Bayesian Feature Discovery for Predictive Maintenance

Amir Dib, Charles Truong, Laurent Oudre, Mathilde Mougeot, Nicolas Vayatis, Heloïse Nonne

Bayesian Feature Discovery for Predictive Maintenance

Amir Dib[†], Charles Truong[†], Laurent Oudre[†], Mathilde Mougeot[†], Nicolas Vayatis[†], Heloïse Nonne[‡].
[†]Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, 91190, Gif-sur-Yvette, France
[‡]ITNOVEM, SNCF, 93120, Saint-Denis, France

***Abstract*—This paper considers predictive maintenance, which is the task of predicting rare and anomalous events (typically, system failures) using event logs data, which are series of time-stamped symbolic codes emitted at regular or irregular intervals by a monitored system. Our objective is to find small sets of codes (called itemsets or patterns) that occur shortly before failures. Current prediction methods either produce patterns at a high computational cost or resort to kernel approaches which are often difficult to interpret. We introduce Bayesian Pattern Feature Discovery (BPFD), a new generic algorithm for pattern discovery. Our method, based on a pattern mining technique, produces informative and explainable features and is computationally efficient. The performance of BPFD is highlighted on real-world data sets, showing that enriching the feature space with the discovered patterns improves significantly the prediction power of a broad range of predictors and offers useful insight on the predictive maintenance task.**

***Index Terms*—Bayesian learning, pattern mining, predictive maintenance, variational inference.**

I. Introduction

Predictive Maintenance (PM) aims to anticipate critical failures of large industrial systems to plan early and cost-effective interventions. Since maintenance can amount to 15% to 70% of the total operational cost [1], PM is an important task to study, with far-reaching applications for the maintenance management of a number of industrial structures: transportation network [2], power equipment [3], factory plant [4]. Many fault-predicting procedures are based on event logs that provide information on the monitored system's health status. Event logs typically consist of event codes emitted at regular or irregular intervals. Formally, such data can be seen as temporal point processes of symbols taken from a finite dictionary. In that context, PM essentially amounts to identifying characteristic sequences (or patterns) of symbols that occur shortly before failures. The management of a railway fleet illustrates particularly well the importance of PM. SNCF, France's main railway company, uses event logs to predict failures of the train door system, one of the most critical equipments of its rolling stock. Any malfunction leads to the complete immobilization of the train and propagates delays to a large portion of the transportation network.

This work's main driver is to design an interpretable and efficient machine learning pipeline to detect potential occurrences of breakdowns of rolling stocks. The prediction procedure uses event logs, which are time-stamped error codes e_t taken from a dictionary E of d distinct codes. These events are collected and processed by on-board equipment according to dedicated rules to which the end-user does not have access. These codes are produced during events deemed relevant by the manufacturer (for instance exceeding the threshold of an electrical signal or a malfunction).

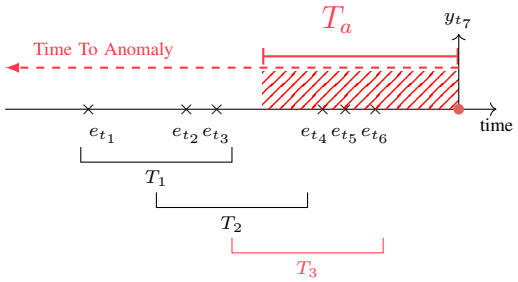

Fig. 1. Temporal aggregation of log-events (e_{t_1}, …, e_{t_6}) over sliding windows (T_1, T_2, T_3). In red, events that occur in the period T_a before y_{t_7} are considered anomalous and labeled $l = 1$. The aggregation produces the itemsets $x_1 = \{e_{t_1}, e_{t_2}, e_{t_3}\}, x_2 = \{e_{t_2}, e_{t_3}\}, x_3 = \{e_{t_4}, e_{t_5}, e_{t_6}\}$ and the labels $l_1 = 0$, $l_2 = 0$ and $l_3 = 1$. The goal is to correctly predict the labels l_i from the itemsets x_i.

