PaZoe: classifying time series with few labels

Mikhail Kamalov
INRIA
Sophia Antipolis, France
mikhail.kamalov@inria.fr

Aurélie Boisbunon
MyDataModels
Sophia Antipolis, France
abb@mydatamodels.com

Carlo Fanara
MyDataModels
Sophia Antipolis, France
cf@mydatamodels.com

Ingrid Grenet
MyDataModels
Sophia Antipolis, France
ig@mydatamodels.com

Jonathan Daeden
MyDataModels
Sophia Antipolis, France
jd@mydatamodels.com

***Abstract*—Semi-Supervised Learning (SSL) on graph-based datasets is a rapidly growing area of research, but its application to time series is difficult due to the time dimension. We propose a flexible SSL framework based on the stacking of PageRank, PCA and Zoetrope Genetic Programming algorithms into a novel framework: PaZoe. This self-labelling framework shows that graph-based and non-graph based algorithms jointly improve the quality of predictions and outperform each component taken alone. We also show that PaZoe outperforms state-of-the-art SSL algorithms on three time series datasets close to real world conditions. A first set was generated in house, taking data from industrial graded equipment in order to mimick DC motors during operation. Two other datasets, which include the recording of gestures, were taken from the public domain.**

***Index Terms*—temporal data, semi-supervised classification, PCA, PageRank, symbolic regression**

I. Introduction

The interest of classifying temporal data originates from many real-world problems, among which classification of failure types in industrial equipment, gesture recognition or even brainwave recognition in EEG (electro-encephalogram) data [1], [2], [3], [4]. The difficulty of time series classification arises from the fact that each event is associated with a sequence of observations over a period of time, and not with a single observation as in tabular data classification. In the context of anomaly detection and prediction, the situation is exacerbated by the rarity of the events of interest - *e.g.* equipment failures - which is fortunate from a cost perspective, but problematic from the learning point of view, because the resulting imbalanced data used for training hamper accurate predictions.

Moreover, time series classification is even harder with extremely few labels, a common problem in Machine Learning (ML) addressed by Semi-Supervised Learning (SSL). SSL algorithms have shown good performance on graph-based datasets such as citation networks [5], [6] and on images datasets [7], [8]. However because of the time correlation, applying common SSL algorithms to time series is difficult. To compensate, some SSL algorithms are based on complicated neural network models, developed explicitly for this type of data [2], [9], [10]. These models are not always responding to the rising demand for interpretability which has been the focus of considerable research [11], mostly concerned with the interpretation of "black-box" models such as deep neural networks and, to a lower extent, random forests [12]. Interpretability can be enhanced by symbolic regression (SR) algorithms, which link the input features to the target with explicit mathematical formulae, thus providing "model-based" interpretability. SR is mostly treated from a genetic programming perspective [13], although attempts from a more traditional ML angle have been made [14]. However, there are very few works on SR in semi-supervised mode [15].

This paper brings the following contributions: we propose a new framework called PaZoe based on the stacking of two recent algorithms that have shown good performances in previous work, namely PageRank & PCA [16] (enabling self-labelling [17]) and Zoetrope Genetic Programming [18], and we adapt this framework to sensor data; we demonstrate that our framework outperforms the state-of-the-art linear and neural network algorithms for SSL in terms of accuracy on different time series datasets; in addition to public domain gesture datasets, we generated a realistic dataset based on a DC motor for the classification of the type of motor imbalance at different rotation speeds.

