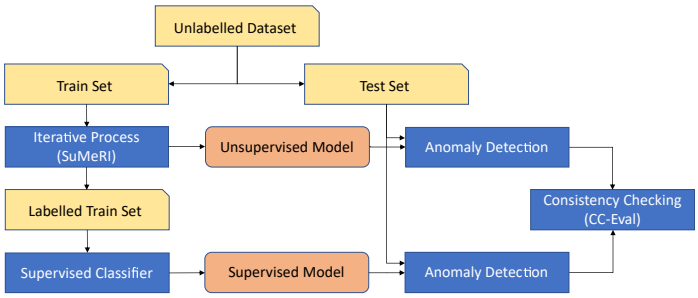


Fig. 1: Architecture of the proposed solution.

3 Unsupervised anomaly detection and agnostic evaluation

3.1 Successive Methods Run Iteratively (SuMeRI) approach

Methods for anomaly detection that do not require labelled data usually need to estimate the anomaly rate in the training set. Yet, this information is unknown in most cases and fixing a too high or low rate leads to a high false positive or high false negative rate respectively. On the other hand, anomalies may be of different types, which may require different anomaly detection methods. That is why we developed the SuMeRI approach that applies specialized anomaly detection methods in successive phases and runs each method iteratively on data that are increasingly free of anomalies. The complete SuMeRI algorithm is given by Algorithm 1.

Iterative process Each iteration detects the current *most outlying instances* and removes them from the training set so that the next iteration trains a new model from a "cleaner" data set. This requires an anomaly score to decide which samples are the most outlying ones and a stopping condition to stop the iterations so that over fitting is avoided. The SuMeRI approach can hence use any anomaly detection method that meets the two following conditions:

– it computes an anomaly score s_i for each data instance i and discriminates anomalous instances from normal ones, hence defining a threshold l_1,
– it allows one to define a measure to evaluate the quality of the model over the iterations.

The "most outlying instances" are identified by a score above the threshold l_2 defined by:

$$l_2 = \underset{s_i \in S_{out}}{\text{argmax}}(\min_{s_j \neq s_i} (|s_i - s_j|)) \quad (1)$$

Let us define S_{in} (S_{out}) as the set of scores associated to normal (outlier) instances and assume that scores are scaled in $[-1, 1]$ at the end of each iteration.

The heuristic used to stop the iterations at each phase is based on the score distribution. Removing the most outlying instances at each iteration shifts the threshold l_1 towards greater score values. It also results in increasing the kurtosis, i.e. the *tailedness*, of the normal score distribution and decreasing the kurtosis of the outlier score distribution. The desired model is the one where the outliers detected by the model and the inliers have similar kurtosis. In practice, we compare the maximal distance between nearest neighbors score values for outliers and for inliers and stop when the condition of Equation (2) is satisfied:

$$d_{max}(S_{out}) = \max_{s_i \in S_{out}} (\min_{s_j \neq s_i} (|s_i - s_j|)) \leq d_{max}(S_{in}) = \max_{s_i \in S_{in}} (\min_{s_j \neq s_i} (|s_i - s_j|)) \quad (2)$$

Algorithm 1 SuMeRI algorithm

Input: $data, methods$ ▷ *methods* contains One-Class SVM and Linear Regression

$data \leftarrow scale(data)$
$l \leftarrow length(methods)$
$models \leftarrow list[length \leftarrow l]$ ▷ One model learned per method
for $i \leftarrow 1, l$ **do**
  $method \leftarrow methods[i]$
  $stopping\_condition \leftarrow False$
  while $stopping\_condition \neq True$ **do** ▷ Each loop on a method is a phase
    $models[i] \leftarrow learn\_model(method, data)$
    $score \leftarrow compute\_score(model, data)$
    $stopping\_condition \leftarrow stopping\_decision(score)$
    $data \leftarrow normal\_data(data, score)$
  end while
end for
return *models*

Successive phases The successive phases in SuMeRI apply different methods to detect different anomaly types; point, contextual, and collective. As mentioned in Section 2, many unsupervised methods make the assumption that there is a non-significant proportion of outliers in the dataset. These methods are thus close to semi-supervised methods, which assume that the training set is representative of normality, thanks to robustness to possible outliers. Because each phase of SuMeRI removes the detected outliers from the training set, SuMeRI allows the

use of these methods for the last phases. Nevertheless, unsupervised methods able to learn without assumption on the anomaly rate in the training dataset should be used in the first phases.