Procedures that make use of log events are particularly challenging since there is no natural order or distance on the space of symbols, thus making most machine learning models unsuitable. This issue can be overcome by kernel methods [5] but these approaches are difficult to interpret, which is a requirement for a predictive solution to be used in an industrial context. Another common strategy consists in transforming the prediction task into a binary classification task. In a nutshell, the signal is aggregated over sliding temporal windows (possibly overlapping) of fixed size. Features are simply the set of collected events within the window (called itemsets). For a given user-defined threshold period $T_a > 0$, a window is considered as anomalous (label "1") if it contains codes emitted in the period T_a before a failure, and normal (label "0") otherwise. This aggregation procedure is schematically illustrated on Figure I. Even though popular [6], classification based solely on this construction is often unable to capture critical patterns of events that can be highly relevant in PM.

To tackle this issue, one can resort to methods from the related domains of Frequent Itemset Mining (FIM) and Discriminative Pattern Mining (DPM). FIM is the task of finding the most common patterns of a set in an exponentially large class of all possible combinations [7]. A famous application is the shopper recommendation problem, where the goal is to find the most common products that are bought together.

DPM aims at searching for the set of patterns that best differentiate two subsets of a data set in the sense that a pattern occurs significantly more frequently in one of the classes. This framework has many applications such as consumer behavior analysis, RNA and DNA gene expression, subgraph mining, and anomaly detection. Generally, DPM algorithms start with a FIM step, where the most frequent itemsets are identified, then compute a statistical test for each itemset to determine if its presence is significantly different between two subsets [8]. This often leads to an exponential number of statistical tests to perform and make many DPM methods computationally intensive.

In this work, we propose a Bayesian approach to explore the space of frequent itemsets in an efficient way. More precisely, we use a Bayesian Mixture Model to infer with a low computational cost the both frequent and discriminative itemsets. Also, we offer empirical proof of the general use of such discriminative patterns by considering them as features for the PM task. This results in a method that can extract an interpretable set of attributes and significantly improve any PM algorithm. Moreover, the Bayesian generative model allows for computing the posterior distribution and estimating the confidence intervals. Finally, additional expert-knowledge can be naturally introduced in the model *via* the choice of prior [9].To the extent of our knowledge (and as pointed in [8]), it is the first Bayesian approach towards DPM, and there has been no investigation of using pattern discovery methods based on discriminant pattern to the Predictive Maintenance task.

In Section II, the basic concepts of FIM are introduced. Section III presents our approach to the DPM problem and application to signals of log events. The experiments are described and commented in Section IV.

II. Background

This section introduces the concepts and main approaches of FIM and DPM.

A. Frequent Itemset Mining

Let $E = (e_1, \dots, e_d)$ the base dictionary of events and $\mathcal{E} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E. The windowing procedure described in Fig. I transforms the sequence of log events into a database $\mathcal{D} = \{(x_i, l_i)_{i=1}^n\}$ of elements of $\mathcal{E} \times \{0, 1\}$ with the binary variable l indicating if a breakdown event occurred soon after the code emission. Note that the set $\mathcal{E}$ can be identified with the d-dimensional hypercube $\mathcal{X} = \{0, 1\}^d$, leading to the equivalence with the binary representation described in Fig. 2. We also denote $\mathcal{D}_0$ (respect $\mathcal{D}_1$) the samples in $\mathcal{D}$ associated with the target value $l = 0$ (respect $l = 1$) so that $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1$.