II. Context

A. Problem and notations

Let $X = [X_i]_{i=1}^n \in \mathcal{R}^{n \times d}$ be the matrix of input features, with dimension d and total number of observations n. Then let $\{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ be the set of k classes, and $Y = [Y_i]_{i=1}^n$ be a label matrix where $Y_i = (Y_{i,j})_{j=1}^k$, such that $Y_{i,j} = 1$ if $X_i \in \mathcal{C}_j$ and $Y_{i,j} = 0$ otherwise. Y is composed of two parts: a labelled one of size n_l, and an unlabelled one of size n_u, typically for SSL $n_l \ll n_u$ and Y_i being the null vector for all unlabelled data. We also define the following graph-based setup which will be used in the sequel: $A = [A_{i,j}]_{i,j=1}^{n,n}$ is an adjacency matrix, $D = \text{diag}(D_{i,i})$ is a diagonal matrix with $D_{i,i} = \sum_{j=1}^n A_{i,j}$. The problem of semi-supervised classification is to find an accurate classification result $\hat{Y} = [\hat{Y}_i]_{i=1}^n$ for Y, with $\hat{Y}_i = (\hat{Y}_{i,j})_{j=1}^k$, based on both labelled and unlabelled data where the amount of labelled data is extremely low.

B. Related works

The research in SSL is split into two main areas, according to the structure of the data. **Graph-based datasets** display both node features and a graph structure. Here, the state-of-the-art (SOTA) algorithms are based on graph propagation strategy, among which Label Propagation (LP) [19], Graph Convolution Network (GCN) [5]; Instead, **Non-graph based datasets** only have object features, such as pixels for images. Here SOTA SSL algorithms are based on the application of semi-supervised regularisation and similarity learning, such as transductive SVM (TSVM) [7], logistic regression (LR) [20], K-nearest neighbours (KNN) [8], and GEML [21].

These SSL algorithms have been developed according to the dataset's nature. As a consequence, algorithms applicable to graph-based datasets (e.g. GCN [5], Label Propagation [19]) are not suitable to non-graph based datasets, and conversely for non-graph based algorithms like LR [20] or GEML [21]. Note that a semi-supervised genetic programming like GEML is comparable only with linear algorithms and outperforms them in the supervised regime most of the time.

Finally, all the aforementioned SSL algorithms do not provide a clear interpretation of the classification results.

In order to address these issues, we propose a combination of a linear algorithm for graph-based SSL (PageRank & PCA (PRPCA)) with a non-linear symbolic regression algorithm, ZGP (Zoetrope Genetic Programming). The interest in PRPCA comes from its applicability to both graph-based and non-graph based datasets, while ZGP keeps the classification results interpretable. We show in the experiment that the combination of PRPCA and ZGP within the PaZoe framework, significantly increases each algorithm's individual performance, and outperforms SOTA algorithms on several time series datasets.

III. PaZoe framework

In our framework, we assume that any data can be represented through a graph structure. Since PRPCA outperforms the linear graph-based as well as non-graph based SSL algorithms, we use it in PaZoe to extend the training set to the self-labelling regime [17]. Then, we conjecture that the Zoetrope mechanism in ZGP can extract useful information by training on PRPCA predictions in a supervised regime. Based on these assumptions, we combine these two algorithms.

A. PageRank & Principal component analysis (PRPCA)

The main idea of PRPCA is to enrich the adjacency matrix A by the information of estimated covariance between objects $S \in \mathcal{R}^{n\times n}$. This enrichment allows spreading information about labelled objects to unlabelled ones. This means that even in the absence of edge between two objects where $A_{i,j} = 0$, we can still spread the information about labels between these objects weighted by their covariance value. The explicit classification solution of PRPCA is given by

$$\hat{Y} = \left(I - \alpha\left(AD^{-1} + \delta SD^{-1}\right)\right)^{-1}(1-\alpha)Y \qquad (1)$$

where $\delta \in (0, 1)$ sets the influence of S on A and $\alpha \in (0, 1)$ is the random jump parameter for PageRank. Let us note that in the normalised $\hat{Y}$ if $\left(AD^{-1} + \delta SD^{-1}\right)$ is a stochastic matrix, equation 1 is an explicit PageRank [22] problem. The classification solution 1 is obtained through the differentiation of the combination Laplacian regularization[1], supervised[2] and PCA[3] losses. Note that the computation of the matrix inversion can be avoided, thanks to numerical iterative methods [16]. PRPCA presents the following interesting and practical features: first, it has an explicit classification solution (Eq. 1) enabling the interpretation of the object's values in each column of Y as the value of its importance in that particular class/column, through the PageRank model; second, it can work in a distributed regime, handling the high amount of unlabelled data without memory issues; and finally, it can support the online learning regime, appending data from a new observed sensor as a new object in a graph and labeling it through its neighbours.

