Classification with Imperfect Labels for Fault Prediction

Ya Xue
GE Global Research
One Research Circle
Niskayuna, NY 12309 USA
xueya@research.ge.com

David P. Williams
NATO Undersea Research Ctr
Viale San Bartolomeo 400
19126 La Spezia (SP), Italy
williams@nurc.nato.int

Hai Qiu
GE Global Research
1800 Cai Lun Road
Shanghai 201203 China
qiu@research.ge.com

ABSTRACT

Classification techniques have been widely used in fault prediction for industrial systems. However, an inherent issue with this approach is label imperfections in training data, since the line of demarcation between classes is determined based on field expert experience and maintenance capability. To address this issue we propose a noisy-label model in which the labeling noise function is derived from a point of view motivated by reliability analysis. We also present a novel label bootstrapping method that can better reflect the true uncertainty of the labeling process than the standard approach for addressing label imperfections. The proposed technique gives encouraging results on two industrial fault-prediction data sets.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Classification, label imperfection, fault prediction, bootstrapping

1. INTRODUCTION

Fault prediction is an important engineering discipline in system health management of many industrial processes. It involves monitoring system operation conditions in order to predict an impending failure event based on symptoms exhibited in sensor-measured process variables and therefore, plays a critical rule in preventing costly component damage or catastrophic failure.

In a fault prediction system, the collected monitoring data often consists of multiple multivariate time series. Each time series is from a different instance of the same complex engineered system; for example, the data might be from a fleet of ships of the same type, yet operational characteristics may vary across systems due to different degrees of initial wear and manufacturing variation (which are considered natural). A system is operating normally at the start of each time series, and develops a fault at some point during the series. A given time series covers the entire operational lifespan of a system, ending with the last measurement point prior to system failure. Each data point in the time series is a vector containing all sensor measurements, *e.g.,* in an commercial aircraft engine system, exhaust gas temperature (EGT), fuel flow (F/F), engine pressure ratio (EPR), engine core speed (RPM).

Although failure of industrial systems may be caused by various reasons, aging is the dominant culprit. Within a certain time period — *i.e.,* a window — prior to failure, substantial damage accumulates, system conditions deteriorate rapidly and certain events (*i.e.,* sensor-signatures) indicative of failure occur more frequently. Therefore, an approach commonly used in practice is to convert failure prediction into a binary-classification task, by assessing whether a system has entered a 'pre-failure' state or is functioning in a 'normal' state [1–3, 10]. Denoting the time of actual failure as t_f, and setting a window length, w, those data points within the time period $(t_f - w, t_f)$ are labeled as belonging to the 'pre-failure' class ($y = 1$), while the other points are labeled as the 'normal' class ($y = 0$). Conventional classification techniques, such as logistic regression or a support vector machine (SVM), can then be readily applied in conjunction with features extracted from each data point's signature.

The binary-classification approach has advantages in terms of ease of use and robustness to machine variances; however, imperfections in the labeling process are inherent in such an approach. The binarization of engine deterioration into 'pre-failure' and 'normal' states is a simplification of the continuous process that occurs in reality. Moreover, the window length is often suggested by experienced field experts or stipulated based on maintenance capabilities. (For example, in a paper-making machine, specific requirements were imposed to signal a warning at least 60 minutes before the breakage of the paper web [1]; for train-wheel failure prediction, an advance notice of three weeks was desired [10].) Therefore, it remains unknown whether a system has started showing actual signs of failure at the beginning of the prescribed time window.

The standard approach for addressing label imperfections [5] is to modify the probability of a label, y_i, of a data point $\boldsymbol{x}_i$ with estimated labeling error $\epsilon_i \in [0, 0.5]$ by writing

$$p(y_i = 1|\boldsymbol{x}_i, \boldsymbol{w}) \to \epsilon_i + (1 - 2\epsilon_i)p(y_i = 1|\boldsymbol{x}_i, \boldsymbol{w}). \qquad (1)$$

This approach does not address the fundamental issue of imperfect labels. Namely, this approach simply weakens the confidence of the given label. It never considers the case in which the label is assigned the opposite label, as it would be if it were indeed incorrect. Therefore, the subsequent classifier that is constructed is signifi-

KDD4Service '11, San Diego, California, USA

cantly biased toward the initial set of labels supplied with the data set.