The support of a pattern $x \in \mathcal{E}$ is defined as the number of samples of the database in which any pattern greater (with respect to $\subseteq$) than x appears. Formally,

$$s(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{x \in \{z \in \mathcal{E} | x_i \subseteq z\}}. \tag{1}$$

TABLE I
Contengency table for a pattern E and a database $\mathcal{D} = \mathcal{D}_0 \cup D_1$ to compute p_F.

	x	x^c	Size
$\mathcal{D}_1$	$s_1(x)$	$\lvert\mathcal{D}_1\rvert - s_1(x)$	$\lvert\mathcal{D}_1\rvert$
$\mathcal{D}_0$	$s_0(x)$	$\lvert\mathcal{D}_0\rvert - s_0(x)$	$\lvert\mathcal{D}_0\rvert$
Column totals	$s(x)$	$n - s(x)$	n

In the same fashion, we denote $s_j(x)$ the support of the pattern $x \in \mathcal{E}$ in $\mathcal{D}_j$. In the context of predictive maintenance, $s_1(x)$ represents the number of times that a pattern of events appears close to a breakdown. Given a threshold $\mu \in [0, 1]$, the FIM task consists of finding the collection $\mathcal{TH}(\mathcal{E}, \mathcal{D}, \mu)$ of all *frequent patterns* in $\mathcal{E}$ defined has having support greater or equal than μ. The computation of such a collection is challenging since any algorithm has to explore a space size of $|\mathcal{E}| = 2^d$ elements and will exhibits exponential complexity $\mathcal{O}(n2^d)$. The key for pruning the set of possible patterns is the anti-monotonicity constraint which states that every sub-pattern of a frequent pattern is frequent. This approach spans a class of problems referred to as the Frequent Itemset Mining algorithms that can be used to extract $\mathcal{TH}(\mathcal{E}, \mathcal{D}, \mu)$ at reasonable computational cost [7], [10].

B. Discriminative Pattern

The classical DPM pattern procedure requires to perform a FIM procedure as described in section II-A to obtain $\mathcal{TH}(\mathcal{E}, \mathcal{D}_0, \mu)$ and $\mathcal{TH}(\mathcal{E}, \mathcal{D}1, \mu)$ and compute the *contingency table* [8]. Table I describes the complete contingency table for a pattern $x \in \mathcal{E}$ as the record of the support of x and x^c (which is the complementary pattern such that $x \cup x^c = E$) in $\mathcal{D}_0$ and $\mathcal{D}_1$. For instance, Fig. 2 displays the occurrence of each code in E in the sample i aggregated over the window T_i. The pattern $x = \{e_7, e_8\}$ produces a contingency table with $s_0(x) = s_1(x)$. Since the data set $\mathcal{D}$ is the result of a stochastic process, one needs to design a statistical test to evaluate the statistical significance of the discrepancy between $s_0(x)$ and $s_1(x)$. The hypergeometric model with a Fisher test is the most commonly used framework for finding statistically significant pattern. Under the null hypothesis, the probability of observing the contingency table associated with x with $s_1(x) = a$ is

$$p_F(a) = \frac{\binom{|\mathcal{D}_1|}{a}\binom{|\mathcal{D}_0|}{s(x)-a}}{\binom{n}{s(x)}}. \tag{2}$$

The p-value is then obtained as the probability of observing a contingency table at least as extreme as the observed one. Since, in the worst case, a number of 2^d patterns must be considered, the probability of false discovery increases drastically and requires corrections. This is the goal of recent work on DPM algorithm such as LAMP and SPuManTe [11].

Nevertheless, all the above methods require the costly computation of $\mathcal{TH}(\mathcal{E}, \mathcal{D}_0, \mu)$ and $\mathcal{TH}(\mathcal{E}, \mathcal{D}_1, \mu)$ and can be challenging to interpret as the choice of the threshold for the p-value is a notoriously difficult problem that leads to misuses [12].