B. Zoetrope Genetic Programming

The Zoetrope Genetic Programming (ZGP) algorithm is a genetic programming approach for symbolic regression (GPSR) which iteratively evolves mathematical formulae towards the one that best fits the data. The particularity of ZGP among symbolic regression methods lies in its formula construction, which allows efficient computations and prevents models to overgrow and become complex, a common drawback in GPSR. This construction mechanism is illustrated in Figure 1 and works as follows. First, a number m_e of elements $(E_1, \ldots, E_{m_e})$ are randomly selected among input features (resp. random constants), with a 90% (resp. 10%) probability. Then, these elements undergo m_m "maturation steps" or "stages", which consists in applying the fusion operation

$$f(E_i, E_j) = r \cdot \mathrm{op}_1(E_i, E_j) + (1-r) \cdot \mathrm{op}_2(E_i, E_j),$$

on couples of elements, where op_i, $i = 1, 2$ are operators[4] uniformly chosen in a predefined set $\mathcal{O}$, and $r = U[0, 1]$; the result of $f(E_i, E_j)$ replaces either E_i or E_j. At the end of the m_m stages, the matured elements – called *"zoetropes"* and denoted by $(Z_1, \ldots, Z_{m_e})$ – are linearly combined via multinomial logistic regression penalized by Elastic net [23]; this last step allows to jointly select the most relevant zoetropes and optimally estimate their weights. The operator set can be adapted to the problem at hand, but is typically taken as $\mathcal{O} = \{+, -, \times, /, \cos, \sin, \mathrm{sqrt}\}$.

[1]Laplacian regularization: $\sum_{j=1}^n A_{i,j}||\hat{Y}_i - \hat{Y}_j||_2^2$
[2]Supervised loss: $\sum_{i=1}^n ||\hat{Y}_i - Y_i||_2^2$
[3]PCA loss: $||\bar{X}\hat{Y}||_2^2$
[4]In case op_1 or op_2 is unary, only E_i is taken into account

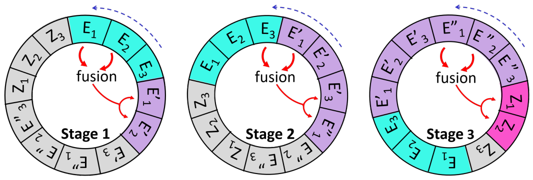

Fig. 1. Illustration of ZGP's model construction with $m_e = m_m = 3$. For the sake of readability, the third fusion, generating $(E"_2, E"_3)$ from (E'_2, E'_3) is not represented. Note that $Z_3 = E"_3$ as no element is left for a fusion.

Genetic programming considers models as individuals of a "species", and evolves them with random perturbations (*mutations*) and by mating pairs into new individuals (*crossover*). ZGP's mutation and crossover are also nonstandard in GPSR: the mutation consists in selecting couples of models, and replace the worst one with a "mutant" of the first one, while the crossover consists in selecting the best and worst in a pool of m_t models, and randomly propagate elements and fusions of the best to the worst model. Note that the "worst" and "best" models are defined with respect to their accuracy on the training set. At the end of each iteration, all the models are evaluated on the validation set, and the best ever is stored. Also, like PRPCA, ZGP can work in distributed regime. For the complete description of the algorithm, see [18].

