BNL-211640-2019-COPA

Failure Analysis on Multivariate Time-series Data given Uncertain Labels

H. Huang, S. Yoo

Submitted to the 15th International Conference on Machine Learning and Data Mining MLDM'2019 Conference
to be held at Newark NJ, USA
July 20 - 25, 2019

Computational Science Initiative

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC), Advanced Scientific Computing Research (SC-21)

Notice: This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Failure Analysis on Multivariate Time-series Data given Uncertain Labels

Hao Huang[1], Shinjae Yoo[2], and Yunwen Xu[3]

[1] General Electric, San Ramon, CA, USA 94583 haohuanghw@gmail.com
[2] Brookhaven National Lab, Upton, CA, USA 11973 shinjae@gmail.com
[3] Amazon, Palo Alto, CA, USA 94303 yunwenxu04@gmail.com

Abstract. Machine failure analysis and detection is critical to today's industrial society. Given a number of failure events on multivariate temporal data, the ability to identify 1) the *discriminative patterns* prior to the events and 2) the most *relevant features* to the failure has practical use for early warning and root cause analysis. However, since these patterns are not necessarily adjacent to the onset of failure in time, faulty labels are often with uncertainty, which makes traditional supervised detection methods inapplicable. To address the *label uncertainty* and learn the complicated correlation in multivariate time series, we design Failure Analysis on Multivariate Time-series Data (MAMT) that jointly selects the most failure-relevant features and time-instances by a novel *dynamic and directional label diffusion* process. Extensive experiments demonstrate that MAMT is more effective while more efficient than popular baselines.

Keywords: Failure Analysis · Multivariate Time-series · Uncertain Labels.

1 Introduction

With the rapid advancement in industrial machinery design, data-driven analytics has gained tremendous success in various industrial aspects. Among these applications, failure analysis is a long-standing crucial problem that can significantly benefit from data mining. One usually assumes that there are discriminative patterns that appeared prior to the failure events on certain features and timestamps. Knowing these patterns not only helps on finding root causes of the failure, but also makes early detection possible in the future. However, extracting such patterns from large-volume multivariate sensor data collected from a number of independently operated data sources (e.g., machines) is a big challenge. The goal here is to find these patterns and design better classifiers upon such patterns.

There are two major challenges in finding such patterns besides the large-scale data processing requirement: 1) the dataset is multivariate and usually collected from multiple data sources, both fault-related or not; 2) while the timestamp and presence of the failure event is known, the contributing features are unknown (feature uncertainty), and the time fault patterns occurred is uncertain and not necessarily adjacent to the event (label uncertainty). So reliable labels on time instances required for supervised machine learning are not available.

To address these challenges, we design a method called Multi-view Failure Analysis on Multivariate Time-series Data (MAMT), which overcomes the limitations of traditional supervised methods on such problems. The concept of the multi-view failure

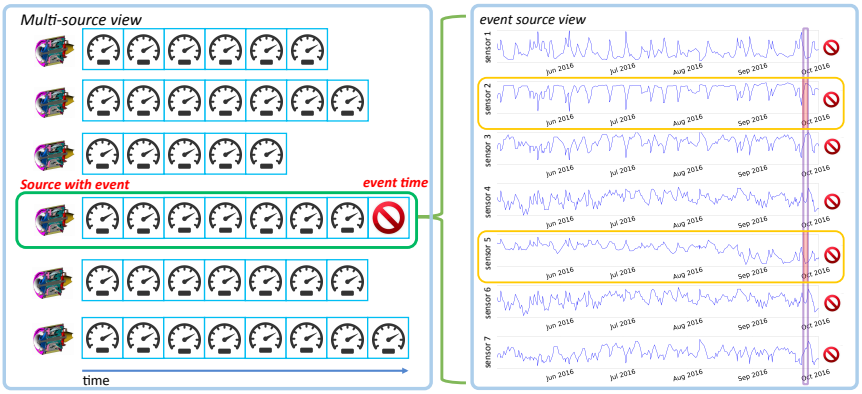


Fig. 1. Illustration of the multi-view failure analysis on a dataset collected from six machines (data sources). Given the event machine (rounded by green box) and its event time, MAMT can bi-select the key features (rounded by yellow box) and the most relevant time-instances (rounded by purple box).

analysis is illustrated in Figure 1. The left panel displays the data viewed by different sources (machines), where the failure source and event time is known. The right panel shows the multivariate temporal plots of the failure source. The fault pattern on the feature view (failure-related features, rounded by the yellow box) and timestamp view (failure-related time-instances, true "1"s, rounded by the purple box) are unknown. Without other prior knowledge, MAMT can select on both views (marked by the red shadow). Our contributions can be summarized as follows:

(1) We design a method that jointly selects feature and time-instance that relevant to failure from multivariate multi-source temporal data.
(2) To address label uncertainty, we propose dynamic and directional label diffusion (DDLD) which corrects timestamps that are falsely labeled as "failure-relevant", and thus locates the true 'failure-relevant" timestamps.
(3) The selected features and instances can be used to build more accurate failure classifiers/detectors.
(4) Compared to popular methods, MAMT is more effective with a lower time/space computational complexity.

2 Preliminaries

2.1 Background

A **failure mode** is a type of failure. An **event** (e.g. the final shutdown of the green rounded machine in Figure 1) is a real failure case happens in one data source at certain timestamps. We assume that there exist discriminative patterns (e.g. the red shaded pattern in Figure 1) prior to failure events that can be used to detect failure. As usually one starts by labeling all time-instances in a time-window (**exploration windows**) prior to the failure event as positive, but often only a small fraction of the instances within certain feature space in such window are relevant to failure. The analytical goal is to bi-select both failure-relevent features and time-instances, which can be used for failure

detection and diagnosis. With certain domain knowledge, we assume the pattern appears no earlier than η timestamps before the events; or we make a formal **assumption** here: the exploration windows contain no more than η time-instances for each event. In practice, η is set conservatively. We will further discuss in Section 6.