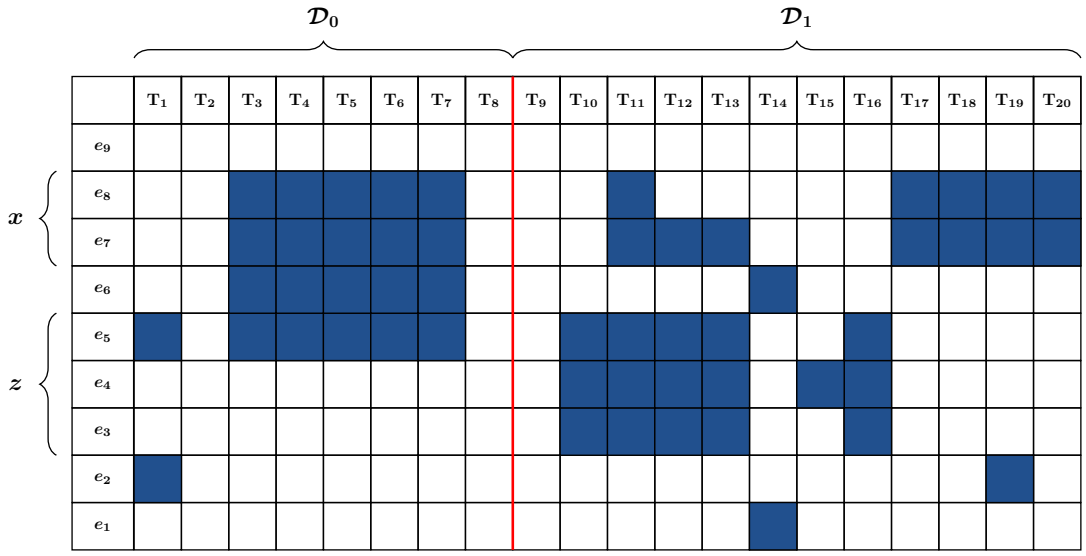

Fig. 2. An example data set of events $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1$. Row corresponds to items in $E = (e_1, \ldots, e_9)$ and columns to $n = 20$ samples. A blue colored area indicates that the item is present in the sample column considered. In this data set, the pattern $x = \{e_7, e_8\}$ in $\mathcal{E}$ seems to be nondiscriminative since $s_0(x) = s_1(x)$. On the contrary, the pattern $z = \{e_3, e_4, e_5\}$ appears to be specific to the positive class $l = 1$.

III. Method

This section introduces a new Bayesian approach for the DPM problem and its application to the signal of log events.

A. Bayesian interference for pattern discovery

Once the signal of error codes has been processed according to the procedure described in Fig. I, we need to choose a generative model for the pattern database $\mathcal{D}$. We believe that a good trade-off is achieved between generality and complexity with a model assuming that the training data set is the result of a Bayesian Mixture Model (BMM) process with K mixture components [13]. This model assumes conditional independence given the mixture class and that the database is the result of sampling from multiple distributions p_k. The final number of parameters to evaluate for a K Bayesian Mixture Model is $K \times d$. We stress out that the choice of K controls the complexity of the model. Taking the number of components K to be large approximates the most exhaustive choice, which is the fully correlated Bernoulli model with 2^d parameters and is computationally intractable for even a moderate dimension d. The simple case of $K = 1$ is the independent and homogeneous Bernoulli model with $i.i.d.$ samples. Simple combinatory calculus gives a support function which only depends on the length of the pattern. Intuitively, it is similar to the experiment of throwing d identical coins with probability θ_0 and computing the probability of a given arrangement with given a number of heads. The too simple previous model assumes interchangeability on the elements e_i, complete independence between them and a similar distribution for all samples of the training data set. In the use case of DPM, this approach has the advantage of allowing computation of any quantity of interest; one computation is needed to infer the parameters and all conclusions can be drawn from it by sampling the posterior predictive distribution. The following gives a formal definition of the model.