C. PaZoe strategy

Our PaZoe framework is given in Algorithm 1 and consists in three main sequential steps:

1) *Transforming data into graph structure.* For non-graph based datasets, where no adjacency matrix A is available, we first generate a synthetic graph structure and retrieve A by K-nearest neighbours (KNN) with Euclidean distance;
2) *Labelling the unlabelled data.* We then compute PRPCA based on the input matrix X and the adjacency matrix A. Predictions generated by PRPCA consider the graph structure, which could be valuable for stacking with existing object features X for further training of ZGP. Also, self-labelling [17] by PRPCA predictions extends the training set for further ZGP training in the supervised regime;
3) *Classifying and recovering the boundary formulae.* We stack the input data X with the predictions from PRPCA and feed the augmented dataset to ZGP for supervised training (where train/test split of dataset is 70%/30%).

This framework is applicable to any kind of data. In order to adapt it to temporal data obtained from sensors, we propose to modify step 1 of PaZoe as follows: we first separately train a KNN algorithm and generate different adjacency matrices for each type of features, e.g. the magnetometer[5] and the gyroscope[6] in the DC motor dataset (see next section for details); then we linearly combine these adjacency matrices into the final one. Similarly in PRPCA, we compute the covariance between objects separately for each feature.

The outline of PaZoe with the modification for sensor data is illustrated in Figure 2. The PRPCA part of the code is publicly available through this link[7]. As for ZGP, we used an open source version of the proprietary algorithm, which is still under testing and has not been released yet.

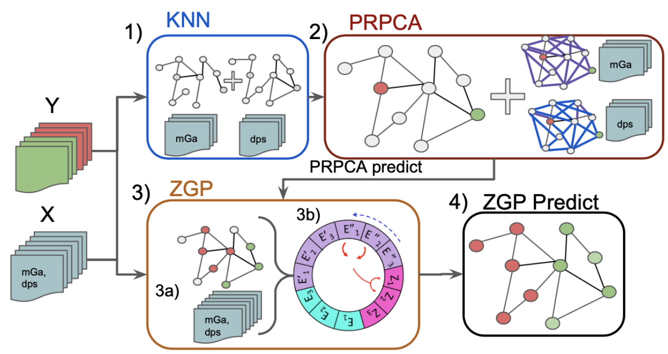

Fig. 2. PaZoe sequence: 1) Generation of graph structure; 2) Self-labelling by PRPCA; 3): 3a) Stack X with PRPCA predictions; 3b) ZGP training; 4) Final predictions from ZGP. Note, X and units therein, refer to the dc motor dataset.

INPUT: X,A, Y, α, δ;
INITIALIZE:
$\bar{X}_i^T = X_i^T - \frac{1}{d}\sum_j^d X_j^T \forall i \in (1,\dots,n)$; $S = \frac{\bar{X}^T\bar{X}}{d-1}$
IF: $A = NaN$:
$A = KNN(X)$
$\hat{Y} = \left(I - \alpha\left(AD^{-1} + \delta SD^{-1}\right)\right)^{-1}(1-\alpha)Y$
$\hat{X} = stack(X, \hat{Y})$
$\hat{Y} = ZGP(\hat{X}, \hat{Y}, m_p, m_i, m_e, m_m, m_t)$

Algorithm 1: PaZoe

IV. Experiments

We apply the PaZoe framework on three time series datasets, the first generated for this work, the others obtained from the public domain. **DC motor dataset (RPM)** – generated with six classes of imbalance failure on a real motor, by collecting data from a sensor tile (see next section); **UWaveGesture (UWave)** [4] – with eight classes of gestures from (x, y, z) accelerometer features; **Gesture WIImote (WII)** [3] – with ten classes of gestures from (x, y, z) accelerometer features by Nintendo Wiimote.