In this work we improve the classification approach by introducing a noisy-label model. Besides the binary labels, there is a labeling noise function associated with the data points in a time series. The function is designed for the following purposes: (i) softening the boundary between two classes and therefore increasing system robustness to the window size choice, and (ii) conforming to the natural aging process of industrial systems.

Moreover, we address label imperfections in a different approach from [5]. Multiple training sets are generated in a bootstrapping manner, in which the feature vectors are the same as in the original training set, while the labels could be flipped and the chance of flipping is determined by the associated noise level. A classifier ensemble is learned with the multiple training sets. This bootstrapping approach explicitly allows the labels initially supplied with the data set to take on the opposite values, in effect reflecting the true uncertainty of the labeling process.

2. LABELING NOISE FUNCTION

Central to the proposed approach incorporating imperfect labels is the specification of a labeling noise function (*i.e.,* the labeling error rates). It is desirable to have a function that: (i) corresponds to the underlying physical process, (ii) maximizes the noise level (*i.e.,* $\epsilon = 0.5$) at around $t_f - w$, (iii) minimizes the noise level (*i.e.,* $\epsilon = 0$) at the early stage and the very end of a system's life span, and (iv) is system-specific if data used for training is collected from multiple systems (*e.g.,* a fleet of aircrafts).

Reliability analysis often uses the Weibull distribution [9],

$$r(t) = \exp\left\{-\left(\frac{t}{\alpha}\right)^{\beta}\right\}, \tag{2}$$

where α is the scale parameter and β is the shape parameter. The value of β is related to the cause and behavior of the failure. When $\beta < 1$, the failure rate decreases, indicating a system that tends to fail early, a scenario referred to as "infant mortality." When $\beta = 1$, the failure rate is constant, an indication that the system is failing from random events. When $\beta > 1$, the failure rate increases, suggesting that a system is more likely to fail as time goes on; this scenario accurately models the aging process in many industrial systems.

In this work, we define the labeling noise function, as a function of time, t, as

$$\epsilon(t) = \left\{ \begin{array}{ll} 1 - \exp\left\{-\left(\frac{t}{\alpha}\right)^{\beta}\right\} & \text{if } 0 \leq t \leq t_f - w, \\ \exp\left\{-\left(\frac{t}{\alpha}\right)^{\beta}\right\} & \text{if } t_f - w < t \leq t_f, \end{array} \right. \tag{3}$$

and impose the constraints $\epsilon(t_f - w) = 0.5$ and $\epsilon(t_f) = \rho$, where ρ is a very small positive value (*e.g.,* $\rho = 0.001$).

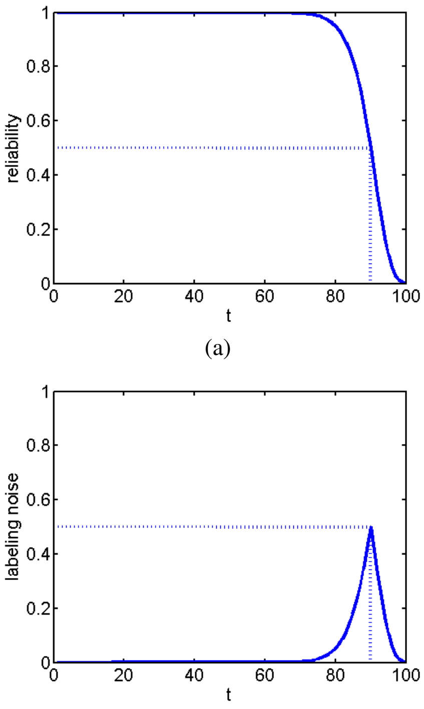
The labeling noise function in (3) is identical to the reliability function in (2) in the 'pre-failure' time window, while it is symmetric to (2) around the line $r(t) = 0.5$ outside the window. The function is maximized at $t = (t_f - w)$ and minimized at $t = 0$ and $t = t_f$. The labeling noise function is plotted in Fig. 1 for the case in which $t_f = 100$, $w = 10$ and $\rho = 0.001$.