2.2 Notations

For a matrix $Z \in \mathbb{R}^{m\times n}$, Z^T denotes its transpose and Z^{-1} denotes its (pseudo)inverse, while $Z(i,j)$ denotes the element in the i-th row and j-th column, and $Z(i,:)$ and $Z(:,j)$ denote its i-th row and j-th column respectively. We use entry-wise norms denoted by $\|Z\|_p$, where $p=2$ gives Frobenius norm $\|Z\|_F = \sqrt{\sum_{ij} Z(i,j)^2} = \sqrt{\mathrm{tr}(ZZ^T)}$, and $p=(2,1)$ gives $\ell_{2,1}$ norm $\|Z\|_{2,1} = \sum_{i=1}^m \|Z(i,:)\|_2$. For a vector $[w_1,...,w_m] \in \mathbb{R}^{m\times 1}$, $\mathrm{diag}([w_1,...,w_m]) \in \mathbb{R}^{m\times m}$ denotes a diagonal matrix with $w_1,...,w_m$ as its diagonal entries, while $\mathrm{diag}(Z)$ for a given square matrix Z is a square matrix that has the same diagonal entries as Z but off-diagonal entries are all zeros. $\mathbb{I}_m$ is an identity matrix of dimension $m \times m$.

3 Problem Formulation

3.1 Problem Setting

Consider a failure analytic task with k available events of the same failure mode. Let any instance have m-dimensional features after feature engineering (see Section 6.1 for details). Without loss of generality, we assume η is less than the smallest time gap between two consecutive events if there are more than one events from one source (that is, the exploration windows from the same source are disjoint). Therefore, any instance can belong to at most one exploration window. Also as time-dependency is captured by the derived features, instances from different timestamps or data sources become independent. Thus, the data matrix can be organized as $X = [x_1, x_2, ..., x_n] = [X_a, X_b] \in \mathbb{R}^{m\times n}$, where n is the number of all instances, $a := k\eta$, $b := n - k\eta$, $X_a = [x_1, x_2, ..., x_a] \in \mathbb{R}^{m\times a}$ are instances from all the exploration windows, and $X_b = [x_{a+1}, x_{a+2}, ..., x_{a+b}] \in \mathbb{R}^{m\times b}$ are all the normal instances that irrelevant to the failure (usually $b \gg a$).

The initial exploration windows define the *tentative event label matrix*, given by $E \in \{0,1\}^{k\times n}$, where $E(j,i) = 1$ if x_i is an instance from the exploration window of the j-th event, otherwise $E(j,i) = 0$. Similarly, we define the *time-instance weighting matrix* as $Y \in \mathbb{R}^{k\times n}$, where the continuous value $Y(j,i)$ represents the relevance of instance x_i to the j-th event. Obviously, the associated label matrices can be written as $E = [E_a, E_b]$ and $Y = [Y_a, Y_b]$, where $E_a \in \{0,1\}^{k\times a}$ and $Y_a \in \mathbb{R}^{k\times a}$ correspond to the instances from X_a, and $E_b = Y_b = \mathbf{0}^{k\times b}$ correspond to normal instances from X_b. Note that, E_a has exact one "1" in each column.

We then introduce the key components of our algorithm to address the feature and label uncertainties.

3.2 Supervised Feature Selection

Suppose reliable labels Y is given, identifying event-relevant features can be addressed by the following supervised feature selection method [9].

$$\min_W \|W^T X - Y\|_F^2 + \alpha \|W\|_{2,1}, \tag{1}$$

where $X \in \mathbb{R}^{m\times n}$ is the input data matrix with n instances and m features, and $Y \in \mathbb{R}^{k\times n}$ is the (binary) class label indicator matrix with $Y(j,i) = 1$ if x_i is labeled as class j, otherwise $Y(j,i) = 0$. The solution $W \in \mathbb{R}^{m\times k}$ gives a *feature weighting matrix*, where $W(i,j)$ represents the importance of the i-th feature to the j-th class.

The first term in Eq (1) is a smooth convex loss function, the second term represents the sparsity of W with α controlling the trade-off. This method has been popularly used in industrial failure analysis domain [8,21] due to its better explainability and outperformance, so we use it as our feature selection baseline.

3.3 Label Diffusion

However, due to uncertainty in the tentative labels, the aforementioned supervised methods will fail. Most existing label noise-reduction methods [4,15] are not suitable either, since they all assume only a small portion of labels are wrong.

Here we leverage the *label propagation* idea from semi-supervised learning, which tries to learn the unknown labels by propagating information from the labeled data. Classical Label Propagation (CLP) [11] defines a label diffusion process on a graph, whose nodes are instances and edges represent the similarity between nodes [11]. The diffusion starts from labeled instances, then propagates to the entire graph via random walks. Its stationary state can be solved analytically by

$$\min_{Y} \sum_{i,j=1}^{n} \tilde{A}(i,j)\|Y(:,i) - Y(:,j)\|_F^2 + \mu \sum_{i=1}^{n} \|Y(:,i) - E(:,i)\|^2, \tag{2}$$

where $E \in \{0,1\}^{k\times n}$ is a binary label initialization, $Y \in \mathbb{R}^{k\times n}$ represents the final (soft) label assignments, and $\tilde{A} \in \mathbb{R}^{n\times n}$ is a normalized transition matrix with $\tilde{A}(i,j)$ defining the transition probability from x_i to x_j. Eq (2) seeks the trade-off between the *global smoothness* meaning that two similar instances should share the same labels (the first term), and the *local fitness* meaning that the final label should not be dramatically different from the initialization (the second term), with μ controlling the balance.

CLP defines a symmetric diffusion process. However, a unique challenge in failure analysis is the asymmetric uncertainty assumption, that is the uncertainty only comes from E_a. To exploit this property, we propose a directional label diffusion (DLD) process, which allows to correct the initial *positive* labels through information propagation. The details of DLD will be discussed in Section 4.1.