A. DC motor data collection

In order to profit from a real dataset on motor failures, we conducted our own experiment to simulate anomalies of DC motors in a production environment. These are later used as classification targets with labelled data generated for training. Motor axis imbalance were generated by loading weights onto a disk plate mounted on top of the motor at varying distances

[5] $X_{mga} \in \mathcal{R}^{n \times d_{mga}}$ where d_{mga} is dimension of magnetometer
[6] $X_{dps} \in \mathcal{R}^{n \times d_{dps}}$ where d_{dps} is dimension of gyroscope
[7] https://github.com/KamalovMikhail/PaZoe

from its axis. The dataset was obtained with industrially graded equipment made of a STMicroelectronics (STM) acquisition board[8], a STM SensorTile with three sensors - accelerometer, magnetometer and gyroscope - and a SD card for data storage. The three components (x, y, z) of each sensor signal were acquired at the default rate of 20 Hz, kept throughout. We recorded three rotation speeds, 620, 420 and 220 RPM. We chose these speeds to show how the performances tend to drop the lower the speed, making the discrimination of anomalies more difficult.

The three sensor quantities and units are as follows:

- **Accelerometer (mg)** - acceleration values in units of mg, where g = 9.81 m/s^2 is the gravity acceleration;
- **Magnetometer (mGa)** - generally used for tracking of moving objects - with values in mGa, where 'Ga' means gauss and 1 $gauss = 10^{-4} T$;
- **Gyroscope (DPS)** - measures rotations in DPS (deg. per seconds), e.g., one needs to convert to rad/s if time or geometric calculations are needed.

The duration of each experiment is close to one ($\sim$ 1) minute.[9]

B. Data utilization strategy

We used the following train/test split strategy for all of these datasets: 20 labelled objects for each class for training and the rest for testing. Note that all these datasets are balanced, e.g. the number of objects in each class is similar. This strategy is standard for SSL learning algorithms [5], [19]. Also, we have to mention that for the DC motor dataset, we considered objects as sensor quantities (e.g. accelerometer, magnetometer, gyroscope) at each moment in time (recording individual data points). In other words, the length of time series (l) for the DC motor dataset was equal to $l = 1$. In practice, it allows us to check the motor's state and signal imbalance failures at any moment. This is because the position of the motor is stable but, at a successive time instant it might not be.

Since WII and UWave datasets have only observations from accelerometers, we considered an object as a time series with length equal to the motion's length (e.g. following the time evolution of the three different coordinates, (x, y, z) during the complete gesture recording). These three datasets, summarised in Table I, and the code for their processing are available through the provided link[10]. Note that the number of observation for the RPM dataset slightly differs depending on the speed, due to the presence of missing values (especially at the end of each observation time).

C. Results

For a fair comparison, we used three types of algorithms: (1) SSL graph-based such as LP, PRPCA and GCN (is a neural netowrk); (2) SSL non-graph based, such as LR and KNN; and (3) supervised algorithms such as SVM, ZGP and the combination of algorithms such as PRPCA & LR (PaLR) and PRPCA & SVM (PaSVM). For each of these algorithms, we took the best hyperparameters defined in their respective works and for PRPCA we used $\alpha = 0.9$, $\delta = 10^{-3}$. We use accuracy as the performance metric since all datasets are balanced. We report the average accuracy on the test set, taken over 20 random splits (k-folds strategy).

The results of PaZoe on the DC motor dataset obtained with various features combinations are presented in Table II. It shows that the best classification accuracy is achieved by using magnetometer (mGa) or gyroscope (dps) with respect to RPM. Since magnetometer (mGa) and gyroscope (dps) separately provide a high classification accuracy for the DC motor dataset, we use the best of them for each RPM (ie. dps for 620, 420 rpm and mGa for 220 RPM) to train the rest of the algorithms. The results of PaZoe compared with all the other algorithms on the DC motor dataset are shown in Table III, along with the performance on the WII and UWave datasets. Several comments can be made on those results: first, PRPCA clearly outperforms the other SSL algorithms on all three datasets; second, combining PRPCA with a supervised classification algorithm only leads to an improvement with ZGP (PaZoe); third, PaZoe considerably outperforms its separate components (PRPCA, ZGP) as well as the rest of the SSL and supervised algorithms on all three datasets, even with only one sensor (accelerometer) in the gesture datasets.

