Learning Decision Rules by Particle Swarm Optimization (PSO) for Wind Turbine Fault Diagnosis

Xiang Ye [1], Yanjun Yan [2], and Lisa Ann Osadciw [3]

[1] *Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244, USA*
xye@syr.edu

[2] *Arcon Corporation, Waltham, MA, 02451, USA*
yanjun@arcon.com

[3] *Sensis Corporation, East Syracuse, NY, 13057, USA*
Lisa.Osadciw@sensis.com

ABSTRACT

The next generation wind turbine systems become more and more complex, which requires a more accurate fault detection method to ensure their efficiency. On a wind farm, sibling turbines should see similar wind speed if they both work normally. Based on this, we design wind speed difference tests to detect both hard failures and soft failures, including anemometer faults. In such tests, it is crucial to determine the decision boundary optimally to tell apart the abnormal state from the normal state. We propose a Particle Swarm Optimization (PSO) based approach to learn from historical data to decide the location and size of the boundary. This procedure is adaptable to each turbine using SCADA (Supervisory Control And Data Acquisition) data only. Our approach is advantageous in its applicability and data-driven nature to monitor a large wind farm. The test result has verified the effectiveness of our approach, and we have observed the anemometer aging in data.

1. INTRODUCTION

In a wind turbine energy generation system, diagnosis of potential faults is crucial to maintain and improve the efficiency of the system (Ribrant & Bertling, 2007; Chen & Blaabjerg, 2006). Currently there are built-in diagnostic units developed for physical model based faults such as bearing faults (Stefani, Bellini, & Filippetti, 2009), and there are some effective features to diagnose specific faults, such as wavelet transformation for spectrum decomposition in mass unbalance fault diagnosis (Yang, Tavner, Crabtree, & Wilkinson, 2010), but other minor yet also detrimental faults, such as anemometer and wind vane faults, are not fully studied (Yan, Osadciw, Benson, & White, 2009; Lu, Li, Wu, & Yang, 2009). We propose to use data-driven approaches to learn the normal and abnormal patterns from the available data, which help both sensor validation and diagnostics in an integrated way. The abnormalities are detected by tracking the turbine state variations (Zaher & McArthur, 2007). If a fault could be detected at an early stage, the potential damage could be minimized or mitigated through early repair avoiding drastic breakdown of the wind turbine, which leads to less downtime and more revenue; meanwhile, the repairs can be optimally scheduled with the regular maintenance trips to minimize the repair costs (Chen, Lian, Yu, & Bao, 2009).

Anemometer measures the wind speed as seen by the turbine. The wind speed measurement is used for configurations and control settings of the turbine, and hence very important for online monitoring of turbine performance (Burton, Sharpe, Jenkins, & Bossanyi, 2001). It is crucial to monitor the anemometer status and detect its failure.

It is observed that on a wind farm there are turbine groups that see similar wind flow, either due to their physical proximity, or their similar configuration in a cluster of turbines including surrounding geography (Yan, Kamath, et al., 2009). These turbines are called sibling turbines. It is assumed that if the sibling turbines all perform properly, then they should measure similar wind speed. Based on this similarity, we begin to compare wind measurements between turbine pairs. If they begin to differ, this is a good indication that one of the wind turbines requires maintenance. We collect a week's worth of data and model the wind speed difference using the Weibull distribution, as the Weibull distribution is very practical for reliability test (Hisada & Arizino, 2002; Yeh & Wang, 2008). The detector uses the estimated Weibull parameters to define the normal and abnormal states of wind turbines. The abnormal wind speed difference patterns are caused either by a faulty anemometer directly or by other faults indirectly. For instance, if a system is shut down due to major component failure, or if the system is under the influences of lightning, the anemometer of that particular turbine will produce very small readings causing detectable discrepancies in the wind speed measurements of the turbine pairs. We have designed multiple tests on other sensor data to exclude faults not caused by anemometers.

In order to determine the decision boundary between the normal and abnormal states in the wind speed difference failure detector objectively and optimally, we propose to use the particle swarm optimization (PSO) algorithm to learn from the historical data. Model parameterization is critical for accurate diagnosis (Bennouna, Heraud, Chafouk, & Notton, 2009), and PSO is an effective optimization algorithm to automate the parameter estimation. PSO is inspired by social behaviors, where a population of particles search through the solution space to find the global optimum of a fitness function defined for each specific application. Each particle represents a complete solution and moves in the search space using both cognitive awareness and social influence. The PSO algorithm has been widely applied in NP-hard optimization problems.

