SGraphZoe: Explainable self-supervised framework for signal-based anomaly detection

Mikhail Kamalov
MyDataModels
Sophia Antipolis, France
mikhail.kamalov@mydatamodels.com

Luca Uggeri
MyDataModels
Sophia Antipolis, France
luca.uggeri@mydatamodels.com

Ingrid Grenet
MyDataModels
Sophia Antipolis, France
ingrid.grenet@mydatamodels.com

Jonathan Daeden
MyDataModels
Sophia Antipolis, France
jonathan.daeden@mydatamodels.com

***Abstract*—Signal-based anomaly detection is a recurring problem that has drawn the attention of many research projects and resulted in the development of multiple solutions. One of the main obstacles to anomaly detection is the rarity of the occurrences of interest. Extremely small amount of labelled data is troublesome from the training perspective since it has a detrimental influence on the accuracy of predictions. The second challenge is providing a clear and understandable model. Answering this second issue is particularly important for a variety of industries since it is beneficial to understand what causes outliers in order to avoid them in the future. To address the aforementioned concerns, we propose a novel self-supervised framework named SGraphZoe which outperforms linear semi-supervised state-of-the-art outlier detection algorithms while maintaining transparency throughout training and prediction steps. This framework is built on a Self-supervised strategy and combines a semi-supervised (Graph Diffusion & PCA) and a supervised (Zoetrope Genetic Programming) algorithms.**

I. Introduction

The detection of anomalies in signals represented by univariate time series data can be challenging. The analysis of the recording of a sensor in a factory perfectly illustrates this type of problem. Indeed, several types of anomalies [1] can occur:

- Process anomalies are unexpected events and failures impacting a process in the plant;
- Anomalies of change of pace are different operations of a machine from one day to the next one;
- Anomalies of data are issues in the data collection and storage system that generate erroneous data and time series.

We must emphasize that the problems with anomaly detection stem not only from the fact that different types of anomalies have different detection patterns but also from the challenges in the rarity of their occurrences (i.e., extremely small amount of labelled data) which is troublesome from the supervised training. Nowadays, it already exists linear state-of-the-art (SOTA) algorithms for the unsupervised outlier, and novelty detection, such as Isolation Forest (IForest) [2] and Local Outlier Factor (LOF) [3] respectively. Even more, there exist semi-supervised[1] linear algorithms such as Label Propagation (LP) [4] which support self-supervised training strategy [5]. However, all of them do not show transparent and explainable training and prediction steps which is an important demand from industrial companies. Indeed, it is beneficial to understand what causes outliers in order to be able to avoid them in the future. Therefore, in this work we concentrated on the development of a framework which allows us to detect any anomalies and explain their likely causes (e.g. server or switch malfunction, Distributed Control System (DCS) issues, sensor failure) and possible sources (e.g. problems derived from physical limits, problems in the data storage system) of their appearance in signals despite of the missing labelled data. We thus developed a novel framework called **SGraphZoe** which is based on a **S**elf-supervised strategy and the combination of two algorithms: **Graph** Diffusion & PCA (GDPCA) [6] and **Zoe**trope Genetic Programming (ZGP) [7].

II. Notations and Framework definition

A. Notations

Let $X = [X_i]_{i=1}^n$, where $X_i = (X_{i,j})_{j=1}^d$, be the matrix of observed time series, with length of time series d and total number of observations n. Then let $\{\mathcal{C}_1, \mathcal{C}_2\}$ be the set of two classes. In our case, $\mathcal{C}_1$ is non-anomalys and $\mathcal{C}_2$ is anomaly. Also, let $Y = [Y_i]_{i=1}^n$ be a label matrix where $Y_i = (Y_{i,j})_{j=1}^2$, such that $Y_{i,1} = 1$ if $X_i \in \mathcal{C}_1$ and $Y_{i,1} = 0$ otherwise. Note that the original Y is contained a labelled observations of size n_l, and an unlabelled one of size n_u, for semi-supervised learning $n_l \ll n_u$ and Y_i being the zeros vector in the case of all unlabeled data. Also, we define the graph-based setup which will be useful further for understanding GDPCA algorithm: $A = [A_{i,j}]_{i,j=1}^{n,n}$ is an adjacency matrix which could be replaced by a similarity matrix $W = [h(X_i, X_j)]_{i,j=1}^n \in R^{n \times n}$ where $h(\cdot,\cdot)$ is a positive definite kernel, $D = \text{diag}(D_{i,i})$ is a diagonal matrix with $D_{i,i} = \sum_{j=1}^n A_{i,j}$.

