frontiers

Wind Turbine Gearbox Failure Detection Through Cumulative Sum of Multivariate Time Series Data

Effi Latiffianti [1,3], **Shawn Sheng** [2] **and Yu Ding** [1*]

[1] *Department of Industrial and System Engineering, Texas A&M University, College Station, Texas, USA*
[2] *National Renewable Energy Laboratory, Golden, Colorado, USA*
[3] *Department of Industrial and System Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*

Correspondence*:
Yu Ding
yuding@tamu.edu

ABSTRACT

The wind energy industry is continuously improving their operational and maintenance practice for reducing levelized costs of energy. Anticipating failures in wind turbines enables early warnings and timely intervention, so that the costly corrective maintenance can be prevented to the largest extent possible. It also avoids production loss owing to prolonged unavailability. One critical element allowing early warning is the ability to accumulate small-magnitude symptoms resulting from the gradual degradation of wind turbine systems. Inspired by the cumulative sum control chart method, this study reports the development of a wind turbine failure detection method with such early warning capability. Specifically, the following key questions are addressed: what fault signals to accumulate, how long to accumulate, what offset to use, and how to set the alarm-triggering control limit. We apply the proposed approach to two years worth of Supervisory Control and Data Acquisition data recorded from five wind turbines. We focus our analysis on gearbox failures detection, in which the proposed approach demonstrates its ability to anticipate failure events with a good lead time.

Keywords: anomaly detection, control chart, CUSUM, early warning, gearbox, minimum spanning tree (MST), unsupervised learning

1 INTRODUCTION

Wind energy is among the fastest growing renewable energy sources. The year of 2020 has been marked as the biggest year ever with a record 93 GW of new installation in that year (GWEC, 2021). IEA (2020) predicted that over 2023-25, average annual wind energy additions could range from 65 GW to 90 GW. Adding to the growth, it was also reported that wind energy has become more cost competitive as indicated by a decreasing trend of the levelized cost of energy (LCOE) (IEA, 2020; U.S. Department of Energy, 2021; GWEC, 2021). A significant portion of LCOE is related to turbine performance (availability and production) and reliability; for instance, Dao et al. (2019) reported a strong and nonlinear relationship between wind turbine reliability and operation and maintenance (O&M) cost. The better the reliability and

performance, the lower the LCOE. The challenge is how to keep the O&M cost low while maintaining a desired level of performance and reliability.

Detecting a component failure relies on identifying anomalies or specific patterns in a dataset. The most commonly used data inputs for anomaly detection in wind turbines are those from the Supervisory Control and Data Acquisition (SCADA) system (de Novaes Pires Leite et al., 2018), failure logs, vibration (Pang et al., 2021; Natili et al., 2021), and occasionally particle counts, status logs, and maintenance record. Chapter 12 of Ding (2019) explains the two major schools of thought of fault diagnosis and anomaly detection: statistical learning based approach (Ahmed et al., 2019, 2021b, 2022; Orozco et al., 2018; Vidal et al., 2018; Moghaddas and Sheng, 2019; Xiao et al., 2022), including control chart approaches (Hsu et al., 2020; Riaz et al., 2020), and physical model-based approach (Guo and Keller, 2020). There are naturally approaches combining the two schools of thought (Yampikulsakul et al., 2014; Guo et al., 2020; Yucesan and Viana, 2021; Hsu et al., 2020). In this study, we focus on the statistical learning-based approaches.

Depending on the availability of data labels in a training set, statistical learning-based approaches can be categorized as supervised and unsupervised learning. Supervised learning needs appropriately labeled data to train a predictive model, which, once a future input is given, predicts whether the future instance is a fault/failure event. Least-squares support vector regression (LS-SVR) (Yampikulsakul et al., 2014), support vector machine or regression (Vidal et al., 2018; Natili et al., 2021), random forest (Hsu et al., 2020; Pang et al., 2021), XG-Boost and long short-term memory (LSTM) networks (Desai et al., 2020; Xiao et al., 2022) are examples of this category. Labeling the training data can be challenging because the fault tags are often added manually. Labeling the training data can also be tricky. Usually, the data point corresponding to the failure instance is labeled as failure and all else are labeled as normal. Consider the typical SCADA data that is recorded every 10 minutes. What such labeling means is that one of the 10-minute data point is labeled as faulty or failure, the data points even only 10 minutes before and after are labeled as normal. But is this a good labeling practice? Since the failures are relatively rare, what such labeling generates is highly imbalanced data, causing many off-the-shelf statistical learning method to render weak detection (Byon et al., 2010; Pourhabib et al., 2015). If more data points than that at the failure instance are to be labeled, then the questions of how many and which data points should be labeled arise but are hard to address. Some work (Williams et al., 2020; Desai et al., 2020) choose to label additional data points prior to the failure instance—so far such action remains *ad hoc*.

When the data label is not available, unsupervised learning is the appropriate approach for anomaly detection. Unsupervised learning relies on the structure or pattern of the dataset to separate any anomalies from the normal data (Wang et al., 2012). One recent developed approach is based on the minimal spanning tree-based distance (Ahmed et al., 2019, 2021b, 2022), which works based on the connectedness of data points with their neighbor and identifies anomalies that are sufficiently different from the majority of its neighbors. Ahmed et al. (2019) demonstrated the application of such an unsupervised learning approach for anomaly detection in hydropower turbines.

In a real-world problem, there is another category approach, referred to as one-class classification (Park et al., 2010) or semi-supervised learning, or in other words, in between the supervised and unsupervised approach. The one-class classification uses only the data under normal operation conditions. This could be because for a turbine, no failure has been recorded yet, or a small number of failures were recorded but the analysts felt they would be better off not using the failure event data. In this case, one can train a model on the normal data and test whether a future observation is conformed with the established normalcy. If not, then such observation is classified as anomalies. Yampikulsakul et al. (2014) is some of such approach, which used the residuals from the modeled normal data to determine abnormality. Technically the control

chart-based methods (Hsu et al., 2020; Riaz et al., 2020; Dao, 2021, 2022; Xu et al., 2020) fall into this category.

Despite the advancement in statistical learning and fault detection, most of the fault detection methods reviewed above do not accumulate effectively small-magnitude early symptoms over time for symptom tracking. As stated earlier, the current approaches respond to a given event individually, to classify it as faulty or non-faulty; such approaches can be called as *point-wise detection*. The lack of symptoms accumulation and tracking explains why the current fault detection systems have very limited early warning capability.

Motivated by the desired capability for symptom accumulation and tracking, we take notice of one quality control method, known as the cumulative sum (CUSUM) method (Page, 1954, 1961). CUSUM is a memory-type control chart and particularly noted for its ability to accumulate consecutive sample points in a process over time and thus effectively detect a small shift in the process that memoryless methods would otherwise fall short of detecting.

Even though CUSUM is a well established approach, it is not easy to apply it to turbine failure detection. The implementation on the complex turbine SCADA data would require some major modification. For this purpose, Dao (2021) adapted a CUSUM-based approach that were typically used to test structural changes in economic and financial time series data. The approach cumulated the standardized residuals after the generator speed is fit through a linear mode and then used the residuals to establish the monitoring chart, of which the control limits were approximated as a function of the data size. This CUSUM-based approach was reported to detect two known failures just a few minutes before the failures took place. Xu et al. (2020) designed an adaptive CUSUM chart to monitor the residuals after the bearing temperature is fit through a random forest model. The alarms from the adaptive CUSUM were issued daily instead of every 10 minutes to reduce the alarm frequency, but there was not reporting that how early the failures could be detected.