The optimal decision boundary between normal and abnormal states is designed as the boundary to minimize the errors due to missed detection and false alarms on the training data. We also compare the PSO algorithm with differential evolution (DE) and evolutionary strategy (ES) algorithms. It turns out that the PSO algorithm achieves the lowest fitness value in a shorter converging time than the other two algorithms. The same decision rule is then tested on an exclusive testing data set for the same turbine. We verify our test results by the automatic failure flagging on the turbines and the monthly operational reports from the wind farm operators.

The rest of the paper is organized as follows. We describe the wind turbine failure detection based on wind speed differences between two neighboring turbines in Section 2. Then we introduce PSO and explain how to apply it to determine the decision boundary in the wind speed difference test in Section 3. We evaluate the performance of our method in Section 4. Finally, we provide concluding remarks in Section 5.

2. WIND TURBINE FAILURE DETECTION BY WIND SPEED DIFFERENCE TEST

A wind farm consists of multiple turbines, even hundreds in some cases, and the coverage of the farm can be on a complicated landscape with obstructions due to local topographic features (Tindal et al., 2008). This implies that each turbine may have its own operational characteristics, which complicates the fault detection (Ribrant, 2006). We propose to use data-driven approaches, which adapt to each turbine automatically. The turbine performance data is collected by the SCADA (Supervisory Control And Data Acquisition) system. The data are collected from dozens of built-in sensors on the turbines measuring various physical quantities.

Some failures such as the bearing and gearbox failures (Robb, 20045) cause the turbines to completely shut down, which is obviously detrimental to the turbine's performance, and, therefore, have been studied thoroughly by many manufactures. Their diagnostic units are built into the wind turbines. These failures are called hard failures. On the other hand, many soft failures, which degrade the turbine's performance but do not necessarily stop the turbine from running, are often overlooked, such as anemometer faults. This kind of soft failure can be also very harmful to the efficiency of the turbines in the same way long term. This paper provides one technique that can recoup some of the losses by catching the soft failures early on and then supporting maintenance plans to minimize their impact.

Wind turbines have different operating modes that match each possible wind speed. The turbine starts generating electricity when the wind speed exceeds a lower-bound threshold, such as 5 m/s. Then the turbine increases its rotation speed as it reaches its maximum power production at a wind speed of approximately $15 - 18$ m/s. If the wind speed exceeds an upper-bound threshold, such as 20 m/s the wind turbine is stalled or braked to prevent damage or an accident. The wind speed measurement is an important control parameter for real time operation, and hence the wind speed measurement from the anemometer is highly correlated with rotor speed, pitch angle, exported power, etc. A faulty anemometer reading can cause severe damage, because if the turbine does not shut down at the right time, the turbine may not keep up and over heat or a blade may come loose. Further, incorrect readings may also result in false alarms that the wind speed is higher than it really is and frequent braking causing unnecessary downtime.

One solution is to pair up all the turbines so that each turbine has a sibling. Two sibling turbines would most likely measure a similar wind speed at the same time. If the measured wind speed is drastically different, it could be due to anemometer faults, turbine malfunctioning or wind blockage. We (Ye, Veeramachaneni, Yan, & Osadciw, 2009) proposed to use wind speed difference between a turbine pair to detect faults and fuse the results from multiple pairs to identify the turbine at fault. Due to the high variation of wind speed, a direct point-to-point comparison of wind speed differences easily triggers false alarms. Therefore, a week's worth of wind speed data is aggregated to increase the robustness of the detection.