Let $\mathrm{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ be an i.i.d.sample of the pattern in the binary labeled database $\mathcal{D} = \{(\boldsymbol{x_i}, l_i)\}_{i=1}^n$ with $\boldsymbol{x_i} = (x_{ij})_{j=1}^d$ elements of $\{0,1\}^d$ and suppose the underlying model is a BMM with K components. For $k \in \{1, \ldots, K\}$, the k-ith sampling distribution $p_k(\boldsymbol{x_i}|\boldsymbol{\theta_k})$ depends only on the parameter $\boldsymbol{\theta_k} = (\theta_{kj})_{j=1}^d$. Denoting λ_k the probability of sampling from the k-th component with $\sum_{k=1}^K \lambda_k = 1$, the global sampling distribution writes

$$p_(\boldsymbol{x_i}|\Theta, \boldsymbol{\lambda}) = \sum_{h=1}^{K} \lambda_k p_k(\boldsymbol{x_i}|\boldsymbol{\theta_k}), \tag{3}$$

where $\Theta = (\boldsymbol{\theta_k})_{k=1}^K$ and $\boldsymbol{\lambda} = (\lambda_k)_{k=1}^K)$. The conditional independence hypothesis for each Bernoulli component applied to the mixture distribution p_k leads to

$$p_k(\boldsymbol{x_i}|\boldsymbol{\theta_h}) = \prod_{j=1}^{d} \theta_{kj}^{x_{ij}} (1 - \theta_{kj}^{x_{ij}}).$$

Since it is unknown to which component $k \in \{1, \ldots, K\}$ a sample i belongs to, it is needed to introduce the unobserved indicator w_{ik} defined by

$$w_{ik} = \begin{cases} 1 & \text{if sample } i \text{ drawn from the } k\text{-th component,} \\ 0 & \text{otherwise.} \end{cases}$$

Knowing the mixture component parameter $\boldsymbol{\lambda}$, the component indicator $\boldsymbol{w_i} = (w_{i1}, \ldots, w_{iK})$ for the sample i is

TABLE II
TEST ACCURACY, RECALL AND AUC 10× CROSS-VALIDATED FOR BPFD AND BC CLASSIFIERS ON DATASETS REPORTED IN TABLE III.