3.2 Agnostic evaluation based on consistency checking (CC-Eval)

To evaluate the performances of anomaly detection models, one commonly uses metrics such as the receiver operating characteristic (ROC) and the precision-recall (PR) curves, using their area under the curve (AUC) and average precision (AP) to be compared between different methods [13]. However, these metrics require labelled data to be used which are not available in our case. An unsupervised solution has been suggested to evaluate methods without the use of labelled data [17], but the consistency of its results in comparison with the ROC and PR curves seems to depend on the datasets.

Thus, we propose a solution to overcome this issue and introduce CC-Eval. The purpose of CC-Eval is twofold: 1) check if there is a learnable logic behind the obtained results, and 2) check if the iterative learning has not gone too far in the search for anomalies, leading to overfitting and high false positive rates.

To achieve this, the datasets are divided into a training set and a testing set (cf. Figure 1), representing 90% and 10% of the whole dataset respectively. To avoid temporal discontinuity in the data that could have an impact on learned models, the splitting is done by blocks from the tails of the dataset. SuMeRI is then applied on the learning set and the whole dataset is labelled by applying the learned model, considering three classes: normal instances, point anomalies and contextual anomalies. Using the learned labels as the ground truth, a supervised classifier is used to learn the classification of data taking for characteristics the real value and contextual values defined earlier. The learned classifier is then used to predict the classes for the test set.

Here, the consistency is defined as the logic of the discrimination learned between inliers and outliers. The assumption behind this method is that a supervised classifier is efficient if there is consistency in the classes to learn. Thus, the consistency is evaluated with the area under the ROC curve (ROC-AUC) which quantifies how well the model is able to distinguish the different classes. Because there are more than two classes, it is required to compute an aggregation on the different ROC curves. We propose to compute the ROC-AUC as the mean of four different approaches:

- macro average with one-vs-rest approach,
- weighted average with one-vs-rest approach,
- macro average with one-vs-one approach,
- weighted average with one-vs-one approach.

The macro average approaches do not take into account the imbalance in classes while the weighted average approaches do. The one-vs-rest approaches compute the mean of the ROC-AUC for each class against all the others (A vs B&C, B vs A&C, C vs A&B). The one-vs-one approaches compute the mean of the

ROC-AUC for each binary combination of classes, excluding the data in the other classes (A vs B, A vs C, B vs C).

In this context, the main issue with the classification problem is that the classes are imbalanced. Usually, there are far more instances in the normal class than instances in the outlier classes, and the most populated class for outliers varies depending on the dataset. Nevertheless, studies have been made to resolve this issue, using for the most of them sampling strategies to reduce the gap between the size of the classes or cost-sensitive learning to reduce the effects of these gaps [18]. Four different classifying methods have been considered which use ensemble of classifiers trained on different samples generated by random under-sampling: 1) EasyEnsemble [19], 2) Bagging predictors [20], 3) Random Forest [21] and 4) RUSBoost [22].

To decide which method to use, we generate ten different models by applying the iterative solution with different parameters on a multivariate labelled public dataset. Then, we compute the ROC-AUC using the known labels on these ten models and establish a ground truth ranking among them based on the results. The best classifier is defined as the one that gives the closest ranking to the ground truth. This experience showed that the EasyEnsemble classifier with an under-sampling on the majority class gives the best results, closely followed by the RandomForest classifier with the same sampling strategy. Both of these methods will be used in the following section.

4 Experimental results

For our experiments, we used the implementation of One-Class SVM and Linear Regression from scikit-learn, a Python library for Machine Learning, in SuMeRI. The ROC-AUC approaches are also implemented in this library. The classifiers used in CC-Eval are from the imbalanced-learn Python library with the names *EasyEnsembleClassifier* and *BalancedRandomForestClassifier*.

With the chosen configuration, SuMeRI only requires three parameters. The first one is the proportion of anomalies to compute the models, it should be low enough to drop only few anomalies at each step. It is used as the nu parameter for the scikit-learn implementation of One-Class SVM and also to fix the separation between anomalies and normal instances during the second phase. The second one is the kernel function used for the kernel trick in One-Class SVM. The last one is the window size on which are computed the contextual metrics.

4.1 SuMeRI setting

In our setting, we want to detect point anomalies and contextual anomalies. Hence our SuMeRI setting includes two phases:

– The first phase uses the One-Class SVM method. Leveraging the kernel trick, the One-Class SVM method learns a non linear separation between normal and anomalous classes. Then, the anomaly score is computed as the signed distance of each instance from the separation.