2.1 Two-Dimensional Failure Detector

We choose a turbine, denoted as A, and its closest neighboring turbine, B, to evaluate the difference in their wind speed measurements. These two turbines are closest to each other, and there are no other turbines around them to interfere the air flow around them. They make a sibling turbine pair. We divide the data set into individual weeks that contain n data points, or less if data is missing. For each week, we calculate the absolute values of wind speed difference between A and B as

$$wsd = |ws_A - ws_B|. \tag{1}$$

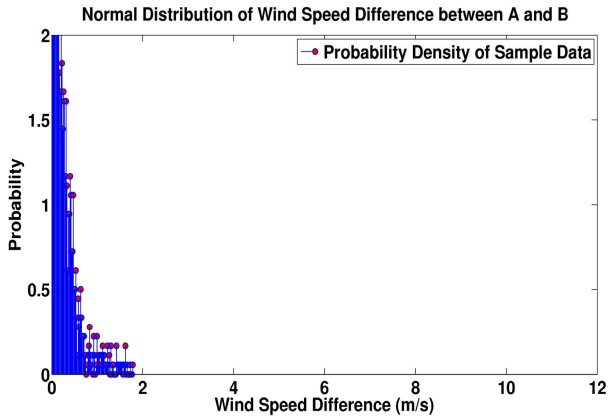
Then the weekly data are estimated by a two-parameter Weibull distribution of

$$pdf(wsd; \lambda, k) = \frac{k}{\lambda}(\frac{wsd}{\lambda})^{k-1}e^{-(wsd/\lambda)^k}, \tag{2}$$

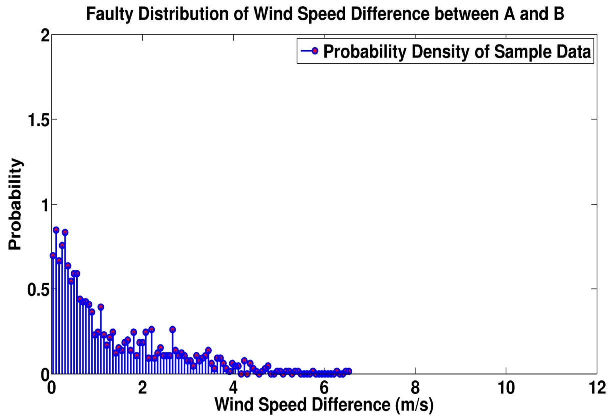
where k and λ are the estimated shape parameter and scale parameter, respectively.

Since both turbines ideally should see similar wind speeds, the Weibull probability density distribution should match and hence produce little spread. Figure 1 (1), (2) and (3) demonstrate the "*normal*", the "*faulty*" and the "*idle*" states of Weibull distribution of wind speed difference, respectively.

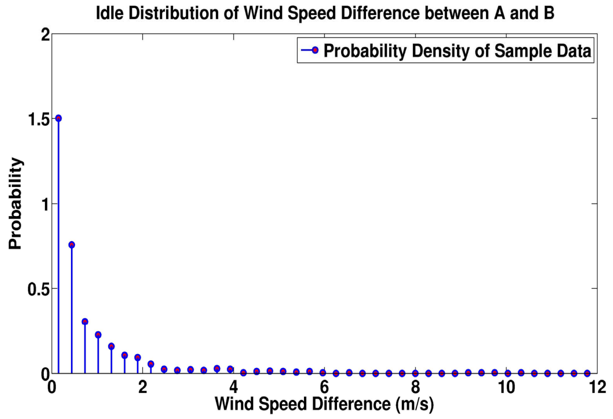
The *normal* state such as in week 31 in Figure 2.1 has a sharp distribution. The wind speed difference in this week is all less than 2 m/s, and the wind speed difference is consistently small. This implies that both turbines work similarly, and the probability that they both work well is high.



(1) normal in week 31



(2) faulty in week 79



(3) idle in week 118

Figure 1: Examples of normal, faulty, and idle distributions of wind speed difference between turbine A and B.

The *faulty* state such as in week 79 in Figure 2.1 has a more spread distribution between 0 m/s and 6 m/s with a larger scale parameter, and its shape parameter is greater than 1. This distribution is indicative of a soft failure in one of the turbine anemometers, mostly due to aging. The faulty turbine's performance degrades gradually.

The *idle* distribution in Figure 2.1, as an example of week 118, is flattened out with a long tail. This kind of Weibull distribution has a large scale parameter and a shape parameter less than 1. An intuitive explanation of such a scenario is that one turbine completely shuts down because of hard failures such as lightning or major component failure, and its anemometer is also turned off to read near-zero wind speed. As a result, the wind speed difference distribution is almost the same as the wind speed distribution of the other normal turbine.