	X Gradient Boosting			Random Forest			Light Gradient-Boosting Machine			Categorical Boosting			Linear Regression			k-Nearest Neighbors		
	BC	PF	BPFD	BC	PF	BPFD	BC	PF	BPFD	BC	PF	BPFD	BC	PF	BPFD	BC	PF	BPFD
ijcnn1																		
AUC	0.728	0.769	**0.927**	0.726	0.767	**0.913**	0.732	0.769	**0.926**	0.727	0.768	**0.927**	0.714	0.732	**0.899**	0.614	0.643	**0.841**
Accuracy	0.906	0.907	**0.929**	0.906	0.907	**0.928**	0.906	0.907	**0.929**	0.906	0.907	**0.93**	0.905	0.905	**0.918**	0.89	0.897	**0.922**
Recall	0.0398	0.0465	**0.403**	0.0411	0.0479	**0.416**	0.0238	0.0372	**0.401**	0.0413	0.0474	**0.407**	0	0.0002	**0.245**	0.106	0.105	**0.419**
F1	0.0742	0.0862	**0.519**	0.0762	0.0885	**0.523**	0.0455	0.0702	**0.516**	0.0765	0.0877	**0.523**	0	0.0003	**0.362**	0.154	0.16	**0.505**
cod-rna																		
AUC	0.776	0.496	**0.815**	0.776	0.496	**0.815**	0.776	0.496	**0.815**	0.776	0.496	**0.815**	0.765	0.495	**0.813**	0.706	0.5	**0.764**
Accuracy	0.718	0.667	**0.775**	0.718	0.667	**0.775**	0.717	0.667	**0.775**	0.718	0.667	**0.775**	0.713	0.667	**0.774**	0.688	0.591	**0.739**
Recall	**0.588**	0	0.383	**0.585**	0	0.386	**0.592**	0	0.384	**0.588**	0	0.384	**0.512**	0	0.364	0.483	0.231	**0.516**
F1	**0.581**	0	0.532	**0.58**	0	0.534	**0.583**	0	0.532	**0.581**	0	0.532	**0.544**	0	0.518	0.503	0.263	**0.568**
a9a																		
AUC	0.89	**0.896**	0.88	0.863	0.869	**0.875**	0.894	0.9	**0.903**	0.894	0.9	**0.904**	0.893	0.902	**0.902**	0.837	0.848	**0.85**
Accuracy	0.841	0.844	**0.846**	0.825	0.826	**0.829**	0.844	0.846	**0.849**	0.844	0.847	**0.848**	0.841	**0.849**	0.847	0.817	**0.826**	0.824
Recall	0.597	0.604	**0.615**	0.564	**0.582**	0.578	0.606	0.613	**0.626**	0.595	0.606	**0.611**	0.581	**0.611**	0.604	0.566	0.584	**0.589**
F1	0.643	0.649	**0.658**	0.607	0.616	**0.619**	0.651	0.656	**0.666**	0.646	0.654	**0.66**	0.637	**0.659**	0.655	0.597	0.616	**0.617**
Doors																		
AUC	0.707	0.691	**0.736**	0.713	0.707	**0.753**	0.706	0.697	**0.739**	0.722	0.715	**0.749**	0.635	0.629	**0.637**	0.557	**0.574**	0.574
Accuracy	0.643	0.629	**0.679**	0.655	0.645	**0.686**	0.647	0.637	**0.681**	0.663	0.657	**0.684**	**0.6**	0.592	0.597	0.546	**0.551**	0.551
Recall	0.614	0.608	**0.642**	0.594	0.585	**0.608**	0.595	0.577	**0.619**	0.569	0.56	**0.592**	0.652	**0.674**	0.648	**0.545**	0.526	0.526
F1	0.632	0.62	**0.667**	0.632	0.622	**0.659**	0.627	0.613	**0.66**	0.627	0.619	**0.652**	0.62	**0.623**	0.617	**0.545**	0.539	0.539

thus distributed as $\mathrm{Multin}(\boldsymbol{\lambda})$. Finally, the joint distribution is derived as

$$\begin{aligned} p(\mathrm{X}, \mathrm{W}|\Theta, \boldsymbol{\lambda}) &= p(\mathrm{W}|\boldsymbol{\lambda}) p(\mathrm{X}|\mathrm{W}, \Theta) \\ &= \sum_{k=1}^{K} \lambda_k \prod_{i=1}^{n} p_k(\boldsymbol{x_i}|\boldsymbol{\theta_k})^{w_{ik}}. \end{aligned}$$

The last step is to choose a proper prior distribution on the parameters. The natural choice [9] is to respectively set a Beta and Dirichlet distribution for the mixture probability of occurrence Θ and the mixture parameters vector $\boldsymbol{\lambda}$. For a set of parameter $\Gamma = (\Theta, \boldsymbol{\lambda}, K)$ associated with the Bayesian Mixture Model $\mathcal{M}$ is summarized as follow

$$\begin{aligned} \boldsymbol{\lambda}|\boldsymbol{\alpha} &\sim \mathrm{Dirichlet}\,(\boldsymbol{\alpha})\,, \\ \boldsymbol{w_i}|\boldsymbol{\lambda} &\sim \mathrm{Multin}(\boldsymbol{\lambda}), \\ \theta_{kj}|\boldsymbol{\beta}, \boldsymbol{\gamma} &\sim \mathrm{Beta}(\boldsymbol{\beta}, \boldsymbol{\gamma}), \\ x_{ij}|\theta_{kj} &\sim \mathrm{Bernoulli}(\theta_{kj}). \end{aligned} \tag{4}$$

B. The BPFD algorithm

The BPFD algorithm is based on choosing the model described in Section III-A as a generative model for the samples $\mathcal{D}$ and computing the *odd ratio support* to compare the patterns between classes. The steps are described in the following.