Similarly inspired by the concept and method of CUSUM, we set out to develop a symptom-accumulating method for fault detection with early warning capability. We targeted weeks ahead detection rather than minutes or hours ahead. But the plain CUSUM (also referred to as the vanilla version of CUSUM) is not effective in handling the complexities associated with a wind turbine system. The new method needs to address the four specific questions:

- *Which fault signals to accumulate?* The plain CUSUM accumulates the raw measurements, or its sample average. Our research shows that for the turbine SCADA data in a high-dimensional space, accumulating the raw measurements or its sample average is not effective.
- *The use of an offset*. What is actually accumulated is the difference between the anomaly score and an offset value, rather than the anomaly score itself. The offset value is used to prevent the accumulation of background noises. This is the aspect that the new method remains the same as the plain CUSUM method, but the way to choose the offset will be different.
- *How long to accumulate?* In the plain CUSUM method, the accumulation is allowed until the instance of failure events. Should we do the same for turbine fault detection, it will cause too many false positives. We therefore set an accumulation window size to balance the two types of error in detection (i.e., false positives versus false negatives).
- *Setting the control limit*. The control limit is also known as the decision threshold. In the plain CUSUM method, the control limit is chosen for producing the desirable average run length performance metric. In our design, we need to link the decision outcomes with the monetary gains and losses associated with the detection performance metrics, specifically, the true positives, false positives, and false negatives.

To demonstrate and evaluate the proposed approach, we implement the approach on real wind turbine datasets with a focus on gearbox failures. It is not surprising that gearbox is chosen for the demonstration purpose as it is the most popular component to investigate (Guo and Keller, 2020; Mauricio et al., 2020; Guo et al., 2020; Liu et al., 2021). Not only has it been one of the components that contribute most to turbine downtime (Tchakoua et al., 2014; Pinar Pérez et al., 2013; Dao et al., 2019; Pfaffel et al., 2017; Liu et al., 2021), but the replacement cost is also prohibitively high (Liu et al., 2021).

The rest of the article is organized as follows. Section 2 explains the dataset used in this study. Section 3 provides details about the proposed method, i.e., the answer to the aforementioned four questions. Section 4 presents the implementation, results, interpretations, and analysis concerning the proposed method. Finally, Section 5 summarizes this work.

2 DATA

The data we use in this work are retrieved from an online open data source (EDP, 2018b). An account registration is required but such registration is free. Except for the dataset about the geographical location of the turbines, all of the datasets from the open data source, including all that we used, are granted a free use CC-BY-SA license.

The open source provides SCADA datasets of five wind turbines from the same wind farm with a two-years time span, which include: a set of signals recorded from the wind turbines, a set of met tower data, a failures log, and a status log. The datasets were collected from January 2016 through December 2017. All these files are provided with the *Wind Farm 1* tag on the file names. They are split into 2016 and 2017 data. The five wind turbines in the data are named as T01, T06, T07, T09, and T11. All belongs to the same model in a 2 MW class with three-stages planetary/spur gearbox. The cut-in, rated, and cut-out wind speeds are 4 m/s, 12 m/s, and 25 m/s, respectively.

We mainly use signals from wind turbines and the failures log for the analysis. These data are of good quality as there are only a very small amounts of missing values. The two-years recorded signals from five turbines are stored in .csv files, forming a table consists of 521,784 rows and 83 columns when combined together.

The rows are the time series. With a 10-minute time resolution, the number of data points per turbine per year is 52,560 without missing values at all. For five turbines and two years, the total data amount would ideally be 525,600. The actual records of 521,784 account for slightly over 99% of the ideal total data.

The 83 columns include the turbine ID, the time stamp, and 81 environmental (outside the nacelle) and turbine condition (inside the nacelle) variables. The environmental variables include wind speed, ambient temperature, wind direction, among others, whereas the condition variables include turbine components temperature, speed of the rotating components, active power etc. The 81 variables are not all physically distinct. Some are associated with the same physical attribute but provides different statistics, such as the average, minimum, maximum and standard deviation of wind speed in the 10-minute periods.

Among the 521,784 records over two years, there are a total of 28 failures recorded in the log file. The source of failures varies but it can be grouped based on the components, i.e., generator, generator bearing, gearbox, transformer, and hydraulic group.

We focus our analysis on the gearbox failure detection. For this purpose, we split the data into 80:20 of training set and testing set. This means the first 20 months of data are used for training and the four last months are used for test. In other words, the training data covers from January 1, 2016 through August 31,

2017 and the test data covers from September 1, 2017 through December 31, 20217. Four gearbox failures were recorded for the entire 2-years period, two are in the training set and the rest are in the test set. Table 1 lists the gearbox failures information.

Table 1. Gearbox failures log

Turbine ID	Time stamp	Remarks	Training/test
T01	2016-07-18 02:10:00	Gearbox pump damaged	In the training set
T09	2016-10-11 08:06:00	Gearbox repaired	In the training set
T06	2017-10-17 08:38:00	Gearbox bearings damaged	In the test set
T09	2017-10-18 08:32:00	Gearbox noise	In the test set

The datasets were previously given as part of two open challenges: *The EDP Wind Turbine Failure Detection Challenge 2021* and *Hack the Wind 2018*. The turbine signals data were very clean and well organized. As part of the 2021 Challenge, we were supposed to take the data as is. Only some basic data cleaning were performed such as removing all the missing values and checking whether data values are within reasonable physical ranges. No other information was provided to us (e.g., how the data provider pre-processed their data is unknown). We did downsize the data resolution from 10-minute to 1-hour averages and normalize the data prior to the implementation of our proposed method.

3 METHODS

Let us first quickly recap how CUSUM works, which offers the blueprint for the design of our proposed method.

Consider a CUSUM control chart for detecting a change in process mean. Denote by μ_0 the baseline mean. The input signal is the sample observation, denoted by x_t at time t. At any given time, a small sample of multiple x_t's, say five of them, are observed. Then the sample average, $\bar{x}_t$, is computed and used as the input value to a CUSUM chart. The sample size is denoted by n. When $n = 1$, i.e., at any given time, a single observation is made, then the sample average is the same as the original observation, i.e., $\bar{x}_t = x_t$. This $n = 1$ circumstance represents the majority of the cases when CUSUM is applied. Taking x_t, the CUSUM method computes a score, through the CUSUM formula,

$$C_t = \max\{0, C_{t-1} + [x_t - \mu_0 - K]\}, \tag{1}$$

where K is the offset, and the initial condition, C_0, is set to zero. The standard CUSUM separates the upward change from the downward change and thus put a superscript "+" on the above CUSUM score, i.e., C_t^+, and create a slightly modified formula for C_t^- for downward detection.

Apparently, the CUSUM score, C_t, accumulates the difference of $x_t - \mu_0$ and K, where $x_t - \mu_0$ is the fluctuation of the process around its baseline mean. To detect, a control limit H is imposed. The score, C_t, is compared with H, and an alarm is triggered when C_t exceeds H. The two parameters, H and K, are the so-called design parameters of a CUSUM method, which are chosen using the training data. The training data are considered all in control, so that CUSUM method falls in the category of one-class classification or semi-supervised learning.

Our CUSUM-inspired method follows the same procedure, but we need to provide our unique and specific solutions to the four questions raised in Section 1. Figure 1 illustrates the overall flow.