TABLE I
DATASET STATISTICS

		620,420,220 RPM	UWAVE	WII
n	No. observations	$\sim$6100	4478	1000
n_l	No. labels	120	160	200
n_l/n	Ratio of labels	$\sim$ 1.9%	3.6%	20%
l	Sequence length	1	315	326
k	No. classes	6	8	10
d	No. features	9	315	326

TABLE II
ACCURACY FOR THE DC MOTOR DATASET WITH VARIOUS FEATURE SETS

RPM	Algorithm	dps, mGa, mg	mGa	mg	dps	mGa,dps
620	PRPCA	61.2	19.2	44.2	71.6	68.2
	ZGP	48.9	16.2	35.7	60.1	63.2
	PaLR	18.4	17.2	19.5	42.9	18.8
	PaSVM	46.7	17.0	41.2	65.8	66.4
	PaZoe	**65.6**	**97.0**	**96.8**	**98.8**	**79.3**
420	PRPCA	38.8	60.8	28.3	66.2	51.8
	ZGP	62.4	64.6	29.2	62.3	65.2
	PaLR	18.0	28.7	22.7	35.2	17.9
	PaSVM	18.7	51.1	26.0	52.5	44.2
	PaZoe	**63.5**	**96.2**	**95.2**	**97.8**	**67.2**
220	PRPCA	30.6	66.3	20.1	29.5	37.2
	ZGP	20.2	18.5	16.8	26.1	27.1
	PaLR	18.5	16.4	17.1	19.0	17.4
	PaSVM	19.8	61.2	18.8	16.5	33.0
	PaZoe	**36.2**	**94.2**	**90.6**	**93.1**	**44.8**

[8]Nucleo G431RB ST L6230 with a GBM2804H brushless motor

[9]Each datapoint has a timestamp dd/mm/yyyy hh:mm:ss.xxx, with differences between adjacent points from 2 to 5 ms around the nominal 50 ms. As the time scale is approximately uniform, the absolute value of the time can be safely ignored and timestamps swapped for indexes as necessary.

[10]https://github.com/KamalovMikhail/PaZoe

TABLE III
ACCURACY FOR DC MOTOR, WII AND UWAVE DATASETS

Dataset	620RPM	420RPM	220RPM	WII	UWave
PRPCA	71.6	66.2	66.3	67.8	70.1
LP	31.2	17.2	16.6	15.2	12.4
KNN	28.6	33.9	60.1	23.7	58.8
GCN	16.9	21.6	18.3	16.7	18.3
ZGP	60.1	62.3	26.1	14.6	17.4
LR	29.7	27.9	16.8	52.9	55.8
SVM	64.1	38.9	25.6	43.3	68.3
PaLR	42.9	35.2	16.4	34.9	62.8
PaSVM	65.8	52.5	61.2	37.3	69.1
PaZoe	**98.8**	**97.8**	**94.2**	**71.8**	**72.3**

V. Conclusions

The problem of label scarcity in data gathered from industrial equipment under working conditions is addressed by generating labels via an efficient SSL algorithm (PRPCA). Its outcomes are then fed into the GPSR based algorithm ZGP, which provides interpretable predictions expressed by a mathematically explicit formula. The working of the two algorithms have been briefly explained and their joint use described as the PaZoe framework. It has been shown that the use of this stacked framework provides a combined performance which overcomes the two algorithms individually. These results were obtained on realistic data, partly generated for this purpose with industrially graded equipment, partly on sensor data available from the public domain. We observe that, similarly to other SSL algorithms (like LP, GCN, KNN) PaZoe does not assume any kind of data distribution (or even require the data to be i.i.d), while it performs better than those.