Lastly, we should mention that our framework was inspired by semi-supervised PaZoe [8] framework, which performed well for general time series classification problems.

[1] Extremely low amount of labels.

979-8-3503-3959-8/23/$31.00 © 2023 European Union

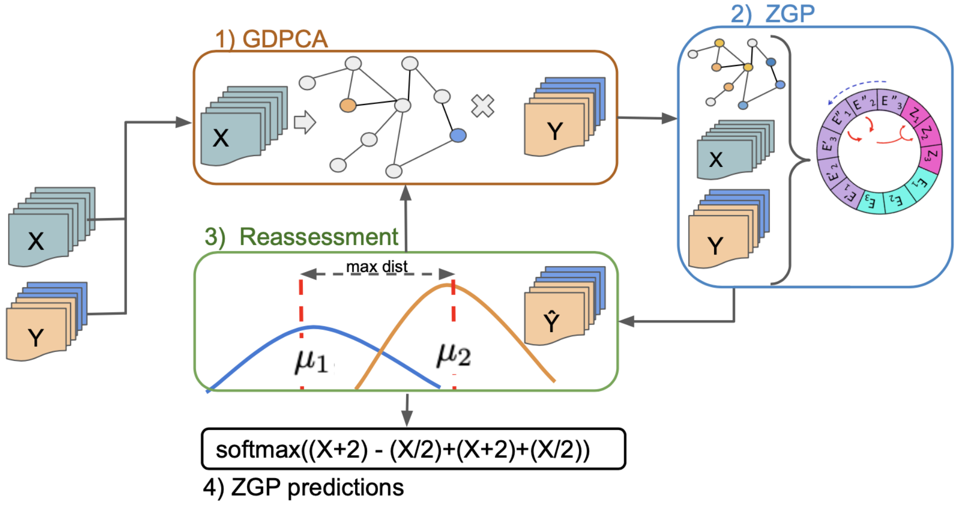

Fig. 1. SGraphZoe sequence: 1) Semi-supervised training by GDPCA; 2) Supervised training by ZGP: Augmentation X with GDPCA inferences; ZGP training; 3) Reassessment by computing distance $Cosine(\mu_1, \mu_2)$; 4) Final interpretable classification formula from ZGP.

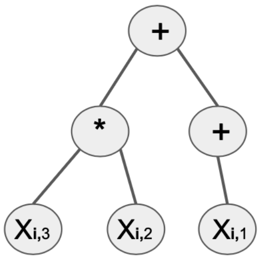

Fig. 2. Example of zoetrope which could be generated from the input features $(X_{i,j})_{j=1}^{d}$.

```
INPUT: $X$, $Y$, $\tau$;
INITIALIZE: $\gamma = \infty$, $Y$
for $t = 0$ to $\tau$ do
    $\hat{Y} = GDPCA(X, Y)$ #Self-labelling;
    $Y = ZGP(X, \hat{Y})$ #Supervised training;
    $\mathcal{C}_1 = \{i | Y_{i,1} > Y_{i,2}\}, \mathcal{C}_2 = \{i | Y_{i,2} > Y_{i,1}\}$
    $\mu_1 \approx \frac{\sum_{i \in \mathcal{C}_1} X_i}{|\mathcal{C}_1|}; \mu_2 \approx \frac{\sum_{i \in \mathcal{C}_2} X_i}{|\mathcal{C}_2|}$;
    $\widetilde{\gamma} = Cos(\mu_1, \mu_2)$ #Label reassessment;
    if $\widetilde{\gamma} < \gamma$ then
        $\gamma = \widetilde{\gamma}, \widetilde{Y} = Y$
    end if
end for
```

Algorithm 1: SGraphZoe

B. SGraphZoe framework

This study is built on the central concept that all data can be represented by a graph, which can then be utilized to improve default features as well as for transparent label diffusion. Because of the aforementioned concepts, SGraphZoe employs the most recent best linear graph/non-graph-based algorithm as GDPCA for the semi-supervised step, which explicitly makes a label spreading through the generated graph for labelling of unlabeled data for further training of the ZGP algorithm in the supervised regime. The detailed process of SGraphZoe is illustrated in Figure 1 and in Algorithm 1 wher e τ is the total numer of iterations and γ is the distance between approximated expectations defined further in Assumption 1. It consists in the following steps:

1) *Semi-supervised learning using GDPCA*: GDPCA is the latest SOTA linear graph-based semi-supervised learning framework [6]. Using GDPCA is interesting since it is suitable for both graph-based and non-graph-based data and is able to train on an extremely small amount of labelled data. This semi-supervised training regime of GDPCA, in particular, enables the labelling of unlabelled data for future training under a supervised regime. In SGraphZoe we utilize GDPCA to enrich the original label matrix Y for further training. The explicit classification step of GDPCA is:
$\hat{Y} = \alpha \left(D^{\sigma-1} A D^{-\sigma} + \delta S D^{-2\sigma+1} \right) + (1 - \alpha) Y$, where $\delta \in (0, 1)$ sets the influence of estimated covariance between signals $S \in \mathcal{R}^{n \times n}$ on A and $\alpha \in (0, 1)$ is the jump parameter of PageRank [9]. Thus, GDPCA allows visualizing labels spreading through the graph;
2) *Supervised learning using ZGP*: ZGP is a genetic programming approach for symbolic regression (GPSR) in which mathematical equations (which can be reprenseted by trees such as in Figure 2) are evolved through several iterations using genetic operators (i.e. crossover and mutation). The final model corresponds to the tree that best fits the training data, according to a fitness function. ZGP is unique in the way the mathematical formulas are built, allowing an efficient calculation and preventing models from overgrowing, a major problem in GPSR [10]. More precisely, these equations are built from a number m_e of randomly chosen features and constants $(E_1, \ldots, E_{m_e})$ on which several ”fusions” are applied through ”maturation steps”. The ”mature” equations finally obtained, called “zoetropes” $(Z_1, \ldots, Z_{m_e})$ (see Figure 2), are linearly combined by multinomial logistic regression penalized by Elastic net [11]. Thanks to this zoetrope

TABLE I. PERFORMANCE COMPARISON ($Macro - F1$)

Dataset	SGraphZoe (LR20%)	SGraphZoe (LR10%)	IForest	LOF	LP (LR20%)	LP-self (LR20%)	PaZoe(LR20%)
Wafer	0.878	0.709	0.749	0.542	**0.949**	**0.949**	0.630
ECGFiveDays	**0.657**	0.530	0.460	0.338	0.607	0.608	0.646
ToeSegmentation2	**0.548**	0.516	0.155	0.155	0.449	0.449	0.524
TwoLedECG	**0.564**	0.528	0.497	0.352	0.559	0.562	0.545

TABLE II. DATASET STATISTICS

Dataset	Type	AR	d	n	LR20%	LR10%
Wafer	Sensor	0.11	152	7164	19	9
ECGFiveDays	Sensor	0.20	136	8842	2	1
ToeSegmentation2	Motion	0.25	343	166	3	1
TwoLedECG	Sensor	0.49	82	1162	2	1

mechanism, ZGP is able to provide interpretable classification formulas for each class (an example of a formula is presented in Equation (1)). Here ZGP is trained on X and Y previously augmented by inferences from GDPCA;

$$Y_{i,2} = 0.5 * X_{i,1} - 0.1 * X_{i,15} + 0.2 * X_{i,2} \forall X_i \in X \quad (1)$$

3) *Label Reassessment*: finally, GDPCA is retrained by using the predictions made by ZGP in the previous step such that the entire framework uses a self-supervised training loop. To understand how to extract the best predictions from this self-supervised training loop, the following assumption is made:
Assumption 1: Let assume that the time series matrix of observations X is sampled from the Gaussian distribution:

$$X_1, \dots, X_{\frac{n}{2}} \sim \mathcal{N}(\mu_1, C)$$
$$and\ X_{\frac{n}{2}+1}, \dots, X_n \sim \mathcal{N}(\mu_2, C),$$

where μ_1, μ_2 are the expectations for the two classes with $Cos(\mu_1, \mu_2) > 0$ where $Cos(\cdot, \cdot)$ is the cosine distance and C is the covariance matrix of classes. Assumption 1 shows that the time series are generated from Gaussian distribution with different expectations (expectation of outlier and non-outlier). Then, based on the Assumption 1, an unsupervised reassessment of predictions is performed within the self-supervised framework by computing the cosine distance between approximated μ_1 and μ_2. If this distance is lower than the current γ (initialized to infinity at the begining of the algorithm), then it becomes the new optimal γ. In particular, this reassessment ensures returning the best optimal prediction over all iterations (τ), which corresponds to the maximum distance between approximated μ_1 and μ_2 ever encountered through the Algorithm 1.

III. EXPERIMENTS

A. Datasets

In the experimental part of this work, we considered three sensor datasets and one motion dataset, which are publicly available[2]:

- *Wafer* [12] is a collection of inline process control measures acquired from various sensors during the processing of silicon wafers for semiconductor manufacture;
- *TwoLeadECG* is an ECG dataset from MIT-BIH Long-Term ECG Database (ltdb) Record ltdb/15814, begining at time 420, ending at 1019;
- *ECGFiveDays* is an ECG dataset: 12/11/1990 ECG date: 17/11/1990;
- *ToeSegmentation* is derived from the CMU Graphics Lab Motion Capture Database (CMU) where motions are classified by their motion descriptions into the normal walk and abnormal walk (e.g. hobble walk).