1. *What is used as the anomaly score?*

Denote the turbine data matrix as $\boldsymbol{X}_{m\times p} := \{x_{tj}\}$ where $t = 1, \ldots, m$, $j = 1, \ldots, p$, and for the whole dataset $m = 521,784$ and $p = 83$. For the training data, its m is about 80% of the whole dataset. At any time point, we have a single observation of dimension p, denoted by $\boldsymbol{x}_t := (x_{t1}, x_{t2}, \ldots, x_{tp})^T$.

This $\boldsymbol{x}_t$ cannot be directly plugged into Equation (1), because Equation (1) is for a univariate detection, meaning that the x therein is of dimension $p = 1$. We acknowledge the existence of multivariate CUSUM, which is of the same concept and uses a similar formula as the univariate CUSUM but can take in a multivariate input, i.e., a vector of $\boldsymbol{x}_t$.

Using a multivariate CUSUM does not produce good detection outcomes for turbine failure detection. When we looked into the reasons behind, we think that one previous research provided the explanation. Ahmed et al. (2019) argued that in a multidimensional data space, anomaly and fault detection should not use Euclidean distances to differentiate data instances, because there is a high likelihood that the multidimensional data space embeds a manifold, known as the manifold hypothesis (Fefferman et al., 2016). In fact, the existence of manifold is rather ubiquitous and confirmed in many applications since its discovery in computer vision (Tenenbaum et al., 2000). A manifold is an inherent data structure restricting the reachability of data instances between each other. When such manifold embedding happens, the use of Euclidean distance is no longer appropriate and could mislead a detection system. Section 12.3 of Ding (2019) presents a detailed account of various distance metrics used in differentiating data instances in statistical machine learning. Section 12.3.4 specifically presents an illustration of how Euclidean distance mis-characterizes the similarity between data instances, thereby leading to wrong detection.

In the multivariate CUSUM, the distance matric used is the statistical distance (explained in details in Section 12.3.3 in Ding (2019)), which is a variant of Euclidean distance. For multidimensional data space embedding manifold, using the statistical distance suffers the same problem as using the Euclidean distance.

The solution for addressing this problem is to use a geodesic distance. But the geodesic distance is not always directly computable but often approximated by some other means. Ahmed et al. (2019) propose to use the minimal spanning tree (MST) to approximate the geodesic distance. They argued that using MST provides one of the best approximations because of two good properties of MST—*minimal* ensures the tightest distance and *spanning* implies ergodicity. They demonstrate, using 20 benchmark datasets and in comparison with 13 existing methods, a clear advantage of using the MST-based anomaly detection method.

Therefore, we choose to adopt the MST approach for our anomaly score calculation. The detailed procedure is explained in Section 12.5.2 of Ding (2019), so we will not repeat it here. Also, Ahmed et al. (2019) makes their computer code available (Ahmed et al., 2021a), which facilitates the implementation of the MST-based anomaly score computing algorithm. The computer code includes the construction of the MST on a given dataset, so that users do not need to construct the MST by themselves, either.

If we treat the MST-based anomaly score computing procedure as a black box, the input to the black box is the multivariate vector $\boldsymbol{x}_t$ and the output of the box is a unvariate anomaly score. Note that using the EDP data for computing the anomaly score, we use the combined data from all five turbines together. Let us denote the anomaly score by z_t, which is normalized to take a value between 0 and 1. A greater score implies a higher possibility for a data point to be anomalous. There are a few variants of the MST-based anomaly score due to the continuous development on this topic (Ahmed et al., 2021b, 2022). The specific

variant we used for turbine fault detection is the Local MST (LoMST) originally proposed by Ahmed et al. (2019) and again exhibited in Chapter 12 of Ding (2019).

2. *How long to accumulate and set the offset?*

Here we discuss the second and third questions together.

Like all point-wise detection methods reviewed in Section 1, the MST-based methods and its variants (Ahmed et al., 2019, 2021b, 2022) do not do symptom accumulation. For this reason, it does not include an offset. The concept of accumulation window does not apply, either.

As shown in Equation (1), the offset, K, is explicitly included in CUSUM. On the other hand, CUSUM does not explicitly impose an accumulation window size. CUSUM is designed to detect simple changes like a mean shift. It is the fluctuation around the mean, $x_t - \mu_0$, less the offset K, that gets accumulated. This value can be positive or negative. When $x_t - \mu_0 - K$ is negative for multiple steps, it could turn C_t to zero, which is known as a reset. With this reset mechanism, CUSUM allows its score to continuously accumulate without manually setting the accumulation window size. The duration when C_t is nonnegative can be naturally considered as the *de facto* accumulation window.

The wind turbine failure detection is far more complicated than detecting a mean shift. It is challenging to know around which baseline its anomaly score z_t fluctuates. This means that its counterpart of μ_0 is difficult to decide. What we propose to do is to take a direct difference between z_t and K, i.e., $z_t - K$. But because K, as an offset, is usually smaller than z_t, $z_t - K$ tends to be positive and does not create the reset mechanism as in CUSUM. If we let $z_t - K$ continue accumulating, the accumulation will almost always go exceeding the control limit, once given sufficient time, leading to too many false positives. Because of this, for our detection method, we impose an explicit accumulation window size, denoted by W, so that the accumulation resets when reaching to the limit of the accumulation window.

We propose to choose the offset K based on the probability distribution of the anomaly scores. The basic idea is as follows. In the absence of anomalous events, one anticipates a natural fluctuation in the anomaly scores, more or less like a normal distribution. When the actual anomaly score distribution exhibits a long tail going beyond the natural fluctuation, the anomaly scores corresponding to the long tail are deemed truly anomalous, whereas the normal distribution-like portion, symmetric with respect to the average, are considered corresponding to the background noises. Figure 2 illustrates the idea. The vertical dashed line is the offset chosen, which separates the density curve into two parts—the blue part is for the background noises and the red part for anomalies. The offset is chosen, so that the blue density curve is roughly symmetric and the curve beyond that point becomes almost flat. This selection approach needs visual judgement, so it does entail certain degree of subjectivity. In this regards, it bears a resemblance with the scree plot, which is used to select the number of principal components in a principal component analysis (PCA) (Jolliffe, 2002). The scree plot is also a graph plot based tool that needs a visual judgement to decide on the particular value to choose. Despite such subjectivity, it is still nonetheless the most widely used tool for deciding the number of principal components.

The choice of the accumulation window size W will be chosen by making use of the training data. A number of considerations include how many clusters are produced, how distinguishable the clusters are based on the distance between them, and how many clusters actually predict the true failures in the training data. True positives way ahead a failure event is most desirable. In practice, it is preferred to tolerate certain number of false positives in exchange for detecting the true failure events over the cases of few false positives but many missed detections, because the cost of a missed detection exceeds by a large margin

that of a false positive. The specific trade off between these cost components is to be optimized using a cost/saving utility function, to be discussed in the sequel.

3. *How to set the control limit?*

To determine the control limit, we make use of the information from the failures log to tag the failure time and then use the training data to optimize the control limit. Different from the plain CUSUM chart that optimizes their average run length performance, we adopt a utility function that connects the failure detection performance with monetary gains and losses. The basic idea is to choose a control limit that maximizes the true detection, while at the same time regulating the number of false positives at an acceptable level. A utility function is an objective function that unifying the gains and losses from different actions. The specific utility function is adopted from an open challenge—*Hack the Wind 2018* (EDP, 2018a)—for its practical relevance and realistic monetary parameters (as it is set by a major wind company). We believe the function adopted bears general applicability, although the specific monetary parameters may be adjusted for particular owners/operators and applications.