The parameters α and β can be obtained by solving the equations

$$\begin{array}{l} \exp\left\{-\left(\frac{t_f}{\alpha}\right)^{\beta}\right\} = \rho, \\ \exp\left\{-\left(\frac{t_f - w}{\alpha}\right)^{\beta}\right\} = 0.5, \end{array}$$

the analytic solution of which is

$$\begin{array}{l} \alpha = \exp\left\{\log(t_f - w) - \frac{1}{\beta}\log(-\log(0.5))\right\}, \\ \beta = \log\left(\frac{\log(\rho)}{\log(0.5)}\right) / \log\left(\frac{t_f}{(t_f - w)}\right). \end{array} \tag{4}$$



(a)

(b)

Figure 1: Example of (a) the reliability function, and (b) the corresponding labeling noise function, for $t_f = 100$, $w = 10$ and $\rho = 0.001$. The values of α and β are calculated using (4).

As noted earlier, the major cause for the failure of industrial systems is aging, a process indicated by $\beta > 1$. In (4), it is guaranteed that $\beta > 1$ when $\rho < 2^{\left(\frac{-t_f}{t_f - w}\right)}$ (the proof is omitted since it is rather straightforward).

Since a system typically deteriorates rapidly once signs of failure appear (*i.e.,* the window size is small), $w \ll t_f$ so the value of $\frac{t_f}{t_f - w}$ is close to unity; therefore the condition that $\rho < 2^{\left(\frac{-t_f}{t_f - w}\right)}$ is easily met.

In the event that data from multiple systems is used in the classifier training process, the maximum of $\frac{t_f}{t_f - w}$ across all systems can be used to determine the value of ρ.

3. CLASSIFICATION WITH LABEL NOISE

Hereafter, we assume we are dealing with a binary (*i.e.,* two-class) classification problem. To address the more nuanced nature of labeling that occurs with real data sets, we describe labels from a quantum perspective. Rather than a label representing a classical bit that takes on the value $y_i = 1$ or $y_i = 0$ with certainty according to its true state, we let a label be represented by a quantum bit, or qubit.

The general state of a qubit is [4]

$$|\psi_i\rangle = \alpha_i|0\rangle + \beta_i|1\rangle \tag{5}$$

where $|0\rangle$ and $|1\rangle$ are the two computational basis states of the qubit and $|\alpha_i|^2 + |\beta_i|^2 = 1$. A qubit can exist in either basis state or even a superposition of the two. However, a *measurement* of a qubit will return only one of the two basis states, $|0\rangle$ or $|1\rangle$, with probability $|\alpha_i|^2$ or $|\beta_i|^2$, respectively. After the measurement, the qubit state collapses irreversibly to the qubit state that is consistent with the measurement.

In this analogy, the measurement of a qubit corresponds to a human labeling a data point. Because of the stochastic nature of the qubit, if a different human instead labeled the data point, it is possible that a different label would be assigned. The case of perfect labeling that is implicitly assumed in most classification problems would correspond to a scenario in which the qubit state of the i-th data point was described with either $\alpha_i = 0$ or $\beta_i = 0$.

Instead, in the case of imperfect labels, when both $\alpha_i \neq 0$ and $\beta_i \neq 0$, it is possible that the i-th data point is assigned either label. The labeling error rate, ϵ_i is intimately connected to the values of α and β. Specifically, if a label originally supplied with a data set is $y_i = 1$ with associated labeling uncertainty $\epsilon_i \in [0, 0.5]$, the general state of the label can be expressed as

$$|\psi_i\rangle = \sqrt{\epsilon_i}\ \ |0\rangle + \sqrt{1-\epsilon_i}\ \ |1\rangle. \tag{6}$$

Taking a measurement of the label's "state" would then produce either label $y_i' = 1$ with probability $1 - \epsilon_i$, or label $y_i' = 0$ with probability ϵ_i.

This quantum interpretation of labels motivates allowing a given label to explicitly take on either label (in a prescribed manner) during the classifier-learning stage. Therefore, we propose a label bootstrapping approach as follows.