– The second phase is a prediction method. For each instance, we use its context as explanatory variables to predict its value. We define the context of an instance temporally with statistical metrics on the previous and following instances in a defined window. These metrics are the minimum, the first quartile, the median, the third quartile, the maximum, the mean, the standard deviation, and the mean squared error. The score is then computed based on the prediction error, as the Euclidean distance between the predicted value and the real value, with a standardization to be comparable with the score of the first phase. Two methods have been considered for this prediction: Linear Regression and Neural Networks. The later gives slightly better results but at a high computational cost, that is why we have opted for the former.

4.2 Public Datasets

SuMeRI and CC-Eval, presented in Section 3, have been tested on two public datasets. The first one is from the Numenta Anomaly Benchmark [23] and it measures the temperature on a machine with known system failures in the dataset. The second contains the Http requests in the KDD Cup 1999 dataset, used for The Third International Knowledge Discovery and Data Mining Tools Competition, as it is presented in the ODDS library [24].

The machine temperature dataset is close to our industrial application domain. It is univariate and contains 22695 instances with three known anomalies: two being linked to a breakdown and a third one that appears as a hint for the occurrence of the second breakdown. The SuMeRI model is learned with the linear kernel, an anomaly rate fixed at 0.001 and a window of size 20, which means that we consider ten periods before and after each instance to compute the contextual metrics. The SuMeRI results are shown in Figure 2 where the three green areas represent the actual anomaly positions.

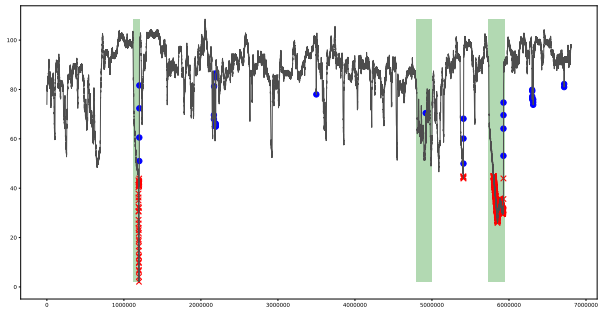


Fig. 2: Results of SuMeRI for the machine temperature dataset. Green areas represent the true anomalies, the left one is the first known anomaly, the middle one is the third and the right one is the second. Outliers detected by the first phase are represented with red crosses and those detected by the second phase with blue circles.

The first iterative phase with One-Class SVM converges after three iterations and is able to detect the outliers in the left and right green areas that correspond to the two big failures. There is still one false positive due to the fact that the iterations stopped one step too late. The second phase converges after six iterations. Because of the use of statistical metrics in a window around each point for the prediction, anomalies are found where the signal has rapid variations. This phase is able to detect anomalies in the three green areas. Let us notice that only the HTM method makes the same detection [23], but it rises more false positives than true positives. The results of CC-Eval are given in Table 1.

Table 1: CC-Eval results for the model learned on the machine temperature dataset. The ROC-AUC are computed with different multi-class and average strategies for the chosen classification methods. The last column is the mean of the four previous ones.

	one-vs-one		one-vs-rest		mean
	macro	weighted	macro	weighted	
EasyEnsemble	0.9915	0.9879	0.9881	0.9794	0.9867
BalancedRandomForest	0.9800	0.9733	0.9678	0.9865	0.9769

The Http dataset contains three variables and is entirely labelled. Among its 567498 instances, there are 2211 anomalies (0.4%) that have to be detected. However, the information on the cause of the anomalies is unknown and this example does not match our application domain, but proves the applicability of the presented solution for multivariate datasets. SuMeRI is applied with the linear kernel, a window of size 20 and an anomaly rate fixed at 0.0001. The results of both phases in comparison with the known labels are shown in Table 2. The first phase converges since the first iteration. The iterative process is not of great use in this case, but the stopping condition occurs at the right iteration according to the results. The second phase converges after three iterations and only detects 11 outliers in the remaining dataset. Its performances are poor in comparison with those from the first iteration with only one true positive. Because we know the labels for this dataset, it is possible to compute the ROC AUC of the model which is equal to 0.9989. We can also compute this score for each phase. The ROC AUC for the model learned during the first phase is equal to 0.9970 for the whole dataset while it is equal to 0.9959 for the model learned during the second phase on the data that are negatively labelled by the first phase (0.4354 when applied on the whole dataset, because the instances labelled positively by the first phase are not detected as outliers, which means the true positive rate is very low). The results of CC-Eval are given in Table 3 and are a bit lower than the true ones.