Three detection possibilities are considered: the true positives (TP), the false positives (FP), and the false negatives (FN). When a true detection happens, a potential saving is in order. The saving amount is related to how early such warning can be issued. Therefore, the TP saving is set in the *Hack the Wind 2018* challenge as

$$TP_{saving} = \sum_{i=1,\ldots,\#\text{TP}} (R_{cost} - M_{cost}) \left(\frac{\Delta t_i}{60} \right), \tag{2}$$

where #TP is the number of true positives, R_{cost} and M_{cost} are the replacement and maintenance cost (also known as repair cost), respectively, and Δt_i is the number of days ahead the failure time. The saving function in Equation (2) assumes that 60 days before the failure event is where the maximum saving can be achieved. The saving decreases as the detection happens closer to the instance of the failure.

When a false negative happens, it means a miss detection. Then, the cost is the replacement cost, which is the most costly option. When a false detection happens, the consequence is an inspection cost, denoted by I_{cost}. As such, the FN and FP cost components are, respectively,

$$\begin{aligned} FN_{cost} &= \#\text{FN} \times R_{cost}, \\ FP_{cost} &= \#\text{FP} \times I_{cost}, \end{aligned} \tag{3}$$

where #FN and #FP are the number of the false negatives and false positives, respectively. The utility function, $U(H)$, combines all the saving and cost elements, where H is the control limit. The control limit is decided by maximizing the utility function, i.e.,

$$\begin{aligned} &\max_{H} U(H), \\ &\text{where} \quad U(H) = TP_{saving} - FN_{cost} - FP_{cost}. \end{aligned} \tag{4}$$

In the *Hack the Wind 2018* challenge, the early warning is assumed up to 60 days in advance. In our study, we extend it to 90 days in advance. This extension is mainly because the source of failures in a wind turbine gearbox varies, from one that is temporary and random to a wear-out failure due to a longtime running in poor working conditions (Liu et al., 2021). The extension is expected to capture this wear-out type of failure. When a failure is not detected before the event, it is considered as a false negative. When an

alarm is issued but with no corresponding failure in the dataset, it is considered a FP. In the *Hack the Wind 2018* challenge, a detection within two days of the failure event is also considered a miss detection, i.e., an FN, as it is too close to the failure event to prevent the failure from happening. We keep the same treatment in this study.

4. *Additional Remarks*

As a summary, our CUSUM-inspired failure detection method entails the following main steps:

1. Compute the anomaly scores for all data points of interest; both training and test sets.
2. Subtract the offset value from the raw anomaly scores, so as to flag only those data points with high anomaly scores as anomalies.
3. Using the training data, determine the accumulation window, a maximum time between two consecutive anomaly data points of which the anomaly scores are to be accumulated.
4. Again use the training data to optimize for the control limit H, beyond which the accumulated anomaly score triggers an alarm.

Figure 3 illustrates the step-by-step process of our proposed method when it is applied to the data of T09. In the actual analysis reported in the next section, we use the data pooled from all five turbines, but the concept and method remain the same.

4 RESULTS AND DISCUSSION

We implement our proposed method aiming at detecting gearbox failures in wind turbines. In this section, we start off explaining further implementation details and the parameters chosen in the proposed detection method. After that, we will discuss the results and evaluate the performance of the method.

4.1 Implementation Details and Parameters

Prior to the implementation of the proposed method, we perform data preprocessing and variables selection. The data is originally a 521,784$\times$83 matrix. We downsize the number of rows by aggregating data from its original 10-minute temporal resolution to 1-hour averages. This proprecessing reduces the number of rows to 87,052 for all five turbines, or about 17,400 rows per turbine.

Variables selection is important for screening the available variables into a smaller set of meaningful and highly relevant variables. We conducted various tests to reduce variables that have a high collinearity with other variables. In the end, we select a subset that consists of gearbox oil temperature, gearbox bearing temperature, nacelle temperature, rotor speed, ambient wind direction, and active power. We perform our detection method, as explained in Section 3, on the data with this subset of variables.

The LoMST anomaly score is computed using the code provided by Ahmed et al. (2021a). In producing the LoMST scores, a local neighborhood size is needed; for that we use 25, which is an empirical choice. The rest of the parameters used in the detection method are: (1) the offset $K = 0.3$ (2) the accumulation window size, $W = 7$ days, and (3) the control limit, $H = 8$. In deciding H, the following cost parameters are used in the utility function: R_{cost} = €100,000, M_{cost} = €20,000, and I_{cost} = €5,000. These cost parameters are taken from the *Hack the Wind 2018* challenge (EDP, 2018a).

4.2 Results

Figure 4 presents the results from the implemented method on the dataset. Recall that there are four gearbox failures recorded within the 2-year time span—two are in the training set and the other two are in the test set. All four failures can be detected by the proposed method.

Table 2 presents the time of alarm of the gearbox failures based on the results in Figure 4. The early warning lead time, measured by the alarm-to-failure time, ranges from 21 to 89 days. The average warning lead time based on the training set is 55 days. We also took a close look at the nature of the gearbox failures. Recall that Liu et al. (2021) classified the faults in the wind turbines gearbox into two categories: the wear-out failures and temporary random faults. Note further, from Table 2, that the first and fourth failures are caused by gearbox pump and noise, the second failure's source is not known, and the third is from the gearbox bearing. A bearing failure is typically a wear-out type that builds up slowly. Our method successfully anticipates this failure, with a 89-day lead time. The second failure is most likely of the similar type, but we do not have adequate information, based on the failure remark in the dataset, to be assertive one way or the other. The other two failures—the pump and the noise—are more of temporary random faults. The lead times of detection are shorter than 60 days.

Table 2. Gearbox failures detection results summary.

Failure ID	Turbine ID	Failure time	Alarm time	Alarm-to-failure
1	T01	2016-07-18 02:10:00	2016-06-27 09:00:00	21 days
2	T09	2016-10-11 08:06:00	2016-07-14 14:00:00	89 days
3	T06	2017-10-17 08:38:00	2017-07-20 18:00:00	89 days
4	T09	2017-10-18 08:32:00	2017-08-26 21:00:00	53 days

4.3 Method Evaluation

Our proposed method works well in anticipating gearbox failures on the given data. Since the method does produce both false positives, in addition to the true detection, we should evaluate the final performance of the method using the total saving formula in Equation 4.

Table 3 presents the saving calculating, as a result of detection performance metrics. We present two scenarios: one uses a common control limit and the other uses individual control limits for each turbine. Following the approach that decides the common control limit for all turbines, the turbinewise control limit could be decided as: 8, 10.5, 18.8, 15.5, and 22 for T01, T06, T07, T09, and T11, respectively. Recall that the common control limit is 8. It turns out that the common control limit works better. It does not have any miss detection, i.e., #FN=0. As a result, its expected saving for the test data is a positive €130K. By comparison, using the turbinewise control limits would miss one true gearbox failure in the test data and would therefore result in a negative €25K test case saving, or equivalent, a €25K expense for the test cases.