3.4 Formal Definition

With the baseline solution suite, the goal of our Multi-view Failure Analysis on Multivariate Time-series Data (MAMT) can be stated as to identify the failure-relevant features and time-instances from massive multi-sourced data. Or more formally, to solve a feature score matrix $W \in \mathbb{R}^{m\times k}$, and an instance score matrix $Y \in \mathbb{R}^{k\times n}$ (or equivalently $Y_a \in \mathbb{R}^{k\times a}$) that minimize the following loss function,

$$\mathcal{M}(W,\hat{Y}) = \underbrace{\|W^T XB - \hat{Y}B\|_F^2 + \alpha\|W\|_{2,1}}_{(I)}$$

$$+ \delta \underbrace{\left[\sum_{i,j=1}^{n} \tilde{A}(i,j) \|\hat{Y}(:,i) - \hat{Y}(:,j)\|_F^2 + \mu \sum_{i=1}^{n} \|\hat{Y}(:,i) - E(:,i)\|^2 \right]}_{(II)}$$

$$+ \gamma \underbrace{\|\hat{Y}^T\|_{2,1}}_{(III)} \tag{3}$$

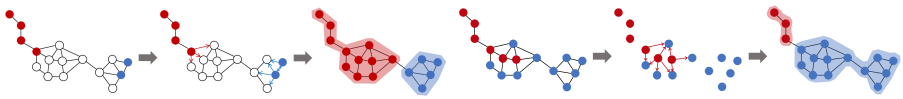
where $\hat{Y} = Y \circ E$ is the Hadamard product of Y and E, and $B = \text{diag}([\beta \mathbf{1}_a, \mathbf{1}_b])$ is the instance weights in a diagonal format with β being the relative weight for all instances in X_a. Eq (3) simultaneously addresses feature selection (Term I) and label correction (Term II). Meanwhile, it also seeks sparsity in selected instances (given by the $\ell_{2,1}$ norm regularization on $\hat{Y}^T$ in Term III). The resulting score matrices $W(i,j)$ and $\hat{Y}(j,i)$ represent the contribution of the i-th feature and instance to the j-th event.

Remark 1: The reason to bring $\hat{Y}$ is to "suppress" Y_b; and the reason to bring B is to introduce cost-sensitive objective to handle data imbalance issue between X_a and X_b.

Remark 2: Although each event may have its respective feature weights, features weights from different events are not totally independent, as they are from the same failure mode. Therefore, we want to find *consistent* event-related features across all events. Eq (3) adopts two strategies to enforce the global consistency: the parameter α controls the global row sparsity of W, and the normalized transition matrix $\tilde{A}$ controls global diffusion process.

Remark 3: In a failure analysis task, interpretability is usually of high priority. The objective is for any given number p and q, provides the top-p features that characterize the most separable pattern between X_b and the q most representative event instances in X_a.

4 Problem Solution



(a) Classical Label Propagation (CLP) (b) Directional Label Diffusion (DLD)

Fig. 2. The difference between our proposed DLD and standard CLP. In the left most plots, red and blue points are instances initially labeled as "1" and "0". In CLP, both red and blue propagate out along the edges. But in DLD, red label can either stay with itself, or diffuse out, while transitions from blue to red or between red nodes are not allowed. As a result, DLD successfully corrects the mislabeled "1"s in (b).

4.1 Directional Label Diffusion

In the initial setting, only the positive labels are assumed to be uncertain. **Our goal is to correct the wrong positive labels**. We propose an asymmetric *Directional Label Diffusion (DLD)* based on a directional random walk (Figure 2(b)), in contrast to the omnidirectional random walk used in the CLP (Figure 2(a)).

Given the whole dataset $X = [X_a, X_b] \in \mathbb{R}^{m\times n}$ with m features and n instances, of which $Y_a \in \{0,1\}^{k\times a}$ is initially labeled as "1" for one event, and $Y_b \in 0^{k\times b}$ are all "0" to start with. We construct an *instance transition matrix* $A \in \mathbb{R}^{n\times n}$ and split it into four blocks

$$A = \begin{bmatrix} A_{aa} & A_{ab} \\ A_{ba} & A_{bb} \end{bmatrix}, \tag{4}$$

where A_{aa} describes the transition within X_a, A_{ab} describes the transition from X_a to X_b, and so forth.

We assume the instances in X_a that carry wrong positive labels are close to (partial of) X_b on *all* feature subspace, and only those labeled correctly are separable from X_b in *certain* feature subspace. Therefore in the diffusion process, we hope that the wrong "1"s are smoothed out through their "0" neighborhood and the right "1"s can be maintained. Specifically, we allow the "1" labels to diffuse from X_a to X_b only. Also we ignore the diffusion within X_b since we assume all "0"s are labeled confidently. Furthermore, within X_a, we only allow self-loops so that right "1"s can be conserved during diffusion process. In other words, diffusion from each instance in X_a to X_b are independent, because transition between two instances in X_a are not defined. As a result, A in our DLD is defined as

$$A = \begin{bmatrix} \mathrm{diag}(A_{aa}) & A_{ab} \\ \mathbf{0}_{ba} & \mathbf{0}_{bb} \end{bmatrix}, \tag{5}$$

where $\mathrm{diag}(A_{aa})$ only keeps the diagonal of A_{aa}, and the original A_{ba} and A_{bb} are set to be $\mathbf{0}$ matrix. Figure 2(b) illustrates our proposed DLD. By using Eq (5), entries of $\hat{Y}_a$ will be decreased and the wrong "1"s will be decreased much faster as we dissipate the confidence. As a result, the wrong positive labels would be corrected and only the right ones would be conserved. Such instance transition matrix A by DLD will be used in Eq (3).