For these two public datasets, the results of CC-Eval are quite good and seem to be consistent with the model performances.

4.3 Real Case Study Datasets

SuMeRI and CC-Eval also have been applied on a real case study dataset where labels were not available but where some anomalies were known, as in the ma-

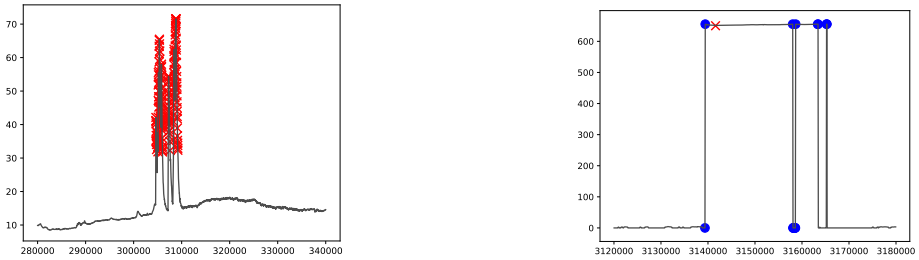
Table 2: Results of SuMeRI for the machine temperature dataset.

	Phase 1	Phase 2	Total
True Positives *(rate)*	2202 *(99.593%)*	1 *(11.111%)*	2203 *(99.638%)*
True Negatives *(rate)*	565241 *(99.992%)*	565231 *(99.998%)*	565231 *(99.990%)*
False Positives *(rate)*	46 *(0.008%)*	10 *(0.002%)*	56 *(0.010%)*
False Negatives *(rate)*	9 *(0.407%)*	8 *(88.889%)*	8 *(0.362%)*

Table 3: CC-Eval results for the model learned on the http dataset, computed as for Table 1.

	one-vs-one		one-vs-rest		mean
	macro	weighted	macro	weighted	
EasyEnsemble	0.9331	0.8999	0.9611	0.9944	0.9471
BalancedRandomForest	0.9836	0.9754	0.9777	0.9986	0.9838

chine temperature dataset used in the previous subsection. This dataset is also univariate and constituted of temperature measurements in a building that is equipped with an intelligent auto-regulating system. This dataset is one of the kind that is targeted by this solution, as a timeserie with changing in the operating mode which makes it difficult to model.



(a) First anomaly to detect

(b) Second anomaly to detect

Fig. 3: Results of SuMeRI for the building temperature dataset. The X-axis represents a number of seconds since the first observation. The first anomaly is well detected by the first phase (red crosses). The second anomaly is not detected by the One-Class SVM phase which consider these values as a normal cluster, but the offset is detected by the second phase (blue circles).

The dataset contains 446581 instances sampled every 15 seconds and there are two areas where we want to detect anomalies. The first one is a provoked growth in temperature with the heat of the sensor, the second one is a treatment issue on the signal with an added offset. The rbf kernel for the SVM method has been used in SuMeRI with an anomaly rate of 0.0001 and a window of size 20. The results are shown in Figure 3. The offset is not detected by the first phase because it is considered as a second normal cluster which causes the score, computed based on the distance to the learned separation, to be the same as the more normal values for the real normal cluster. However, the instances where the changes happened are detected as anomalous by the second

phase. The Table 4 displays the results of CC-Eval, which are quiet weak for the *EasyEnsembleClassifier* in comparison with the results for the public datasets. This is consistent with the fact that the learned model does not perform as well.

Table 4: CC-Eval results for the model learned on the building temperature dataset, computed as for Table 1 and Table 3.

	one-vs-one		one-vs-rest		mean
	macro	weighted	macro	weighted	
EasyEnsemble	0.8333	0.7500	0.6686	0.4927	0.6817
BalancedRandomForest	0.9997	0.9996	0.9998	0.9999	0.9998

5 Conclusions and Future Works

In this paper, an approach for anomaly detection in unlabelled industrial datasets has been proposed. The approach, called SuMeRI, is based on applying successive specialized anomaly detection methods in an iterative way. The methods admissible in SuMeRI should meet two requirements: the availability of an anomaly score and the ability to measure the quality of the model. The results presented in this paper are obtained with the application of two methods, one specialized in detecting point anomalies and the other in contextual anomalies. Because it is difficult to evaluate the results for unlabelled datasets, an agnostic consistency checking method named CC-Eval has been proposed.