We wish to sum up saying that the turbine working status is reflected by the estimated Weibull distribution of the weekly wind speed difference data, represented by the scale and shape parameters. Based on this phenomenon, the main goal is, in the 2-dimensional plot of Weibull scale and shape parameters, to reveal the normal and abnormal regions of turbine performance. Improved from our previous work (Ye et al., 2009), in this paper we propose a new method to determine a circle-shaped decision boundary between the normal and abnormal states using PSO algorithm. The weeks inside the circle are decided to be *normal*, when both turbines are functioning properly. The ones outside the circle are decided *faulty* or *idle*, where the *faulty* state often indicates soft failures, and the *idle* state often indicates hard failures. Compared to differential evolution (DE) and evolution strategy (ES) algorithms, PSO achieves better solution with lower fitness value in a shorter convergence time.

2.2 One-Dimensional Failure Detector

As a way to measure the integrated effects of the scale and shape parameters, we designed another test based on the area under the Weibull cumulative distribution function (CDF). The area definition is given later in equation (4). The Weibull CDF function is defined as

$$\mathrm{cdf}(wsd; \lambda, k) = 1 - e^{-(wsd/\lambda)^k}. \tag{3}$$

If both turbines work well and similarly, the cumulative probability function rises to value 1 fast. Otherwise, if one turbine is faulty, then the wind speed difference is larger, and the Weibull cumulative distribution approaches 1 slower. For instance, Figure 2 illustrates that the Weibull cumulative distribution of wind speed difference in week 31 is steeper than those in week 79 and week 118, and week 31 is a *normal* week when both turbines work well.

As shown in Figure 2, when the Weibull cumulative distribution rises to 1 fast, the curve covers more area under it. So we consider the normalized total area under the CDF curve (AUC) as an indicator of such a phenomenon,

$$\mathrm{AUC}(wsd = wsd_{max}; \lambda, k) = \frac{\int_0^{wsd_{max}} \mathrm{cdf}(w; \lambda, k) dw}{\int_0^{wsd_{max}} 1 dw}, \tag{4}$$

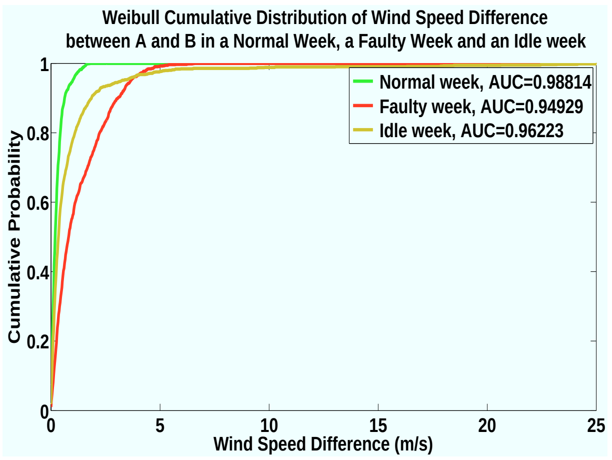


Figure 2: Weibull Cumulative Distribution of Wind Speed Difference between A and B in Week 31, 79 and 118

where the infimum of the area is

$$\mathrm{AUC}(wsd = wsd_{max}; \lambda, k) \rightarrow \frac{\int_0^{wsd_{max}} 1dw}{\int_0^{wsd_{max}} 1dw} = 1, \tag{5}$$

and in our application, we have $wsd_{max} = 25$ here. The bigger the AUC, the better both turbines work. Figure 3 plots the AUC curves versus week numbers. An area of $\int_0^{25} 1 - e^{-(wsd/0.9)^{0.9}} dw = 0.96382$, shown as the green straight line, serves as a reference when the scale and shape parameters are both 0.9. The line is used as the threshold to declare problematic weeks for assigning the weekly health flags later on.

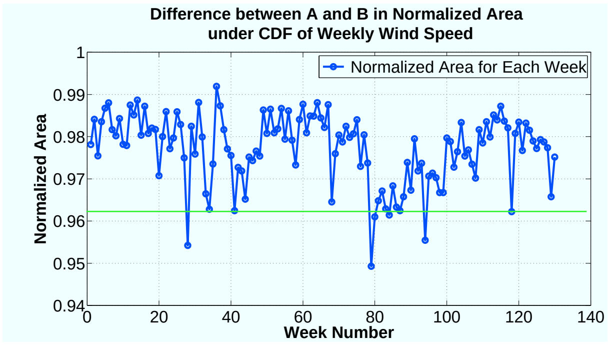


Figure 3: Normalized Area under Weibull Cumulative Distribution of Weekly Wind Speed Difference, AUC, between Turbine A and B

2.3 Relation between 2D and 1D Failure Detectors

In this paper, the 2D method uses 1D method to help training the decision boundary, so 2D method is enhanced by 1D method. On the other hand, if only using 1D method, we can't tell apart the hard failures and soft failures. These two methods complement each other, and we do not compare them.