Let $\boldsymbol{x}_i \in \mathbb{R}^d$ denote a vector of d features representing the i-th data point of a training set of N such points. Let $y_i \in \{0, 1\}$ denote the label, originally supplied with the data set, that corresponds to the i-th data point, $\boldsymbol{x}_i$. Let $\epsilon_i \in [0, 0.5]$ denote the estimated labeling error associated with the label y_i.

For each data point, $\boldsymbol{x}_i$, a new label (*e.g.,* corresponding to a different human labeler) is then drawn independently from a Bernoulli distribution with parameter $1 - \epsilon_i$,

$$y_i' \sim \mathcal{B}(1-\epsilon_i) = \begin{cases} y_i, & \text{with probability } 1-\epsilon_i; \\ 1-y_i, & \text{with probability } \epsilon_i. \end{cases} \tag{7}$$

Thus, the original label y_i is flipped with probability ϵ_i.

Collect this new set of N training data labels, $\{y_i'\}_{i=1}^N$. Denote the m-th such set of N labels as $Y_{(m)}'$. Conduct M such sampling rounds so that we possess M (potentially, but not necessarily, unique) sets of N labels. Learn a classifier using the data $\{\mathbf{X}, Y_{(m)}'\}$, where $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, and denote it $\boldsymbol{w}_{(m)}$. Repeat the classifier learning using each of the M data sets.

Let $p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{(m)})$ be the probability that a testing data point $\boldsymbol{x}_*$ belongs to class 1 based on classifier $\boldsymbol{w}_{(m)}$ (and in turn, the label set $Y_{(m)}'$). Thus, the final prediction is the average of outputs of the classifier ensemble:

$$p(y_* = 1|\boldsymbol{x}_*, \{\boldsymbol{w}_{(m)}\}_{m=1}^M) = \frac{1}{M}\sum_{m=1}^{M} p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{w}_{(m)}). \tag{8}$$

For the degenerate case in which it is assumed that there is no labeling error, $\epsilon_i = 0 \ \ \forall i$, the labels will never deviate from the initial ones supplied with the data set. That is, $y_i' = y_i$ always, so every learned classifier $\boldsymbol{w}_{(j)}$ would be identical.

Table 1: Number of data points for each operating condition.

Operating Condition					
A	B	C	D	E	F
6954	6882	6771	6881	6859	11571

To determine an appropriate number of sampling rounds, M, to consider, we exploit the work developed for multiple imputation [7], which replaces each missing *feature* value with a set of M samples. The efficiency of an estimate based on M imputations is approximately $(1 + \gamma/M)^{-1}$, where γ is the fraction of missing information for the quantity being estimated [7]. In our case, γ corresponds to the fraction of labels with a non-zero labeling error rate, ϵ_i. Even if $\gamma = 1$, a high efficiency can be achieved with a relatively small number of label sampling rounds, M.

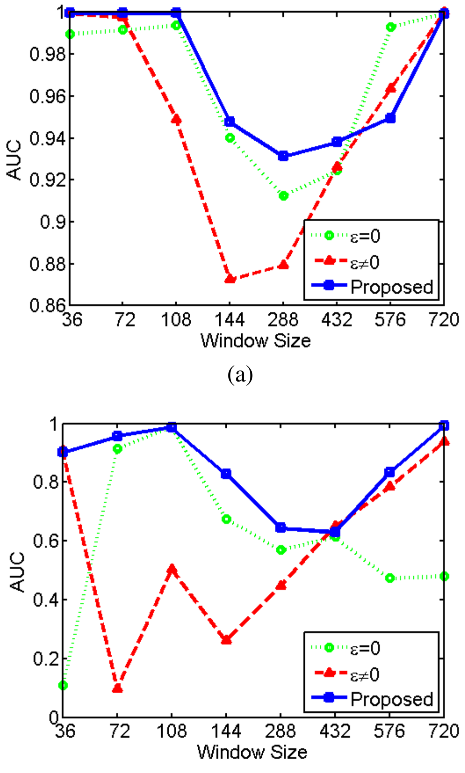
4. EXPERIMENTAL RESULTS

4.1 Data Sets

We test the proposed approach with a measured bearing-wear data set [6] and a simulated aircraft engine-degradation data set [8].