The statistics for the aforementioned datasets are presented in Table II where AR is the anomaly ratio and $LR20\% = |\mathcal{C}_2| * 0.2$, $LR10\% = |\mathcal{C}_2| * 0.1$ are the label ratio that fix the number of labelled nodes required for semi-supervised training from each class. In addition, for all of the aforementioned datasets, we used the following training/testing strategy: at first, we merged training and test observations available by default for aforementioned datasets into one dataset; we took $20\% or 10\%$ labelled observations for each class and trained on the merged dataset, with a final estimation of performance based solely on test observations. This strategy of combining the training and testing observations for the model training provides for a fair comparison of unsupervised, semi-supervised, and self-semi-supervised models, as all of them may use all available unlabeled data throughout the training process.

B. State-of-the-art (SOTA) algorithms

In order to compare the performance of SGraphZoe with SOTA algorithms, we took several types of linear algorithms: unsupervised outlier detection such as IForest [2] and LOF [3]; semi-supervised ones such as LP [4] and PaZoe [8] and a self-semi-supervised one named LP-self. Note that LP-self is a combination of SelfTrainingClassifier[3] [13] and LP [4]. For a fair comparison, we have trained the aforementioned algorithms with respect to their best hyperparameters defined in their respective works. In particular, for GDPCA and ZGP algorithms applied inside of SGraphZoe, we selected the default hyperparameters from [6] and [7], respectively. We defined the number of iterations $\tau = 10$. We utilize $Macro - F1$ (2) score for assessment since all of the datasets are imbalanced.

$$Macro - F1 = 2 * \frac{MacroPrecision * MacroRecall}{MacroPrecision + MacroRecall} \quad (2)$$

[2] https://www.timeseriesclassification.com/index.php

[3] https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.SelfTrainingClassifier.html

C. Results

The results of SGraphZoe on various datasets are presented in Table I. It shows that SGraphZoe outperforms the linear unsupervised, semi-supervised and self-semi-supervised SOTA algorithms in almost all of the datasets. Even more, we could note that with a reduction of LR (from 20% to 10%), SGraphZoe performs closely to an unsupervised regime and still provides competitive performance. Finally, we should note that SGraphZoe not only shows high performance but also provides transparent and explainable predictions through final classification formulas extracted from ZGP. Moreover, SGraphZoe benefits of a transparent training process since it uses the label diffusion through the graph mechanism from GDPCA in combination with ZGP zoetropes.

IV. Conclusion

In this paper, we demonstrated how, in SGraphZoe framework, the critical issue of a lack of labeled observations in the case of anomaly detection in signals, could be managed by labels diffusion through the generated graph (GDPCA) in conjunction with ZGP algorithm under self-supervised iterations. Moreover, SGraphZoe provides more interpretable classification formulas for each class (e.g. Equation (1)) compared to the other SOTA algorithms and shows high performance on various types of datasets (see Table I).

In future work, we want to assess SGraphZoe's performance on a broader range of datasets (e.g., pictures, networks) and adapt it to a fully unsupervised environment.

References

[1] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–33, 2021.

[2] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[4] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," *CMU Technical report*, 2002.

[5] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information systems*, vol. 42, no. 2, pp. 245–284, 2015.

[6] K. Avrachenkov, A. Boisbunon, and M. Kamalov, "Graph diffusion & pca framework for semi-supervised learning," in *The 15th Learning and Intelligent Optimization (LION)*, 2021.

[7] A. Boisbunon, C. Fanara, I. Grenet, J. Daeden, A. Vighi, and M. Schoenauer, "Zoetrope genetic programming for regression," in *Genetic and Evolutionary Computation Conference (GECCO)*, 2021.

[8] M. Kamalov, A. Boisbunon, C. Fanara, I. Grenet, and J. Daeden, "Pazoe: classifying time series with few labels," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1561–1565.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[10] J. Žegklitz and P. Pošík, "Symbolic regression algorithms with built-in linear regression," *arXiv preprint arXiv:1701.03641*, 2017.

[11] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[12] R. T. Olszewski, *Generalized feature extraction for structural pattern recognition in time-series data*. Carnegie Mellon University, 2001.

[13] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *33rd annual meeting of the association for computational linguistics*, 1995, pp. 189–196.