a) ***Preprocessing****:* The first step is to transform the sequential data to a binary matrix as described in Fig. I. Note that any continuous feature can be transformed into a multi-categorical feature.

b) ***Inference****:* Set the hyperparameter $\boldsymbol{\alpha} = (\frac{1}{K}, \ldots, \frac{1}{K})$. An Expectation Minimization [14] procedure is performed on $\mathcal{D}_0$ and D_1 to infer the set of parameters Γ_0 and Γ_1 associated with the models $\mathcal{M}_0$ and $\mathcal{M}_1$.

c) ***Discriminant Pattern computation****:* The discriminative power of a pattern $x \in \mathcal{E}$ is evaluated through the odd ratio support

$$r(y) = \frac{p(\mathcal{M}_1 \mid x)}{p(\mathcal{M}_0 \mid x)} \tag{5}$$

$$= \frac{p(\mathcal{M}_1)}{p(\mathcal{M}_0)} \times \frac{p(x \mid \Gamma_1)}{p(x \mid \Gamma_0)}. \tag{6}$$

d) ***Classification****:* The best discriminative patterns are then added to the original training data set $\mathcal{D}$ and classification is performed.

The main advantage of this automatic feature extraction method is that it can be applied to any data and will return new features that will often be easy to interpret. The method does not require a threshold μ and can thus discover patterns that the traditional approach would not explore. Additionally, since the posterior sampling distribution can be simulated thanks to 4, the confidence interval on the value of $r(y)$ can be directly obtained. Note that the potential imbalance between the two classes is naturally taken into account by the prior distribution effect [9]. Finally, the method is computationally efficient since the EM algorithm converges rapidly to a local minimum of the log posterior distribution.

IV. EXPERIMENTS

The BPFD was initially designed to tackle the problem of Discriminative Pattern Mining for Predictive Maintenance on rail stock. Nevertheless, this approach is general and can be applied to any supervised classification problem. To demonstrate the validity and effectiveness of our approach and ensure full reproducibility, we evaluate the BPFD algorithm on various widely used and publicly available[1] data sets as well as on the industrial **Doors** data set. In addition, the method is compared across multiple classifiers against the Base Classifier

[1]https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

TABLE III
CHARACTERISTIC OF THE EXPERIMENTAL DATASETS.

Name	n	d	$\frac{\lvert\mathcal{D}_0\rvert}{\lvert\mathcal{D}_1\rvert}$
ijcnn1	91701	35	0.10
cod-rna	271617	17	0.5
a9a	32561	124	0.31
Doors	6349513	153	0.03

(BC) and the popular Polynomial Feature (PF) approach [15]. The results are reported in Table III.

A. Setup

The BPFD algorithm presented in section III-B and the Expectation-Minimization procedure are implemented using the Tensorflow 2.4 and Python 3.8. The experiments run on a Quad-core Intel i7 10th Gen @ 2.5 GHz. The source code and complementary experiments, including additional classifiers and data sets, are available online[2] for reproducibility.

B. Experiments

a) Data sets: The BPFD algorithm is tested on three public data sets commonly used for benchmark: **ijcnn1** consists of binarized maintenance data, **cod-rna** is a table of labeled strains of RNA and **a9a** is a record of census data to predict income of a household. The **Doors** data set has been provided by the French National Railway Company and consists of a database of log-events emitted by 143 trains' doors collected over twenty-four months. For each data set, the number of samples n, the size of the base dictionary $d = |E|$ and the class imbalance $\frac{\mathcal{D}_0}{\mathcal{D}_1}$ is reported in Table III.