4.1.1 Bearing Failure

The first data set was collected from a real bearing run-to-failure test [6]. The signature of a damaged bearing consists of exponen-



(a)

(b)

Figure 2: Bearing-failure data set results in terms of AUC as a function of the window size, (a) when training with data for bearing C and testing with data for bearing D, and (b) vice versa.

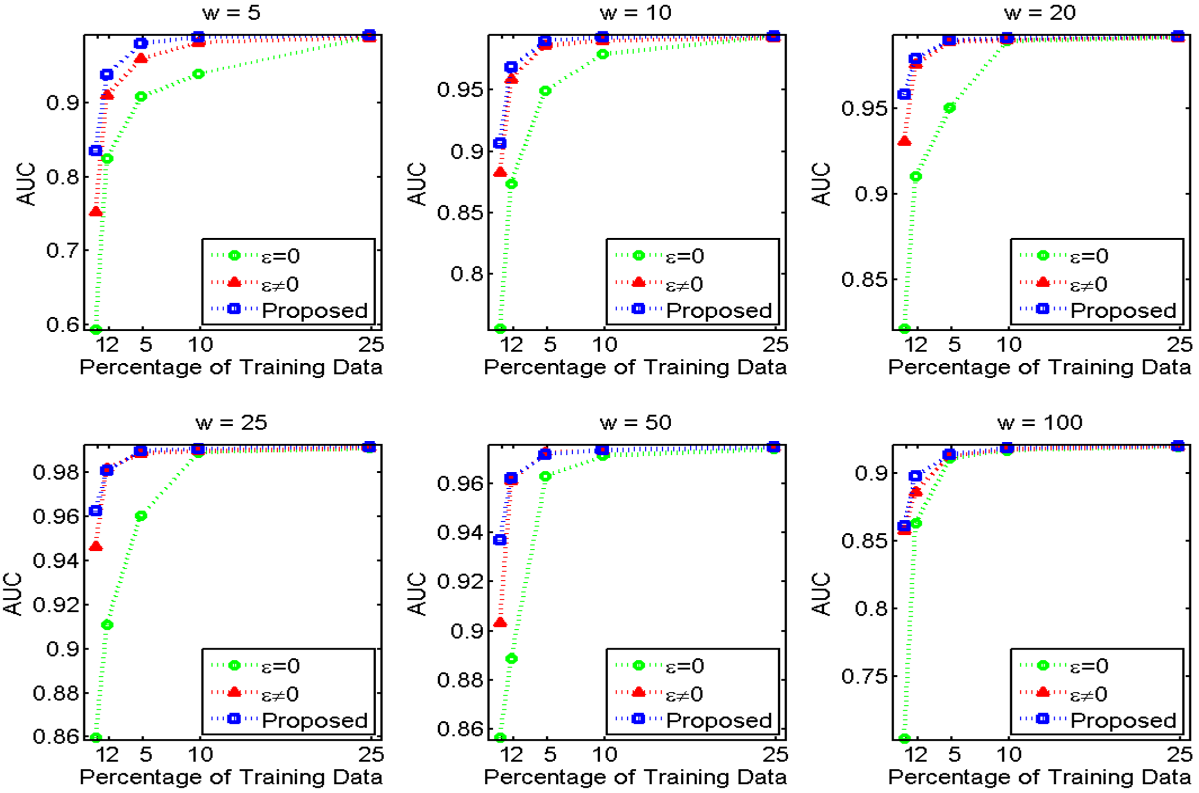


Figure 3: For Operating Condition A, mean AUC (averaged over 50 random training/testing splits) as a function of the amount of training data, for different window sizes, w.

tially decaying ringing that occurs periodically at the characteristic frequency. Because the vibration signal of a defective bearing is usually amplitude modulated at the characteristic defect frequency, widely used features for fault detection include the vibration signals' root mean square (RMS), crest factor, and kurtosis.