4.2 Proposed Algorithm

We now present an iterative solution to solve Eq (3). By rewriting the Term II of Eq (3) in matrix form, $\delta tr(\hat{Y}(\mathbb{I}_n - \tilde{A})\hat{Y}^T) + \mu\delta tr(\hat{Y} - E)(\hat{Y} - E)^T$, it is easy to verify that the loss function $\mathcal{M}$ is jointly convex in W and $\hat{Y}$, so a global optimal solution exists. It is difficult to optimize W and $\hat{Y}$ simultaneously. Therefore we adopt an alternating and iterative optimization procedure, which works well for a number of practical optimization problems[15].

Given $\hat{\mathbf{Y}}$, optimize $\mathbf{W}$. Given $\hat{Y}$, the optimal W can be computed by minimizing the following objective function

$$\mathcal{M}(W) = \|W^\top XB - \hat{Y}B\|_F^2 + \alpha\|W\|_{2,1} + \text{const wrt W}. \tag{6}$$

The update rule for W at each iteration is given by

Theorem 1. *When fixing $\hat{Y}$, W can be obtained by*

$$W = (XB^2X^\top + \alpha D_W)^{-1}(XB^2\hat{Y}^\top), \tag{7}$$

where D_W is a diagonal matrix with the i-th diagonal element as $D_W(i,i) = \frac{1}{2\|W(i,:)\|_2}$.

Proof. The cost function (without constant term) in Eq (6) can be written as

$$\mathcal{L}(W) = tr(W^\top XB^2X^\top W - 2W^\top XB^2\hat{Y}^\top) + \alpha\|W\|_{2,1}.$$

This is quadratic (thus convex) in W, so taking the partial derivative of $\mathcal{M}$

$$\partial\mathcal{L}(W)_W = 2XB^2X^\top W - 2XB^2\hat{Y}^\top + 2\alpha D_W W.$$

and setting it to be 0 yields the update rule as in Eq (7).

Given W, optimize $\hat{\mathbf{Y}}$. Given W, the optimal $\hat{Y}$ can be computed by minimizing the following objective function

$$\begin{aligned}\mathcal{M}(\hat{Y}) &= \min_{\hat{Y}} \|W^\top XB - \hat{Y}B\|_F^2 + \gamma\|\hat{Y}^\top\|_{2,1} \\ &+ \delta\Big[\sum_{i,j=1}^{n} \tilde{A}(i,j)\|\hat{Y}(:,i) - \hat{Y}(:,j)\|_F^2 + \mu\sum_{i=1}^{n}\|\hat{Y}(:,i) - E(:,i)\|^2\Big]. \end{aligned} \tag{8}$$

Below is the update rule for $\hat{Y}$ in each iteration.

Theorem 2. *When fixing W, $\hat{Y}$ can be obtained by*

$$\hat{Y} = (W^\top XB^2 + \delta\mu E)(B^2 + \gamma D_{\hat{Y}^\top} + \delta\tilde{L} + \delta\mu\mathbb{I}_n)^{-1}, \tag{9}$$

where $D_{\hat{Y}^\top}$ is a diagonal matrix with the i-th diagonal element as $D_{\hat{Y}^\top}(i,i) = \frac{1}{2\|\hat{Y}(:,i)\|_2}$, and $\tilde{L} = \mathbb{I}_n - \tilde{A}$ is a Laplacian matrix of $\tilde{A}$ from Section 4.1.

Proof. We rewrite the cost function in Eq (8) as

$$\begin{aligned}\mathcal{L}(\hat{Y}) &= tr(\hat{Y}B^2\hat{Y}^\top - 2W^\top XB^2\hat{Y}^\top) + \gamma\|\hat{Y}^\top\|_{2,1} \\ &+ \delta tr\Big[\hat{Y}\tilde{L}\hat{Y}^\top + \mu(\hat{Y} - E)(\hat{Y} - E)^\top\Big]. \end{aligned} \tag{10}$$

The partial derivative of $\mathcal{L}(\hat{Y})$ for $\hat{Y}$ is therefore

$$\begin{aligned}\partial\mathcal{L}(\hat{Y})_{\hat{Y}} &= 2\hat{Y}B^2 - 2W^\top XB^2 + 2\gamma\hat{Y}D_{\hat{Y}^\top} \\ &+ 2\delta\hat{Y}\tilde{L} + 2\delta\mu(\hat{Y} - E). \end{aligned} \tag{11}$$

Since Eq (10) is convex, setting the above partial derivative to zero gives us Eq (9).

The directional label diffusion is essential for correcting the initial labels and it relies on the accuracy of the transition matrix A. Here we introduce a dynamic process to build a more adaptive A by considering the most current feature weighting. Within each iteration, once the W got updated we use it to attach importance to each feature. In other words, A is being updated during the optimization process. The A matrix at the t-th iteration, A_t, is built upon a weighted dataset $X^\top W_t$ where W_t is the W matrix at the t-th iteration. Therefore our proposed **Dynamic and Directional Label Diffusion (DDLD)** is controlled by the following (time-varying) transition matrix

$$A_t = \begin{bmatrix} \mathrm{diag}(X_a^\top W_t {W_t}^\top X_a) & X_a^\top W_t {W_t}^\top X_b \\ \mathbf{0}_{ba} & \mathbf{0}_{bb} \end{bmatrix}, \tag{12}$$

where we utilize the *traditional linear kernel* by default. For the non-linear problem, non-linear kernels such as Gaussian kernels could be used in a similar way.

4.3 Efficient Implementation

The running complexity is determined by Eq (7), (9), and (12). When the number of instances n is large, each iteration consumes a lot of time and memory. In this section we introduce an efficient implementation by taking advantage of the sparsity in $\hat{Y}$ and A.