The analysis presented in Table 3 also reaffirms an important message we articulated earlier, which is that detecting a true failure is far more beneficial than reducing a few additional false positives. If we look at the number of false positives (column #FP) in Table 3, we can see that using the turbinewise control limits is very good at reducing the number of false alarms. Yet, the one missing detection costs much more than reducing three false positives in the test data. This is much expected, as the replacement cost, the consequence of a missed detection, is twenty times of the inspection cost, the consequence of

Table 3. Calculated savings from detection.

dataset	#TP	#FP	#FN	Calculated saving
Using common threshold for all turbines				
Training set	2	11	0	€53,000.00
Test set	2	4	0	€130,666.67
Training & test set	4	15	0	€183,666.67
Using individual turbine threshold				
Training set	2	2	0	€94,000.00
Test set	1	1	1	- €25,000.00
Training & test set	3	3	1	€69,000.00

a false positive. This cost imbalance is generally true, although the specific numerical ratio depends on applications.

To evaluate the merit of the proposed CUSUM-LoMST method, we compare it with the following alternatives:

- Pointwise LoMST. This is the original LoMST method without accumulation.
- Traditional CUSUM, based on (Dao, 2021). This is the CUSUM without using the LoMST score and other modifications made in this paper.
- Correlation-based feature selection, before applying the proposed CUSUM-LoMST method.
- PCA-based feature selection, before applying the proposed CUSUM-LoMST method.

The third and fourth alternatives in the above list are suggested by one of the reviewers for testing whether different feature selection approaches could help improve the performance of the proposed CUSUM-LoMST method. The correlation-based feature selection is based on (Castellani et al., 2021), which is to include the features that have a high Pearson correlation score with the gearbox speed and gearbox bearing temperature. The PCA-based feature selection is to use the first few significant principal components of the features selected by using the Pearson correlation score.

Table 4 presents the failure detection results and the respective savings. From the results we can see that the two alternative feature selection approaches do not help in this case. Their main shortcoming is that they produce more false alarms as compared to the proposed approach. On a positive note, both feature selection approaches still yield positive savings on the test data, as they are able to detect the two true failures.

The pointwise LoMST and the tradition CUSUM method do not perform well. The principal problem of the pointwise LoMST is its inability to detect the true failures on the test data. This is not surprising, as from the get-go, our argument is that the pointwise methods would miss the failure events without accumulating the signals. The traditional CUSUM method (Dao, 2021) was able to successfully detect the true failures in the test data but did so at the expense of producing a lot more false alarms. In fact, traditional CUSUM produced more false alarms than all other alternatives in comparison. Figure 5 presents a small section (the first quarter of Year 1) of the CUSUM plot. We notice that the plot suffers from seasonal effect and it has to be reset several times; otherwise the CUSUM score will stay outside the control limits for very long time. The high number of false alarms eventually forces the traditional CUSUM method to enter the region of economic loss (or negative savings) on the test data.

Table 4. Comparison of four alternative methods with the proposed CUSUM-LoMST.

Training set				
	#TP	#FP	#FN	Calculated saving
CUSUM-LoMST (proposed method)	2	11	0	€53,000.00
Pointwise LoMST	1	13	1	−€139, 666.67
Traditional CUSUM	2	61	0	−€219,666.67
Correlation-based feature selection	1	13	1	−€85,000.00
PCA-based feature selection	1	23	1	−€135,000.00
Test set				
	#TP	#FP	#FN	Calculated saving
CUSUM-LoMST (proposed method)	2	4	0	€130,666.67
Pointwise LoMST	0	5	2	−€225,000.00
Traditional CUSUM	2	26	0	−€6,000.00
Correlation-based feature selection	2	14	0	€52,666.67
PCA-based feature selection	2	11	0	€71,666.67

5 CONCLUSION

We propose a method that combines the use of LoMST and a CUSUM approach for detecting anomalies and failures. This method is applied to two years worth of wind turbine data for detecting gearbox failures in wind turbines. Compared to pointwise detection methods without accumulation or a traditional CUSUM method without adaptation to the wind turbine specifics, the proposed CUSUM-LoMST method produces better detection outcomes and longer lead time, leading to more savings to the industry.

Through this study, we would like to offer the following insights:

- Correctly detecting true failure events with sufficient lead time is far more important than keeping the number of false alarms low. This is not to say that reducing false alarms is not important. But a detection method that does not detect is practically useless. Until the day when one reaches the ideal state of having both high detection rates and low false positive rates, the emphasis should be prioritized towards detection capability.
- Accumulating small-magnitude symptoms is key to enable early warning capability. But the very action of accumulation exacerbates the delicate trade off between true detections, false positives, and false negatives, which means that accumulation-capable methods need a careful design to strike the right balance.
- For the detection in a multidimensional space, selecting the right variables and reducing them further into a scalar anomaly score for accumulation is a challenging job but the final detection performance depends heavily on such choices. Our proposed use of the MST-based anomaly scores appears advantageous, at least for the data we tested. But we acknowledge that on this aspect much more research is needed to make the treatment systematic and less subjective.

We did apply the proposed CUSUM-LoMST detection method to other faults in the EDP Open Data, which includes those from transformer, generator, generator bearing, and hydraulic group. In those detections, our proposed method remains strong in terms of detection power, but the number of false alarms increases too fast, overwhelming the benefit of the detections and sometimes tipping the balance over towards an overall loss. Continuing the improvement so that the right balance of true detections and false alarms can be reached is indeed our ongoing research pursuit.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

EL, SS and YD contribute to the formulation of the problem. EL and YD contribute to the design of the solution method. EL contributes to the implementation and fine-tuning of the method, while SS and YD provide technical advices. EL and YD contribute to the writing of the paper and SS contribute to the editing and comment of the paper.

FUNDING

Latiffianti's research is supported by a Fulbright Scholarship in collaboration with the Indonesian Government (DIKTI-Funded Fulbright). Ding's research is partially supported by NSF grants IIS-17411731 and CCF-1934904. This work was authored (in part) by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, under Contract No. DE-AC36-08GO28308 for the U.S. Department of Energy. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

ACKNOWLEDGMENTS

The authors acknowledge Dr. Sarah Barber, Wind Energy lead at the Eastern Switzerland University of Applied Sciences and President of the Swiss Wind Energy R&D Network, and her WeDoWind platform for hosting a 2021 Data Challenge using the EDP Open Data. The solution method was conceived when the first author participated in the 2021 Data Challenge under the supervision of the two senior authors.

DATA AVAILABILITY STATEMENT

The datasets analyzed for this study can be found in the Open EDP website `https://opendata.edp.com/explore/?refine.keyword=visible&sort=modified`.

REFERENCES

Ahmed, I., Dagnino, A., and Ding, Y. (2019). Unsupervised anomaly detection based on minimum spanning tree approximated distance measures and its application to hydropower turbines. *IEEE Transactions on Automation Science and Engineering* 16, 654–667. doi:10.1109/TASE.2018.2848198

[Dataset] Ahmed, I., Dagnino, A., and Ding, Y. (2021a). Dataset and code for "Unsupervised anomaly detection based on minimum spanning tree approximated distance measures and its application to hydropower turbines". Zenodo Data Sharing Platform. doi:10.5281/zenodo.5525295

Ahmed, I., Galoppo, T., Hu, X., and Ding, Y. (2022). Graph regularized autoencoder and its application in unsupervised anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* in press. doi:10.1109/TPAMI.2021.3066111

Ahmed, I., Hu, X. B., Acharya, M. P., and Ding, Y. (2021b). Neighborhood structure assisted non-negative matrix factorization and its application in unsupervised point-wise anomaly detection. *Journal of Machine Learning Research* 22(34), 1–32

Byon, E., Shrivastava, A., and Ding, Y. (2010). Ensemble classifier for highly imbalanced class sizes. *IIE Transactions* 42, 288–303. doi:10.1080/07408170903228967