b) Feature Discovery: We consider the $10\times$ cross-validated F_1, Area Under the Curve (AUC), Recall and Accuracy metrics to evaluate the improvement over the classifiers reported in the result Table II with $70\% - 30\%$ train-test split. In particular, the proposed approach improves the overall AUC score for almost all data sets and classifiers considered. For instance, the **ijcnn1** experiment exhibits an AUC of 0.927 for the Extreme Gradient Boosting (XGB) classifier whereas the vanilla approach scores at 0.769. On all data sets, the gain seems particularly significant for the Recall metrics. It seems that the discriminating pattern mined allows the classifier to be more sensitive. This is particularly important in the Predictive Maintenance domain where false negatives are generally the most costly type error that can be made.

c) Discriminative Patterns: BPFD is compared with state-of-the-art SPuManTe [11] test and retrieve most of the patterns with comparable significance level. These patterns revealed to be very informative about the link between a breakdown and pattern of code emission as well as explaining why a given algorithm would produce an incorrect prediction. As an example, in the case of **Doors** fault prediction, the Base Classifier would typically raise the probability of breakdowns after a manual blocking of a door by the onboard personnel represented by the event $e_m = \{"Locking\ \ Door"\}$. Our approach shows that some patterns that indicate whether this blocking is intended or not. For instance, the pattern $x = \{"Locking\ \ Door", "Unlocking\ \ Door"\}$ is not interpreted as an alert with BPFD as it is likely to be a handling error. More complex events have been extracted and their relevance validated with maintenance experts.

[2]https://github.com/amirdib/bpfd

V. CONCLUSION

In this work, we introduced a new algorithm for DPM and derived a Feature Discovery method to improve performance of any classifier in the supervised learning framework. This method is tested on various real-world and production data. In addition to the metric score improvement, our approach offers explainable insights on the classification task. Some extensions of this work could include using the bread-stick model to alleviate the need for a mixture parameter K. The present framework can easily be extended to multi-categorical classification. We plan to consider it in future work.

REFERENCES

[1] M. Bevilacqua and M. Braglia, "The analytic hierarchy process applied to maintenance strategy selection," *Reliability Engineering & System Safety*, vol. 70, no. 1, pp. 71–83, Oct. 2000.

[2] F. Ghofrani, Q. He, R. M. P. Goverde, and X. Liu, "Recent applications of big data analytics in railway transportation systems: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 226–246, May 2018.

[3] S. Koukoura, J. Carroll, S. Weiss, and A. McDonald, "Wind turbine gearbox vibration signal signature and fault development through time," in *2017 25th European Signal Processing Conference (EUSIPCO)*. Kos, Greece: IEEE, Aug. 2017, pp. 1380–1384.

[4] N. Kolokas, T. Vafeiadis, D. Ioannidis, and D. Tzovaras, "Forecasting faults of industrial equipment using machine learning classifiers," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*, Jul. 2018, pp. 1–6.

[5] S. Y. Kung, *Kernel Methods and Machine Learning*. Cambridge University Press, 2014.

[6] L. Basora, X. Olive, and T. Dubot, "Recent advances in anomaly detection methods applied to aviation," *Aerospace*, vol. 6, no. 11, p. 117, 2019.

[7] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," in *In: Proceedings of the 1993 Acm Sigmod International Conference on Management of Data, Washington Dc (Usa*, 1993, pp. 207–216.

[8] W. Hämäläinen and G. I. Webb, "A tutorial on statistically sound pattern discovery," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 325–377, 2019.

[9] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis, Third Edition*. CRC Press, Nov. 2013.

[10] P. Fournier Viger, C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z.-H. Deng, and H. Lam, "The SPMF Open-Source Data Mining Library Version 2," Sep. 2016.

[11] L. Pellegrina, M. Riondato, and F. Vandin, "SPuManTE: Significant pattern mining with unconditional testing," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1528–1538.

[12] S. Goodman, "A Dirty Dozen: Twelve P-Value Misconceptions," *Seminars in Hematology*, vol. 45, no. 3, pp. 135–140, Jul. 2008.

[13] K. Pearson, "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[15] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press, 2019.