3. PARTICLE SWARM OPTIMIZATION BASED DECISION BOUNDARY DETERMINATION

3.1 Particle Swarm Optimization Algorithm

Inspired by the fact that simple behaviors of individuals lead to much complicated societal phenomenon, Kennedy and Eberhart first proposed the particle swarm optimization (PSO) algorithm in 1995 (Kennedy & Eberhart, 1995). Particles in PSO are randomly deployed to effectively explore the solution space, while the fitness function (the notation of "fitness" function is by convention, more accurately, it should be called an objective function without the implication that the higher the value, the more "fit") guides the particles to exploit the promising regions with randomness. PSO has been widely applied in various optimization problems, especially the NP-hard problems, when other optimization algorithms do not work. We introduce the generic PSO algorithm in this section, and then we adapt it for the wind speed difference test problem in the next section.

Without losing generalizability, assume that the optimization is a minimization problem (negating a maximization yields minimization), and the function $f(\mathbf{x})$ of multivariate $\mathbf{x}$ is to be optimized,

$$\mathbf{x} = \arg minf(\mathbf{x}). \tag{6}$$

Then the particle in PSO is $\mathbf{x}$, and the function value $f(\mathbf{x})$ is the fitness of the solution $\mathbf{x}$. Suppose that there are N particles in the swarm with the fitness of each particle as $Fitness_i$ ($i \in [1, N]$). The particle's movement is affected by its inertia, its cognitive awareness (p_{best}, the best location that the particle has been before) and social influence (g_{best}, the best location within the population during the iterations). The algorithm of PSO is as follows.

1. Initialize a population of N particles. Each particle is a solution, $\mathbf{x}_i$, $i \in [1, N]$. $Fitness_i$, p_{best_i}, and g_{best}, are all initialized to be infinity.
2. Until the iteration index t reaches the maximum iterations t_{max}, or, some other termination condition is satisfied, do the following.
   (a) Evaluate $Fitness_i = f(\mathbf{x}_i)$.
   (b) Update $p_{best_{i,t}}$ for each particle, as cognitive awareness.
   (c) Update g_{best_t} for the population, as social influence.
   (d) Move the particles by

$$\mathbf{x}_{i,t+1} = \mathbf{x}_{i,t} + \mathbf{u}_{i,t+1}, \tag{7}$$

where $\mathbf{u}_{i,t+1}$ is the influence defined by

$$\begin{aligned}\mathbf{u}_{i,t+1} =& \omega \cdot \mathbf{u}_{i,t} + \\ & c_1 \cdot r_1 \cdot (p_{best_{i,t}} - \mathbf{x}_{i,t}) + \\ & c_2 \cdot r_2 \cdot (g_{best_t} - \mathbf{x}_{i,t}),\end{aligned} \tag{8}$$

where $\mathbf{u}_{i,t}$ is the velocity of the ith particle at time instance t, $\mathbf{u}_{i,t+1}$ is the velocity of the next step, and ω is an inertial constant, less than 1, to retain the information of previous velocity. $\mathbf{x}_{i,t}$ is the current particle's location that needs to be updated. c_1 and c_2 are constants to weigh these influences. r_1 and r_2 are uniform random numbers to randomize the influences.

(e) $t = t + 1$.

3. Finish. Print out the best solution.

3.2 Application of PSO to Determine the Decision Boundary

A major problem in the wind speed difference test is to determine the decision boundary between the normal and abnormal states in the feature space objectively and optimally. The feature space spanned by the shape and scale parameters of the estimated Weibull distribution of the weekly wind speed differences is illustrated in Figure 4.

When soft failures such as the anemometer faults happen, the wind speed difference is not drastic, but the difference is consistently there, and hence the distribution is skewed towards the bigger wind speed difference. In this case, both the shape parameter and the scale parameter are large numbers, pointing to the upper right region.