Future works will focus on the improvement of SuMeRI in the following directions:

- Instantiate SuMeRI with more than two successive methods while still limiting the number of hyper-parameters to be tuned. This could be done with the use of ensemble methods.
- Include a specialized method for collective anomalies.
- Make explicit and explain automatically the rules underlying the consistency found by CC-Eval.
- Give a better evaluation of the whole solution and compare it to state-of-the-art methods.

References

1. Garcia, M.C., Sanz-Bobi, M.A., del Pico, J.: SIMAP: Intelligent System for Predictive Maintenance: Application to the health condition monitoring of a windturbine gearbox. E-Maintenance Special Issue, Comput. Ind. **57**(6), 552–568 (2006)
2. Edgeworth, F.Y.: On discordant observations. Phylosophical Magazine **23**(5), 364–375 (1887)
3. Gao, Z., Cecati, C., Ding, S.X.: A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. IEEE Transactions on Industrial Electronics **62**(6), 3757–3767 (2015)
4. Peischl, B., Wotawa, F.: Model-based diagnosis or reasoning from first principles. IEEE Intelligent Systems **18**(3), 32–37 (2003)

5. Trave-Massuyes, L., Bridging Control and Articial Intelligence Theories for Diagnosis: A Survey. Engineering Applications of Artificial Intelligence, **27** 1–16 (2014)
6. Cordier, M.O., Dague, P., Pencole, Y., Trave-Massuyes, L.: Diagnosis and supervision: model based approaches. In: A Guided Tour of Artificial Intelligence Research, vol.1, Springer (2019)
7. Biswas, G., Cordier, M.O., Lunze, J., Staroswiecki, M., Trave-Massuyes, L. (Eds): IEEE SMC Transactions - Part B, Special Issue "Diagnosis of Complex Systems: Bridging the methodologies of the FDI and DX Communities", **34**(5) (2004)
8. Hawkins, D.M.: Identification of outliers. Springer, Dordrecht (1980)
9. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection: A Survey. ACM Computing Surveys **41**(3), Article 15 (2009)
10. Breuning, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: Identifying Density-Based Local Outliers. In: Proc. ACM SIGMOD 2000 Int. Conf. On Management of Data, pp. 93–104, ACM, Dallas (2000)
11. Kriegel, H.-P., Kröger, P., Schubert, E., Zimek, A.: LoOP: Local Outlier Probabilities. In: Proceedings of the 18th ACM conference on Information and knowledge management (CIKM'09), pp. 1649–1652, ACM, Hong Kong (2009)
12. Padamitriou, S., Kitagawa, H., Gibbons, P.B., Faloustos, C.: LOCI: Fast Outlier Detection Using the Local Correlation Integral. In: Proceedings of the 19th International Conference on Data Engineering, pp. 315–326, IEEE, Bangalore (2003)
13. Domingues, R., Filippone, M., Michiardi, P., Zouaoui, J.: A comparative evaluation of outlier detection algorithms: Experiments and analyses. Pattern Recognition **74**, 406–421 (2018)
14. Martin, R.A., Schwabacher, M., Oza, N.C., Srivastava, A.N.: Comparison of Unsupervised Anomaly Detection Methods for Systems Health Management Using Space Shuttle. In: Proceedings of the Joint Army Navy NASA Air Force Conference on Propulsion (2007)
15. Auslander, B., Gupta, K.M., Aha, D.W.: Comparative evaluation of anomaly detection algorithms for local maritime video surveillance. In: Proceedings of SPIE - The International Society for Optical Engineering, SPIE (2011)
16. Goldstein, M., Uchida, S.: A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. PLoS ONE **11**(4) (2016)
17. Goix, N.: How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms?. hal-01341809 (2016)
18. Ganganwar, V.: An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering **2**(4), 42–47 (2012)
19. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **39**(2), 539–550 (2008)
20. Breiman, L.: Bagging predictors. Machine learning, **24**(2), 123–140 (1996)
21. Chen, C., Liaw, A., Breiman, L.: Using Random Forest to Learn Imbalanced Data. University of California, Berkeley **110**, 1–12 (2004)
22. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: RUSBoost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics, Part A (Systems and Humans) **40**(1), 185–197 (2009)
23. Lavin A., Ahmad S.: Evaluating Real-Time Anomaly Detection Algorithms–The Numenta Anomaly Benchmark. In : 14th International Conference on Machine Learning and Applications (ICMLA) pp.38–44, IEEE, Miami (2015)
24. Rayana, S.: ODDS Library [http://odds.cs.stonybrook.edu]. Stony Brook University, Department of Computer Science (2016)