The bearing test consisted of four bearings, each of which had two accelerometers — one vertical, Y, and one horizontal, X — installed to measure the vibration signals. The accelerometers took measurements once every 10 minutes. The data sampling rate was 20 kHz and each measurement lasted 1 second. Six features were then extracted from the measured vibration signals: X-channel RMS, Y-channel RMS, X-channel crest factor, Y-channel crest factor, X-channel kurtosis and Y-channel kurtosis. The test was carried out for 35 days (the test paused from time to time) until two of the four bearings, bearings C and D, failed. Therefore the data for each bearing is a time series of length 2516, with each point represented by a 6-dimensional feature vector.

4.1.2 Aircraft Engine Failure

The aircraft engine data set was created by an engine-degradation simulation carried out using C-MAPSS [8], a tool for simulating a realistic large commercial turbofan engine. There are 218 time series (systems) comprising a total of 45918 data points. Each data point is represented by 21 features (sensor measurements). The systems are simulated to operate in one of six operational settings, nominally denoted A, B, C, D, E, and F. Because the operational setting strongly impacts unit performance, each of the six operating conditions is treated as a separate data set; however, each undergoes the same experimental set-up. The number of data points that corresponds to each operating condition is shown in Table 1.

4.2 Experimental Set-up

Three different classification methods are applied to each of the data sets. In all three approaches, a logistic regression algorithm is employed as the classifier. The differences among the methods are in the manner that labeling error is handled.

The first approach, denoted "$\epsilon = 0$," ignores the possible labeling errors and is therefore a standard logistic regression classifier. The second approach, denoted "$\epsilon \neq 0$," assumes imperfect labels, and treats them in the traditional manner [5]. The third approach is the proposed method using $M = 50$ sampling rounds.

All three approaches use the standard logistic function in the prediction phase. This is because — regardless of whether labeling errors were assumed to be present in the training data — no labeling error can exist when no labeling has transpired for testing data points.

For each classification method, the output for a given unlabeled data point is the probability of belonging to the 'pre-failure' class. The Classification accuracy is evaluated by the Area Under the ROC Curve (AUC).

4.3 Results for Bearing Data Set

Only the data for the two failed bearings are used in these experiments, as only a single class is represented by the two healthy bearings. In the first experiment, the data from bearing C is treated as training data while the data from bearing D is treated as testing data. In the second experiment, the training and testing data sets are reversed.

We consider window sizes of 36, 72, 108, 144, 288, 432, 576 and 720 measurements, which correspond to time windows of 0.25, 0.5, 0.75, 1, 2, 3, 4, and 5 days. The resulting AUC as a function of the window size are shown in Fig. 2 for each of the three methods considered. As can be seen in the figures, the proposed approach consistently outperforms the other two approaches.

4.4 Results for Aircraft Engine Data Set

A more extensive set of experiments is conducted using the aircraft engine data set as follows. Performance is assessed for different amounts of training data and different window sizes, w. Specifically, we consider when 1%, 2%, 5%, 10%, and 25% of the data set is training data. Furthermore, we consider window sizes of 5, 10, 15, 20, 25, 50, and 100. For each amount of training data assumed, 50 trials are conducted. In each trial, the data set is randomly partitioned into disjoint training and testing sets.

To allow direct comparisons among the three classification methods, each of the methods uses the same disjoint sets of training and testing data. Moreover, the same training/testing splits of data for a given amount of training data are used for all window sizes. Therefore, the effects of window size on performance can also be observed.

The performance of the considered methods in terms of mean AUC, averaged over 50 random training/testing splits, as a function of the amount of training data, for different window sizes, are shown for operating condition A in Figure 3.

To more clearly observe the relative difference in performance between the proposed approach and each competing approach, Tables 2 and 3 are presented. Positive values in the tables indicate that the proposed approach performed better than the competing approach.

As can be seen from the figures and tables, the proposed method outperforms the two competing methods consistently, and usually to a statistically significant degree. The most significant advantage of the proposed approach occurs when less training data is available.

The evolution of the proposed method's mean AUC (averaged over 50 random training/testing splits) as a function of the number of sampling rounds used, for operating condition A, is shown in Figure 4. As can be seen from the figure, the AUC stabilizes after a relatively small number of sampling rounds. It can also be seen that the AUC stabilizes more quickly when there is more training data used.