Castellani, F., Astolfi, D., and Natili, F. (2021). SCADA data analysis methods for diagnosis of electrical faults to wind turbine generators. *Applied Sciences* 11, 3307. doi:10.3390/app11083307

Dao, C., Kazemtabrizi, B., and Crabtree, C. (2019). Wind turbine reliability data review and impacts on levelised cost of energy. *Wind Energy* 22, 1848–1871. doi:https://doi.org/10.1002/we.2404

Dao, P. B. (2021). A CUSUM-based approach for condition monitoring and fault diagnosis of wind turbines. *Energies* 14, 3236. doi:10.3390/en14113236

Dao, P. B. (2022). Condition monitoring and fault diagnosis of wind turbines based on structural break detection in scada data. *Renewable Energy* 185, 641–654. doi:https://doi.org/10.1016/j.renene.2021.12.051

de Novaes Pires Leite, G., Araújo, A. M., and Rosas, P. A. C. (2018). Prognostic techniques applied to maintenance of wind turbines: A concise and specific review. *Renewable and Sustainable Energy Reviews* 81, 1917–1925. doi:https://doi.org/10.1016/j.rser.2017.06.002

Desai, A., Guo, Y., Sheng, S., Phillips, S., and Williams, L. (2020). Prognosis of wind turbine gearbox bearing failures using SCADA and modeled data. In *Proceedings of the Annual Conference of the PHM Society 2020*. vol. 12, 1–10. doi:10.36001/phmconf.2020.v12i1.1292

Ding, Y. (2019). *Data Science for Wind Energy* (Boca Raton, FL, USA: Chapman & Hall). doi:https://doi.org/10.1201/9780429490972

EDP (2018a). Hack the Wind 2018 - Algorithm Evaluation Accessed August 17, 2021.

[Dataset] EDP (2018b). Wind farm 1. Accessed June 30, 2021.

Fefferman, C., Mitter, S., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society* 29, 983–1049

Guo, Y. and Keller, J. (2020). Validation of combined analytical methods to predict slip in cylindrical roller bearings. *Tribology International* 148, 106347. doi:https://doi.org/10.1016/j.triboint.2020.106347

Guo, Y., Sheng, S., Phillips, C., Keller, J., Veers, P., and Williams, L. (2020). A methodology for reliability assessment and prognosis of bearing axial cracking in wind turbine gearboxes. *Renewable and Sustainable Energy Reviews* 127, 109888. doi:https://doi.org/10.1016/j.rser.2020.109888

GWEC (2021). *Global Wind Report* (Brussels: Global Wind Energy Council)

Hsu, J.-Y., Wang, Y.-F., Lin, K.-C., Chen, M.-Y., and Hsu, J. H.-Y. (2020). Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning. *IEEE Access* 8, 23427–23439. doi:10.1109/ACCESS.2020.2968615

IEA (2020). *Renewables 2020 - Analysis and Forecast to 2025* (Paris: International Energy Agency)

Jolliffe, I. T. (2002). *Principal Component Analysis* (New York, USA: Springer), 2nd edn.

Liu, W. Y., Gu, H., Gao, Q. W., and Zhang, Y. (2021). A review on wind turbines gearbox fault diagnosis methods. *Journal of Vibroengineering* 23, 26–43. doi:10.21595/jve.2020.20178

Mauricio, A., Sheng, S., and Gryllias, K. (2020). Condition monitoring of wind turbine planetary gearboxes under different operating conditions. *Journal of Engineering for Gas Turbines and Power* 142, 031003. doi:10.1115/1.4044683

Moghaddas, R. and Sheng, S. (2019). An anomaly detection framework for dynamic systems using a bayesian hierarchical framework. *Applied Energy* 240, 561–582. doi:https://doi.org/10.1016/j.apenergy.2019.02.025

Natili, F., Daga, A. P., Castellani, F., and Garibaldi, L. (2021). Multi-scale wind turbine bearings supervision techniques using industrial SCADA and vibration data. *Applied Sciences* 11, 6785. doi:10.3390/app11156785

Orozco, R., Sheng, S., and Phillips, C. (2018). Diagnostic models for wind turbine gearbox components using SCADA time series data. In *Proceeding of the 2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*. 1–9. doi:10.1109/ICPHM.2018.8448545

Page, E. S. (1954). Continuous inspection schemes. *Biometrics* 41, 100–115

Page, E. S. (1961). Cumulative sum control charts. *Technometrics* 3, 1–9

Pang, B., Tian, T., and Tang, G.-J. (2021). Fault state recognition of wind turbine gearbox based on generalized multi-scale dynamic time warping. *Structural Health Monitoring* 20, 3007–3023. doi:10.1177/1475921720978622

Park, C., Huang, J. Z., and Ding, Y. (2010). A computable plug-in estimator of minimum volume sets for novelty detection. *Operations Research* 58, 1469–1480

Pfaffel, S., Faulstich, S., and Rohrig, K. (2017). Performance and reliability of wind turbines: A review. *Energies* 10, 1904. doi:10.3390/en10111904

Pinar Pérez, J. M., García Márquez, F. P., Tobias, A., and Papaelias, M. (2013). Wind turbine reliability analysis. *Renewable and Sustainable Energy Reviews* 23, 463–472. doi:https://doi.org/10.1016/j.rser.2013.03.018

Pourhabib, A., Mallick, B. K., and Ding, Y. (2015). Absent data generating classifier for imbalanced class sizes. *Journal of Machine Learning Research* 16, 2695–2724

Riaz, M., Abbasi, S. A., Abid, M., and Hamzat, A. K. (2020). A new HWMA dispersion control chart with an application to wind farm data. *Mathematics* 8, 2136. doi:10.3390/math8122136

Tchakoua, P., Wamkeue, R., Ouhrouche, M., Slaoui-Hasnaoui, F., Tameghe, T. A., and Ekemb, G. (2014). Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges. *Energies* 7, 2595–2630. doi:10.3390/en7042595

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323

U.S. Department of Energy (2021). *Land-Based Wind Market Report: 2021 Edition* (Oak Ridge: U.S. Department of Energy)

Vidal, Y., Pozo, F., and Tutivén, C. (2018). Wind turbine multi-fault detection and classification based on SCADA data. *Energies* 11, 3018. doi:10.3390/en11113018

Wang, X., Wang, X. L., and Wilkes, D. M. (2012). A minimum spanning tree inspired clustering-based outlier detection technique. In *Advances in Data Mining. Applications and Theoretical Aspects (Lecture Notes in Computer Science)*, ed. P. Perner (Berlin: Springer). 209–223

Williams, L., Phillips, C., Sheng, S., Dobos, A., and Wei, X. (2020). Scalable wind turbine generator bearing fault prediction using machine learning: A case study. In *Proceedings of the 2020 IEEE International Conference on Prognostics and Health Management (ICPHM)*. 1–9. doi:10.1109/ICPHM49022.2020.9187050

Xiao, X., Liu, J., Liu, D., Tang, Y., and Zhang, F. (2022). Condition monitoring of wind turbine main bearing based on multivariate time series forecasting. *Energies* 15, 1951. doi:10.3390/en15051951

Xu, Q., Lu, S., Zhai, Z., and Jiang, C. (2020). Adaptive fault detection in wind turbine via RF and CUSUM. *IET Renewable Power Generation* 14, 1789–1796. doi:https://doi.org/10.1049/iet-rpg.2019.0913