In terms of potential future work, we want to evaluate PaZoe for handling the case when we lose/add some features (channels) from a sensor tile during the training. In particular, we will test the use of pre-trained ZGP models when adding/removing a sensor or when handling data streams (e.g. online learning). Also, we want to develop a non-linear distributed version of PRPCA to improve the self-labelling for PaZoe. Finally, for practical implementation and simpler parametrisation, it would be interesting to directly include PRPCA into ZGP instead of running the two algorithms separately in sequence.

Acknowledgment

This work was made possible by the financial support of MyDataModels to the first author. We also acknowledge the technical support of ST Microelectronics, for kindly providing some of the material used in the experiments and for suggestions about their use.

References

[1] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," Sustainability, vol. 12, no. 19, p. 8211, 2020.

[2] G. Safont, A. Salazar, J. Belda, and L. Vergara, "Semi-supervised learning for dynamic modeling of brain signals during visual and auditory tests," in 26th European Signal Processing Conference (EUSIPCO). IEEE, 2018, pp. 1192–1196.

[3] J. Guna, I. Humar, and M. Pogačnik, "Intuitive gesture based user identification system," in 35th International Conference on Telecommunications and Signal Processing (TSP). IEEE, 2012, pp. 629–633.

[4] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," Pervasive and Mobile Computing, vol. 5, no. 6, pp. 657–675, 2009.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in 5th International Conference on Learning Representations (ICLR), 2017.

[6] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," in Proc. of the 33rd International Conference on Machine Learning (ICML), 2016, pp. 40–48.

[7] T. Joachims, "Transductive inference for text classification using support vector machines," in Proc. of the 16th International Conference on Machine Learning (ICML), 1999, pp. 200–209.

[8] E. Fix, Discriminatory analysis: nonparametric discrimination, consistency properties. USAF school of Aviation Medicine, 1951.

[9] A. Diment, T. Heittola, and T. Virtanen, "Semi-supervised learning for musical instrument recognition," in 21st European Signal Processing Conference (EUSIPCO). IEEE, 2013, pp. 1–5.

[10] S. R. Chetupalli, A. Gopalakrishnan, and T. V. Sreenivas, "Feature selection and model optimization for semi-supervised speaker spotting," in 24th European Signal Processing Conference (EUSIPCO). IEEE, 2016, pp. 135–139.

[11] J. W. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Definitions, methods, and applications in interpretable machine learning," Proc. of the National Academy of Sciences, vol. 116, no. 44, pp. 22 071–22 080, 2019.

[12] C. Molnar, "Interpretable machine learning-a guide for making black box models explainable. lulu (2019)," 2020.

[13] M. Virgolin, T. Alderliesten, C. Witteveen, and P. A. N. Bosman, "Improving model-based genetic programming for symbolic regression of small expressions," Evolutionary Computation, p. 1–27, Jun 2020.

[14] T. McConaghy, "FFX: Fast, scalable, deterministic symbolic regression technology," in Genetic Programming Theory and Practice IX. Springer, 2011, pp. 235–260.

[15] Y. M. El Hasadi and J. T. Padding, "Solving fluid flow problems using semi-supervised symbolic regression on sparse data," AIP Advances, vol. 9, no. 11, p. 115218, 2019.

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

[17] 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.

[18] 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.

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

[20] M.-R. Amini and P. Gallinari, "Semi-supervised logistic regression," in Proc. of the 15th European Conference on Artificial Intelligence (ECAI), 2002, pp. 390–394.

[21] J. M. Fitzgerald, R. M. A. Azad, and C. Ryan, "GEML: Evolutionary unsupervised and semi-supervised learning of multi-class classification with grammatical evolution," in 7th International Joint Conference on Computational Intelligence (IJCCI), vol. 1. IEEE, 2015, pp. 83–94.

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

[23] 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.