All of the above performance assessment for operating conditions B through F was also conducted and similar trends hold for these other five operating conditions. Details are omitted due to space constraints.

5. CONCLUSIONS

A noise-label framework was proposed for fault prediction of industrial systems. The framework is general in that it can be used in conjunction with any classification method. The proposed technique was demonstrated on two industrial fault detection data sets involving bearings and aircraft engines. As for future work, we plan to develop a principled means of determining an appropriate number of sampling rounds in the label bootstrapping algorithm.

6. REFERENCES

[1] Y. Chen and P. Bonissone. Paper web breakage prediction using principal components analysis and classification and regression trees. *US Patent 6,466,877*, October 2002.

[2] X. Hu, N. Eklund, and K. Goebel. A data fusion approach for aircraft engine fault diagnostics. In *Proceedings of ASME Turbo Expo 2007*, May 2007.

[3] S. Létourneau, C. Yang, C. Drummond, E. Scarlett, J. Valdés, and M. Zaluski. A domain independent data mining methodology for prognostics. In *Proceedings of Essential Technologies for Successful Prognostics*, 2005.

[4] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[5] M. Opper and O. Winther. Mean field methods for classification with Gaussian processes. In *Advances in NIPS 11*, pages 309–315, 1998.

[6] H. Qiu, J. Lee, J. Lin, and G. Yu. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, 289(4-5):1066 – 1090, 2006.

[7] D. Rubin. *Multiple imputation for nonresponse in surveys*. John Wiley and Sons, 1987.

[8] A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *Proceedings of International Conference on Prognostics and Health Management 2008*, pages 1–9, Oct. 2008.

[9] W. Weibull. A statistical distribution function of wide applicability. *J. Appl. Mech.-Trans. ASME*, 18(3):293–297, 1951.

[10] C. Yang and S. Létourneau. Learning to predict train wheel failures. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 516–525. ACM, 2005.

Table 2: For Operating Condition A, difference between the proposed method and the method that assumes perfect labels. Table entries show the mean ± one standard deviation, $\mu(z) \pm \sigma(z)$, from 50 random training/testing splits, where $z = \text{AUC}(\text{proposed}) - \text{AUC}(\epsilon = 0)$. Entries in bold are statistically significant at a 95% level according to a paired *t*-test.

Window Size	Percentage of Training Data				
	1	2	5	10	25
5	**0.2419 ± 0.3251**	**0.1132 ± 0.1494**	**0.0715 ± 0.1305**	**0.0503 ± 0.0911**	**0.0017 ± 0.0013**
10	**0.1511 ± 0.1522**	**0.0944 ± 0.1184**	**0.0411 ± 0.0601**	**0.0138 ± 0.0388**	**0.0007 ± 0.0005**
15	**0.1726 ± 0.1647**	**0.0743 ± 0.0761**	**0.0464 ± 0.0707**	**0.0022 ± 0.0018**	**0.0007 ± 0.0005**
20	**0.1368 ± 0.1475**	**0.0690 ± 0.0642**	**0.0394 ± 0.0618**	**0.0018 ± 0.0010**	**0.0006 ± 0.0004**
25	**0.1029 ± 0.1151**	**0.0696 ± 0.0866**	**0.0291 ± 0.0610**	**0.0016 ± 0.0013**	**0.0006 ± 0.0005**
50	**0.0805 ± 0.0845**	**0.0736 ± 0.0758**	**0.0090 ± 0.0312**	**0.0023 ± 0.0018**	**0.0009 ± 0.0006**
100	**0.1568 ± 0.1440**	**0.0349 ± 0.0811**	**0.0027 ± 0.0069**	**0.0018 ± 0.0055**	0.0005 ± 0.0053

Table 3: For Operating Condition A, difference between the proposed method and the method that assumes imperfect labels. Table entries show the mean ± one standard deviation, $\mu(z) \pm \sigma(z)$, from 50 random training/testing splits, where $z = \text{AUC}(\text{proposed}) - \text{AUC}(\epsilon \neq 0)$. Entries in bold are statistically significant at a 95% level according to a paired *t*-test.