Yampikulsakul, N., Byon, E., Huang, S., Sheng, S., and You, M. (2014). Condition monitoring of wind power system with nonparametric regression analysis. *IEEE Transactions on Energy Conversion* 29, 288–299. doi:10.1109/TEC.2013.2295301

Yucesan, Y. A. and Viana, F. A. (2021). Hybrid physics-informed neural networks for main bearing fatigue prognosis with visual grease inspection. *Computers in Industry* 125, 103386. doi:https://doi.org/10.1016/j.compind.2020.103386

FIGURE CAPTIONS

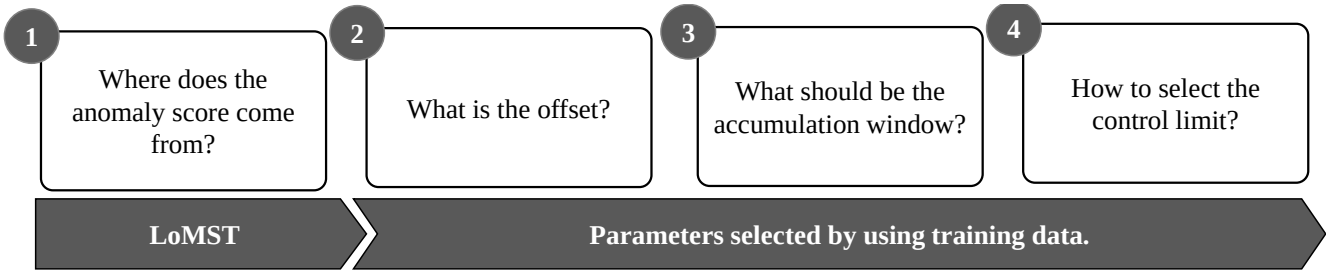

Figure 1. The flowchart of the main steps in the CUSUM-inspired detection method.

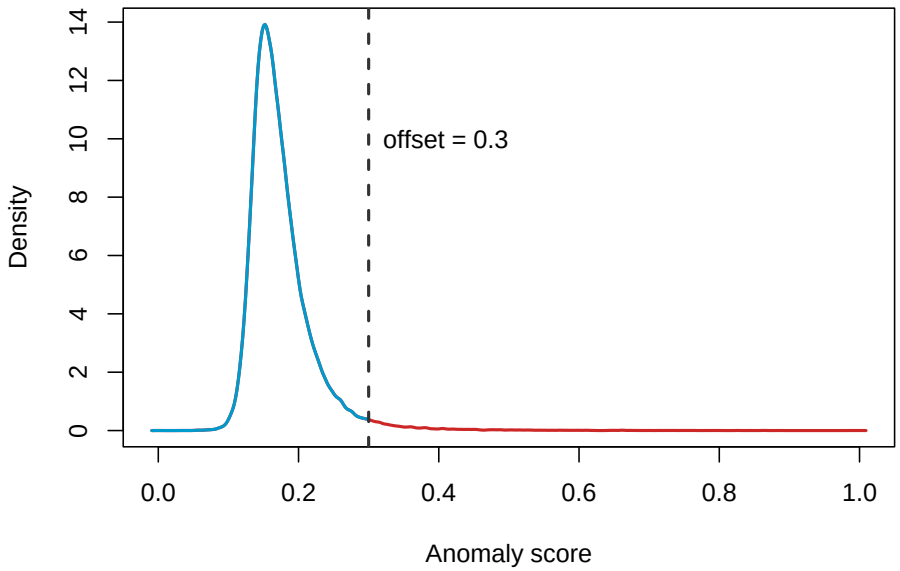

Figure 2. The density curve of the anomaly scores of the entire data points. The vertical dashed line marks the offset, which is 0.3 for this particular analysis.

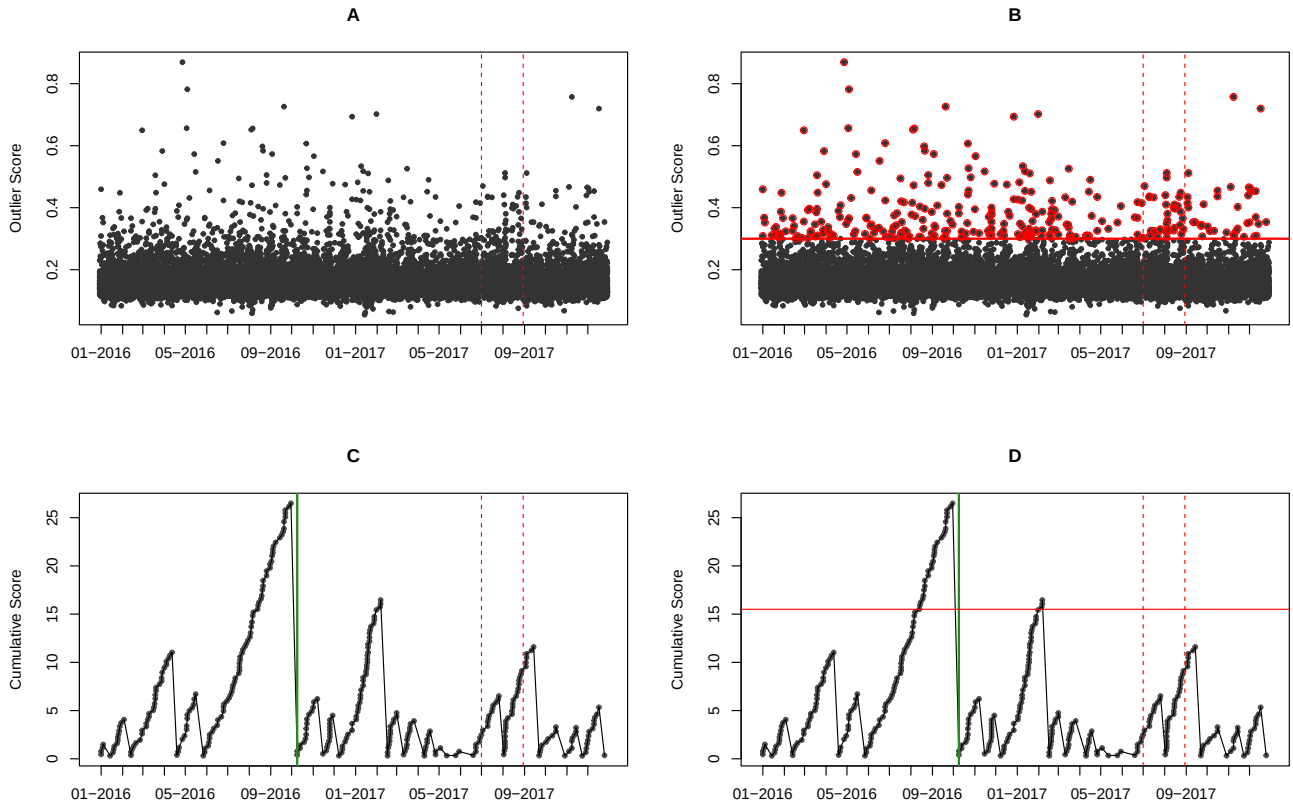

Figure 3. Illustration of the actions in the proposed method. In this example, only T09 data is used. In each of the figures, there are two vertical dashed red lines. The one to the right is the time boundary between the training and test data. The one to the left is the 60-day mark before the test set. **A** plots the output of LoMST anomaly score calculation. **B** adds the offset, so that only anomaly scores above the offset are accumulated in the next step. Those scores are highlighted in red color. **C** is the plot of cumulative anomaly score, where one can see the effect of accumulation and tracking. The vertical green line indicates the time of a true gearbox failure recorded in the training data. **D** adds the control limit for detection, which is the horizonal red line. With this control limit, it flags two alarms in the training data and none in the test data. One of the two alarms is a true positive, with the early warning lead time of 89 days and the other is a false alarm.