Proposition 1. *Eq* (7) *can be rewritten as*

$$W = (X_b X_b^\top + \beta^2 X_a X_a^\top + \alpha D_W)^{-1}(\beta^2 X_a Y_a^\top). \tag{13}$$

The label update rule in Eq (9) only depends on the part corresponding to X_a. Since the other part in $\hat{Y}$ corresponding to X_b is always suppressed by E in $\hat{Y} = Y \circ E$. We then have the following:

Proposition 2. Y_a *can be extracted from Eq* (9) *by*

$$Y_a = (W^\top X_a \bar{B} + \delta\mu E)(\bar{B} + \gamma D_{\hat{Y_a}^\top} \bar{\mathbb{I}}_{an} + \delta \bar{L} + \delta\mu \bar{\mathbb{I}}_{an})^{-1}, \tag{14}$$

where $\bar{B} = [\mathrm{diag}(\beta^2 \boldsymbol{1}_a), \boldsymbol{0}_{ab}] \in \mathbb{R}^{a\times n}$, $\bar{\mathbb{I}}_{an} = [\mathbb{I}_a, \boldsymbol{0}_{ab}] \in \mathbb{R}^{a\times n}$, *and* $\bar{L} \in \mathbb{R}^{a\times n}$ *is the upper part of* $\tilde{L}$.

Both Proposition 1 and 2 are easy to prove. To save space we omit the details. Now we summarize our final Algorithm 1. Given $b \gg max(a, m)$, the time complexity of each iteration is $O(amb + a^2 b + m^2 b)$ and space complexity is $O(a^2 + ab)$.

Algorithm 1: MAMT

Input: X, E, α, β, γ, δ, μ,
$X = [X_a, X_b] \in \mathbb{R}^{m\times n}$ where n is #instances and m is #features, while $X_a \in \mathbb{R}^{m\times a}$ corresponds to the instances initially labeled as "1" and $X_b \in \mathbb{R}^{m\times b}$ are those initially labeled as "0" (usually $b \gg a$). And $E \in \mathbb{R}^{k\times n}$ to be the initial event labels, where $E(j, i) = 1$ if x_i is initialized as one of the event instances for the j-th event, otherwise $E(j, i) = 0$.
Output: Feature weight $W \in \mathbb{R}^{m\times k}$ and instance weight $Y_a \in \mathbb{R}^{k\times a}$.

```
Initialize Y = E and Y_a to be the first a columns of Y;
Initialize D_W = I_m;
while Not convergent do
    Calculate W using Eq (13) ;
    Calculate L̃ = I_n − Ã where Ã = D^{-1} A and A is from Eq (12) ;
    Calculate Y_a using Eq (14) ;
    Update Y by Y = [Y_a, 0_b] ;
end while ;
```

5 Related Work and Discussion

Our problem is unique in the following aspects: 1) it simultaneously conducts instance selection (to remove label uncertainty) and feature selection while begins with very

rough label initialization; 2) data may come from multiple sources which involve healthy and event ones. Here we introduce the most related work and discuss accordingly.

Our problem definition can be viewed as an extension of multiple instance learning to time series with feature selection. Many works have been done on computer aided diagnosis or image analysis [12,7] with global consistency as constraint. Comparatively, our setting is flexible as we allow each event case to have its own feature and instance weighting.

Unsupervised feature selection has been widely studied [19,2]. But they don't consider instance weighting. Some researches proposed feature and instance selection in the same framework [20] based on global normal distribution only. When applied to our problem setting, they could not differentiate the anomalous part from normal.

(Semi-)supervised learning can learn from the label initialization [9,18]. Classic label propagation (CLP) [17,11] shed insights into the design of our algorithm, but none of them focuses on both feature/instance selection. Furthermore, CLP is semi-supervised, while our problem is more close to supervised learning. Another key difference is that, CLP assumes that the initial labels are truly reliable, while in our problem setting the positive label may contain wrong labels and our goal to correct them and learn a more accurate model.

Researches have been focusing on time series event analysis [16,1,3], many of which are built upon auto-regression (e.g. ARIMA, VAR). Cheng et.al. proposed a ranking causal anomaly detection based on vanishing correlations on sensor graph [3]. It assumes that there exists an invariant network on sensors/features, which is not true in many real world problems where the normal data usually contains different patterns and the underlying distribution changes over time. And it is only for a single data source analysis. Comparatively, our MAMT allows dynamic patterns in normal data and it is capable to build one single model for multiple data sources. The method by Batal et al. [1] converts time series into intervals of temporal abstractions and detects discriminative patterns according to predefined abstract states. But it is sensitive to the setting of abstract states and cannot handle high dimensional data. On the contrary, MAMT directly learns the weight of features and is well applicable to a large number of features.

6 Experiment

6.1 Experiment Setup

Baseline methods

We consider five categories of baselines as follows.

MI-RVM [12]: a multiple instance learning algorithm with feature selection, which is completely automatic and does not require tuning any parameters. *Coselect [15]:* uses E as initial labeling seed but iteratively filters out the noisy labels with high residuals between predicted and observed labels. *SVM-FS:* trains support vector machine (SVM) on all X (X_a and X_b) with initial label E, then relabels the top instances of X_a as 1 (instance selection) according to their predicting scores, and finally performs the supervised $\ell_{2,1}$ norm-based feature selection given by Eq (1) with X_b labeled as 0. *One-class SVM (or 1SVM) [14] and Gaussian Mixture Model (GMM)-FS [10]:* train on true *normal data* X_b only, then label the *most anomalous* instances of X_a as 1 (instance selection), and finally perform supervised $\ell_{2,1}$ feature selection with X_b labeled as 0 [4].

[4] Although some anomaly detection methods like 1SVM also output feature weight, the weights are purely based on the normal set distribution (i.e., X_b here). As our goal is to use the

Table 1. Baseline Methods

	Description	Methods
#1	Multiple instance learning	MI-RVM [12]
#2	Iteratively update feature weights and instance weights	Coselect [15]
#3	Perform (supervised) classification first and feature selection later	SVM-FS
#4	Perform (unsupervised) anomaly detection first and feature selection later	1SVM-FS [14], GMM-FS [10]
#5	Perform feature selection first and anomaly detection later	FS-GMM

FS-GMM: first performs supervised $\ell_{2,1}$ feature selection with the initial label E, and then performs GMM anomaly detection (trained on X_b and tested on X_a) on the top features subspace to choose the most anomalous instances (instance selection).