Window Size	Percentage of Training Data				
	1	2	5	10	25
5	**0.0836 ± 0.2609**	0.0280 ± 0.1509	**0.0206 ± 0.0652**	**0.0072 ± 0.0121**	**0.0031 ± 0.0037**
10	0.0235 ± 0.1135	0.0093 ± 0.0725	**0.0042 ± 0.0101**	**0.0025 ± 0.0030**	**0.0019 ± 0.0016**
15	**0.0374 ± 0.1157**	0.0105 ± 0.0539	**0.0018 ± 0.0057**	**0.0007 ± 0.0018**	**0.0004 ± 0.0008**
20	**0.0273 ± 0.0959**	0.0033 ± 0.0314	**0.0011 ± 0.0044**	**0.0011 ± 0.0023**	**0.0009 ± 0.0010**
25	**0.0162 ± 0.0674**	-0.0010 ± 0.0134	**0.0010 ± 0.0029**	**0.0010 ± 0.0026**	**0.0007 ± 0.0010**
50	**0.0339 ± 0.0927**	0.0010 ± 0.0197	-0.0006 ± 0.0016	0.0001 ± 0.0015	0.0002 ± 0.0009
100	0.0035 ± 0.0765	0.0117 ± 0.0632	0.0007 ± 0.0064	0.0006 ± 0.0034	**0.0006 ± 0.0017**

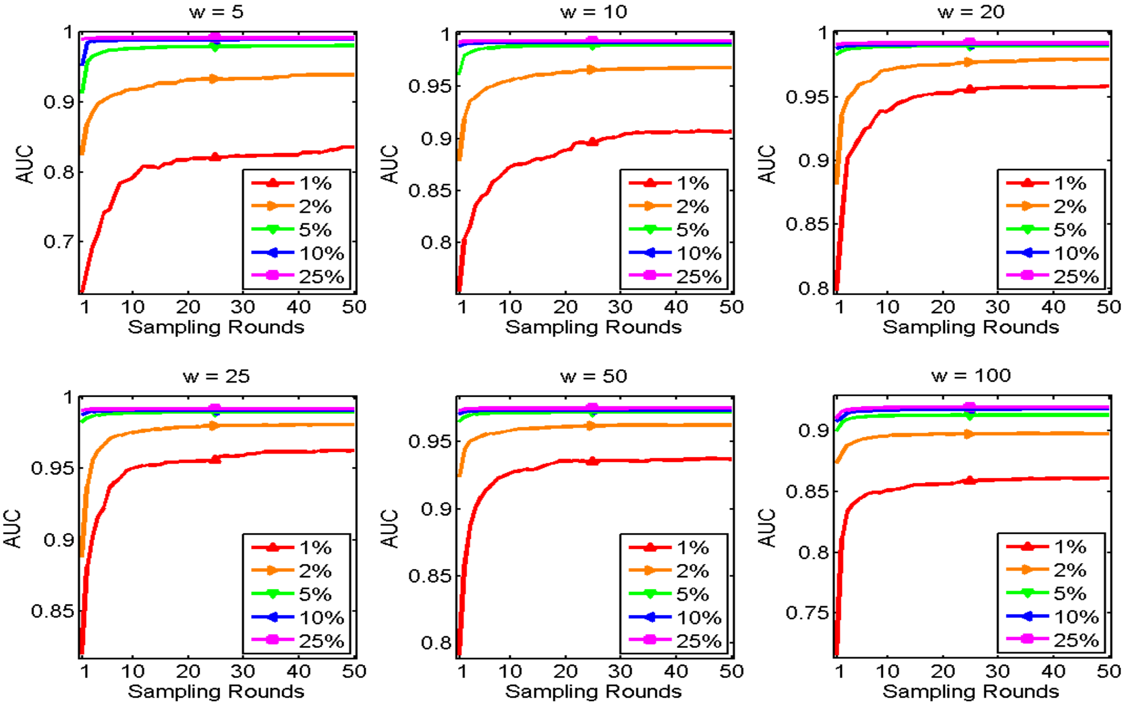


Figure 4: For Operating Condition A, evolution of the proposed method's mean AUC (averaged over 50 random training/testing splits) as a function of the number of sampling rounds used, for different amounts of training data (legend entries) and for different window sizes, w.