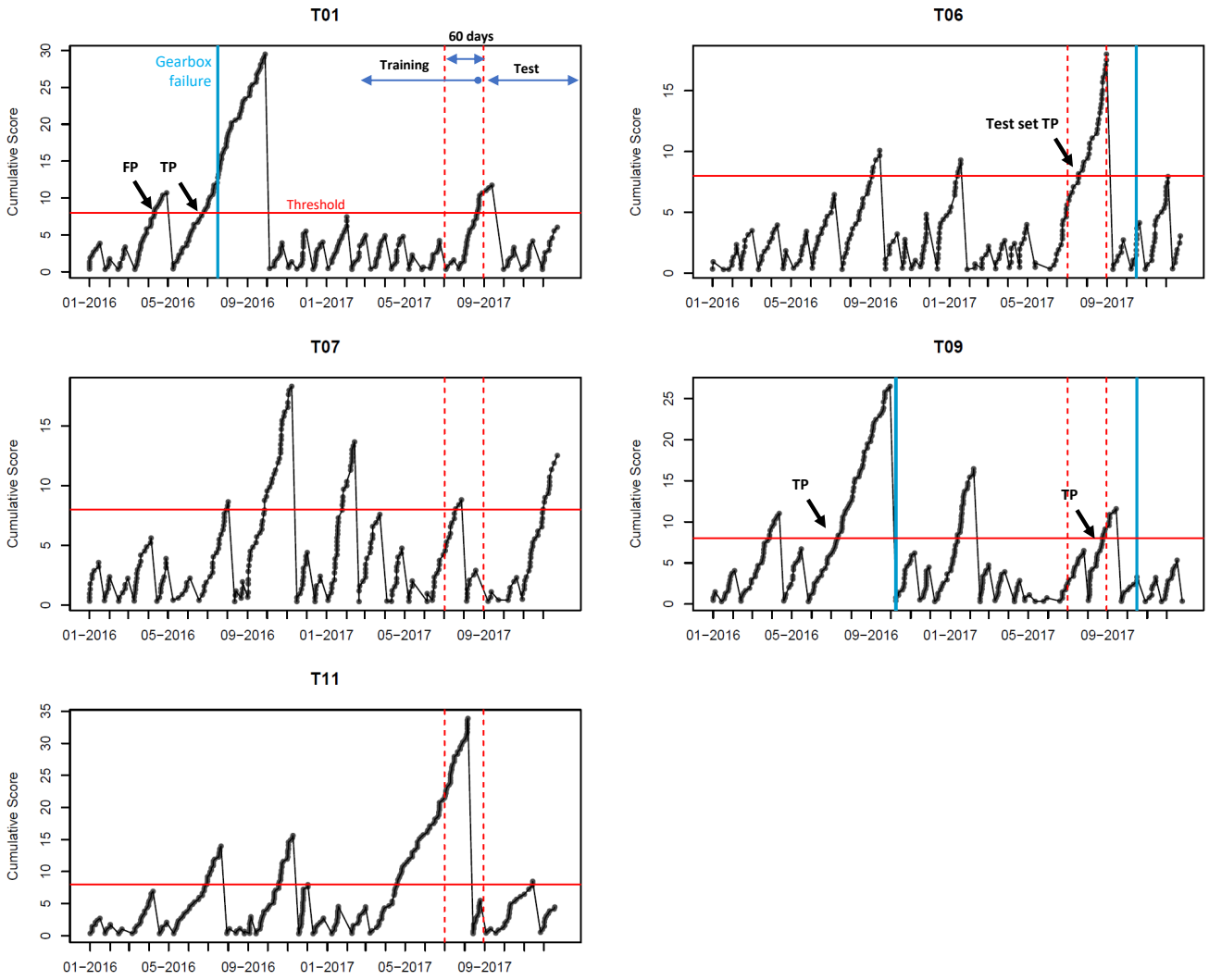

Figure 4. Gearbox failure detection results using the proposed method. These results are obtained by setting the control limit as $H = 8$, which is common for all turbines.

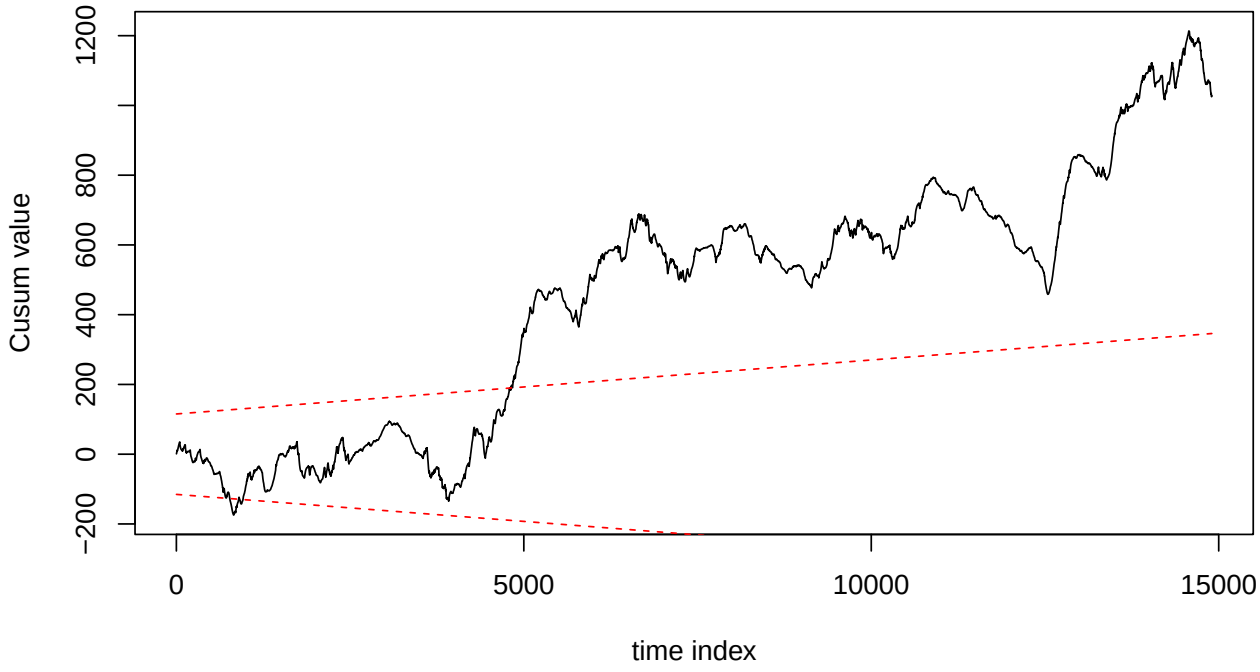

Figure 5. An example of CUSUM plot based on the method in (Dao, 2021). The time axis is in the unit of 10 minutes, as the method in (Dao, 2021) uses 10-minute data. This plot covers the first quarter (three months) of the data. In this particular set of data, the plot goes outside the control limit twice; once went out of the lower limit and once of the upper limit. After going up beyond the upper limit at around $5,000^{th}$ data point, the plot almost consistently stays above the line.