We do not include Classical Label Propagation (CLP) as baseline here because it is semi-supervised method and therefore not applicable in our problem setting.

In model training, we use linear kernel for all the SVM (i.e. libsvm) due to its popularity and interpretability in industry [6,13]. Specifically, i) all the parameters in the baselines are tuned to get the best performance; ii) the number of components for GMM is set to be six; iii) the linking graph R in Coselect is constructed with $R(i, j) = 1$ if x_i and x_j belong to the same event in X_a, zero otherwise. As for our MAMT, we choose the default regularization parameters in Algorithm 1 as: $\alpha = n/m$, $\beta = \sqrt{b/a}$, $\gamma = 1$, $\delta = 1$ and $\mu = 0.3$. The selection is motivated by balancing the sample sizes from normal and faulty classes; however, our sensitivity study in Section 6.4 indicates the algorithm performance is reasonably stable to such parameters tuning.

The selection of a (exploration window size) usually requires field knowledge. Generally it should not be either too small (not to exclude the failure-relevant instances) or too large (not to include other failure-irrelevant anomalies).

Table 2. Dataset summary: number of data sources, events(k), features(m), initial event instances(a) and normal instances(b).

data name	# sources	# k	# m	# a	# b
Data A	1599	12	77 (s=11)	3452	2893581
Data B	135	22	90 (s=12)	31680	2321200

Real world dataset

We use two industrial datasets to evaluate the algorithm performance (dataset summarized in Table 2). In the first dataset, sensor data from a fleet of 1599 engines are available. This fleet have 2 years running life on average, and 12 engines have demonstrated a thrust deficit problem after certain usage. The sensor data were recorded two

reweighted (or selected) features to distinguish the abnormal patterns in X_a, the supervised feature selection is more suitable.

times a day, and contain 11 raw measurements, including engine fan speed, compressor pressure and temperature, exhaust gas temperature, oil temperature, fuel flow and vibration parameters and so on. The second dataset was collected from electrical pump system that includes a fleet of 135 pump machine, among which 22 have pump failure after certain usage. The sensor data were recorded at a frequency of every 30 minutes. There are 12 measurements including electric current, intake pressure and temperature, output frequency, vibration and voltage parameters. Domain experts believe that the exploration window should not exceed 3 months for the first dataset and 1 month for the second dataset, and the value of a was set accordingly.

Time-series feature construction
We embed the temporal dependency into feature space with temporal feature engineering. Different methods (e.g. EMD, FFT) have been explored [5]. Here we use sliding-window-based constructions. Given the window size ρ, each operation g is an algorithm that takes a time series window $(x'_1, x'_2, ..., x'_\rho)$ as input, and outputs a single number. The x'_i is usually univariate or bivariate. Here we apply auto-correlation (of univariate raw sensor) and cross-correlation (of pairwise raw sensors). Together with the raw sensor record, the dataset dimension becomes $m = 2s + s(s-1)/2$ given s as the number of raw sensors. The setting of ρ should involve domain knowledge as well. In our experiments we set $\rho = 20$, and each dimension of X is then scaled into $[0, 1]$.

Evaluation method
The biggest challenge in evaluating failure analysis is the limited ground truth. There are two ways to verify the selected features/instances: one is by domain expert examination, and the other is by checking the classification capability upon selected features/instances. We show the latter comparison in Figure 3, but also discuss the former in Section 6.3. In detail, our evaluation method is designed to test the event classifier developed with the selected top q-instances and p-features. The value of p and q are selected in advance, and usually set for the interpretability requirement and investigation budget. In our problem, we also got the insights from industrial domain experts. Nevertheless, the numerical experiments also indicates that the MAMT is reasonably stable for p and q tuning (as shown in Figure 3).

Having the classifiers from each methods, we evaluate their performance by the leave-one-out cross-validation (LOOCV). Every time we randomly train on $k-1$ events to generate the top instances and features; then a linear SVM classifier is constructed. The classifier is tested on the left-out event, and detection quality is measured by precision and recall. In particular, a detection is a *true positive* if a predicted value "1" is obtained within three months prior to the final failure (effective event detection), otherwise it is false positive. The recall value equals to one if a model successfully predicts the event with a lead-time no longer than predefined window (longer lead-time is considered as irrelevant or false positive), and otherwise equals to zero.

6.2 Experiment Result Analysis

Besides the baselines in Table 1, we also included a linear SVM without feature/instance selection to show the necessity of feature/instance selection. Figure 3 shows the performance with different number of features (when fixing $q = 5$) and different number of instances (when fixing $p = 2$). There are a few importance observations:

(1) Classifiers upon bi-selection clearly outperform simple SVM without any selection.

(2) MAMT gives the best classifier with consistently the highest precision and recall.
(3) MAMT has relatively low false positive rate. For Data A, the false positive by MAMT include only four data sources on average, which is much lower than the false positive by the other that involve more than eight data sources. For Data B, the false positive by MAMT include only three data sources on average, but the second best baseline has more than nine on average.
(4) The precision/recall usually drop in the end with large number of features/instances. It is because more *irrelevant* features/instances introduce more noise. This is different from the standard binary classification, for which the performance usually increases (both precision and recall) as more *relevant* instances are used for training, if the labels are reliable.
(5) In Figure 3, precision/recall by MAMT usually reach its top performance with intermediate number of features/instances. This suggests that our MAMT perform better due to the combination of both better feature selection and better instance labelling.

6.3 Identified Features by MAMT

For Dataset A, the top features by MAMT are edge deteriorations and exhaust gas temperature. The domain experts confirmed that they were physically responsible to the engine thrust deficit problems. While the top features from Dataset B by MAMT are relatively high or low intake temperature and motor temperature, which are also confirmed as physically driven features of pump failure by domain experts. On the other hand, the top features by the other baselines are either physically irrelevant or incomplete compared against our MAMT selection.

6.4 Parameter Sensitivity and Scalability Analysis

Although MAMT requires some user-defined parameter (α, β, γ, δ), we observe that the MAMT performs stably in a wide range of values in Figure 4, which illustrates the precision/recall value by respectively fixing two of the four parameters (α, β, γ, δ) and using grid search on the rest two. Each parameter is timed by a base-10 power scale.

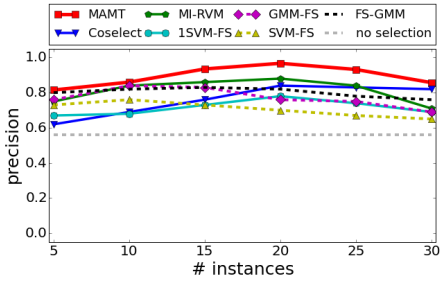
Figure 5 shows the scalability comparison with a simulation dataset, in which we change the size of X_b, the key affecting factor for complexity, while holding the rest parameters fixed. The linear SVM (libsvm) training time is non-linear, and Coselect requires to build a full instance graph, therefore they are not scalable. With the fast implementation in Section 4.3, MAMT has the second best scalability. FS-GMM is the fastest since its GMM is only trained on the selected feature subspace, making it more efficient than GMM-FS which trains on the full feature space.

7 Conclusion

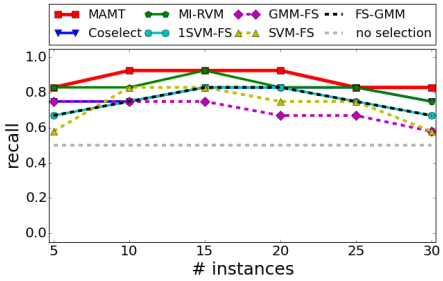
This paper presents a failure-relevant feature/time-instance selection method for machine failure detection and analysis on multivariate temporal data. Different from the standard supervised methods, MAMT identifies the most relevant features and time-instances which can be used for better classification. Furthermore, MAMT can involve multiple data sources, which is difficult to any uni-source analytics. We show that MAMT is effective and efficient, verified by theoretical analysis and industrial experiments.

References

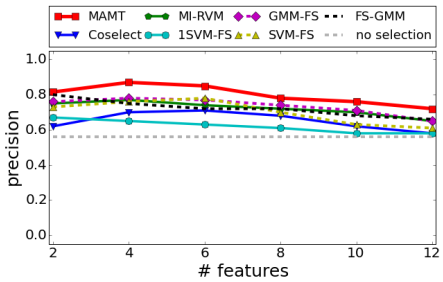
1. Batal, I., Fradkin, D., Harrison, J., Moerchen, F., Hauskrecht, M.: Mining recent temporal patterns for event detection in multivariate time series data. In: ACM SIGKDD. pp. 280–288. ACM (2012)
2. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: Proceedings of ACM SIGKDD (2010)
3. Cheng, W., Zhang, K., Chen, H., Jiang, G., Chen, Z., Wang, W.: Ranking causal anomalies via temporal and dynamical analysis on vanishing correlations. In: ACM SIGKDD (2016)
4. Frénay, B., Doquire, G., Verleysen, M.: Estimating mutual information for feature selection in the presence of label noise. Computational Statistics & Data Analysis **71**, 832–848 (2014)
5. Fulcher, B.D., Jones, N.S.: Highly comparative feature-based time-series classification. IEEE TKDE (2014)
6. Guo, D., Raghavendra, C.S., Yao, K.T., Harding, M., Anvar, A., Patel, A., et al.: Data driven approach to failure prediction for electrical submersible pump systems. In: SPE Western Regional Meeting. Society of Petroleum Engineers (2015)
7. Jhuo, I.H., Lee, D., et al.: Multiple-instance learning: Multiple feature selection on instance representation. In: AAAI (2011)
8. Li, D., Zhou, Y., Hu, G., Spanos, C.J.: Optimal sensor configuration and feature selection for ahu fault detection and diagnosis. IEEE Transactions on Industrial Informatics **13**(3), 1369–1380 (2017)
9. Liu, J., Ji, S., Ye, J.: Multi-task feature learning via efficient l 2, 1-norm minimization. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press (2009)
10. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: CVPR. IEEE (2010)
11. Nie, F., Xiang, S., Liu, Y., Zhang, C.: A general graph-based semi-supervised learning with novel class discovery. Neural Computing and Applications **19**(4), 549–555 (2010)
12. Raykar, V.C., Krishnapuram, B., Bi, J., Dundar, M., Rao, R.B.: Bayesian multiple instance learning: automatic feature selection and inductive transfer. In: Proceedings of the 25th international conference on Machine learning. pp. 808–815. ACM (2008)
13. Santos, P., Villa, L.F., Reñones, A., Bustillo, A., Maudes, J.: An svm-based solution for fault detection in wind turbines. Sensors **15**(3), 5627–5648 (2015)
14. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural computation **13**(7), 1443–1471 (2001)
15. Tang, J., Liu, H.: Coselect: Feature selection with instance selection for social media data. In: Proceedings of SIAM SDM (2013)
16. Verron, S., Tiplica, T., Kobi, A.: Fault detection and identification with a new feature selection based on mutual information. Journal of Process Control **18**(5), 479–490 (2008)
17. Wang, B., Tsotsos, J.: Dynamic label propagation for semi-supervised multi-class multi-label classification. Pattern Recognition (2016)
18. Yoon, H., Shahabi, C.: Feature subset selection on multivariate time series with extremely large spatial features. In: ICDM Workshops (2006)
19. Yoon, H., Yang, K., Shahabi, C.: Feature subset selection and feature ranking for multivariate time series. IEEE transactions on knowledge and data engineering **17**(9), 1186–1198 (2005)
20. Zhang, L., Chen, C., Bu, J., He, X.: A unified feature and instance selection framework using optimum experimental design. IEEE Transactions on Image Processing **21**(5), 2379–2388 (2012)
21. Zhao, M., Chow, T.W., Zhang, H., Li, Y.: Rolling fault diagnosis via robust semi-supervised model with capped l 2, 1-norm regularization. In: Industrial Technology (ICIT), 2017 IEEE International Conference on. pp. 1064–1069. IEEE (2017)



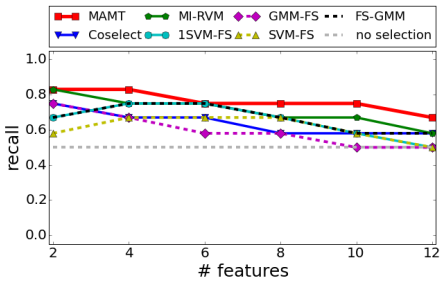
(a) Data A: Precision across different # instances



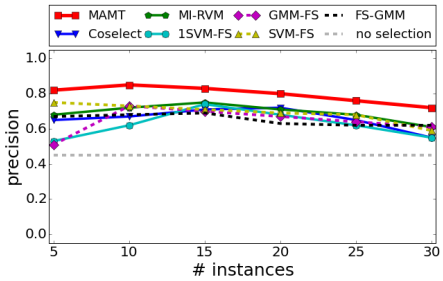
(b) Data A: Recall across different # instances



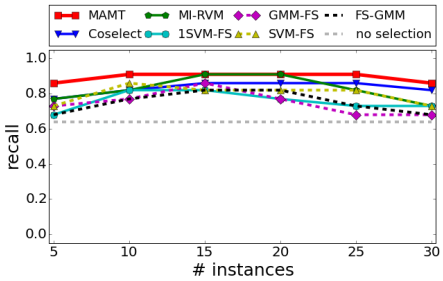
(c) Data A: Precision across different # features



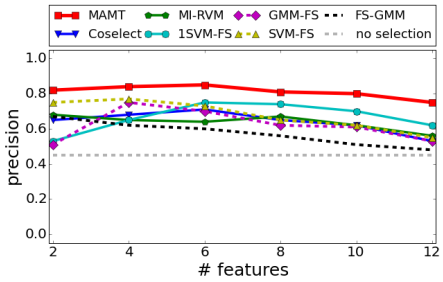
(d) Data A: Recall across different # features



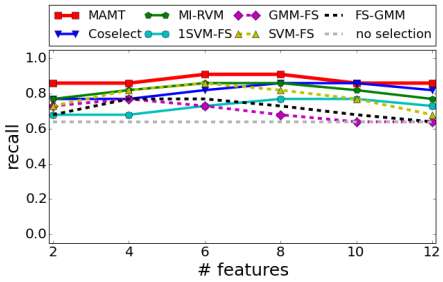
(e) Data B: Precision across different # instances



(f) Data B: Recall across different # instances

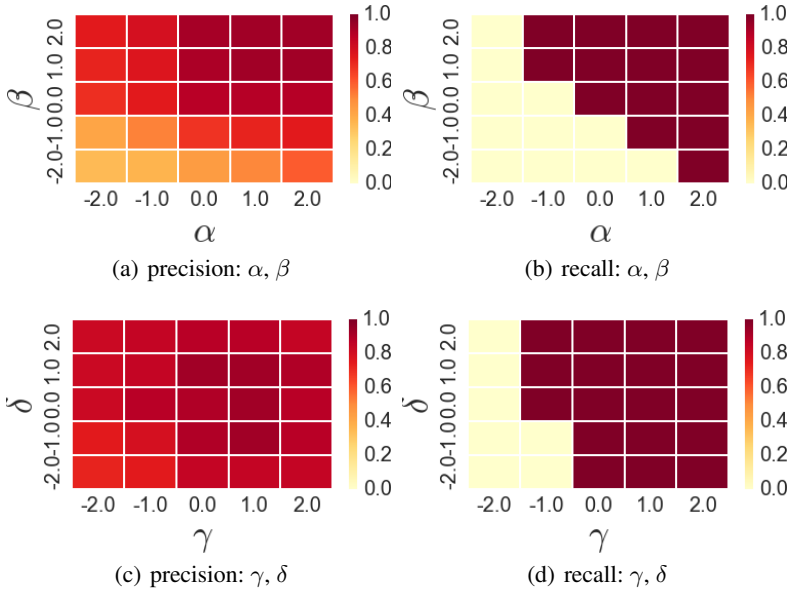


(g) Data B: Precision across different # features



(h) Data B: Recall across different # features

Fig. 3. Classification performance of each method with different p (# top features) and q (# top instances).



(a) precision: α, β (b) recall: α, β

(c) precision: γ, δ (d) recall: γ, δ

Fig. 4. Parameter sensitivity analysis

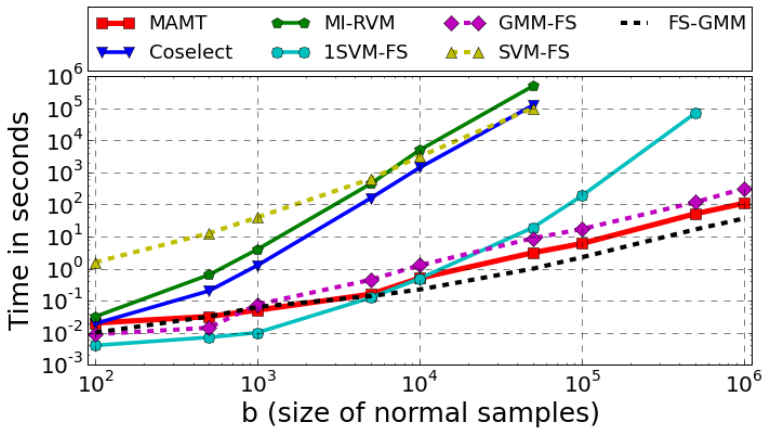


Fig. 5. Scalability Comparison.