When hard failures such as major component failures happen, one turbine completely shuts down and its anemometer reads zero wind speed, causing maximum wind speed difference between a turbine pair. The distribution is thus nearly flattened out with a very heavy tail, represented by a bigger scale parameter but small shape parameter, namely, the lower right region.

Based on the above analysis, we note that the features of the normal weeks are clustered together, and yet the features of the abnormal weeks can spread around anywhere outside of that cluster. We propose a heuristic that the features of the normal weeks are clustered in a circular region, which is the least informative shape, yet the location and size of such a circle in the feature space is turbine specific. In order to determine the origin and the radius of the circle, we utilize PSO algorithm to learn from historical data. Then we use this decision rule to test on future data.

The particle in our application is defined as

$$\{P_i : x_i, y_i, r_i\}, \tag{9}$$

where the particle index is i going from 1 to N, the origin of its circle is (x_i, y_i), and the radius of its circle is r_i.

Each SCADA data record is associated with a manual or automatic label indicating the event happening at that time, which is used to label whether the turbine works normally at that instance. If the labels associated with faults account for more than 20% percentage of the weekly data, this week is regarded as a problematic week, and hence a health flag of -1 is applied to this week. Otherwise this week is assigned with a health flag of 1. An exception is that if there are too many missing data and hence the data are not sufficient, a health flag of 0 is assigned, which does not affect the optimization procedure.

However, the SCADA flagging is not completely dependable, especially on the degradation of the faulty components. This is also why we need to design better diagnostics and prognostics algorithms. Another factor to label whether the week is healthy or problematic is the area value under the Weibull CDF curve of the wind speed difference in that week, as defined in equation (4). A health flag of -1 overrides a flag of 1 if the area value is less than 0.96382.

Note that labelling is based on both the reported events and AUC, where AUC serves as a data-based evaluation to enhance the reliability of event flagging for labelling, which is for training purpose. In testing, we often do not have enough resources to label, and then testing is based on classification.

After the health flag of each week is assigned, any decision boundary may incur two types of errors. One error is miss detection, defined as the number of weeks with a health flag of -1 but located inside the circle as being normal,

$$(E_{\text{miss detection}})_i = \sum_j I(\text{normal}|\text{health flag}_j = -1), \tag{10}$$

where j is the set of week indices in the training set. $I(\cdot)$ is a counting function, with value 1 when the argument is true, or else 0. The other error is false alarm, defined as the number of weeks classified as abnormal outside the circle but with a health flag of 1,

$$(E_{\text{false alarm}})_i = \sum_j I(\text{abnormal}|\text{health flag}_j = 1). \tag{11}$$

The fitness function is the summation of the above two errors associated with each particle,

$$Fitness_i = f(P_i) = (E_{\text{miss detection}})_i + (E_{\text{false alarm}})_i. \tag{12}$$

The optimal solution minimizes the fitness function.

4. SYSTEM SIMULATION AND RESULTS

We have 130 weeks' worth of data, and we split them into training and testing sets. The training set includes the first 70 weeks, and each week is labelled with 1, -1 or 0 indicating whether that week is healthy, problematic or missing-too-much-data. In Figure 4, the green points represent healthy weeks, and the red ones problematic weeks. If one green point is outside the candidate decision boundary, it causes false alarm error; however, if one red point is inside the boundary, it causes miss detection error. We use 200 particles and let them search in 100 iterations. Note that the number of particles could be reduced to maintain a similar performance, and our analysis indicates that more particles do not necessary improve performance, and hence we use 200 particles. We have tried different total number of iterations. PSO often converges in less than 50 iterations, and we choose 100 iterations to allow some tolerance.

In each iteration, the fitness of each particle is evaluated by equation (12). The best solution seen by each particle is used to update p_{best_i}, and the best solution seen by the whole population is used to update g_{best}. Through iterations, all particles move towards to the optimal locations driven by their own cognitive awareness and social influence.

Figure 4 shows the 2-dimensional plot of Weibull scale and shape parameters of wind speed difference between turbine A and B. Each point represent one week. We use the first 70 weeks as training data, and then apply PSO to determine the optimal decision boundary with the lowest error amount. In Figure 4, the red circle represents the optimal decision boundary obtained by PSO algorithm from the training data. This circle separates the normal and abnormal regions of the wind speed difference test incurring minimum errors. In this feature space, the weeks associated with soft failures are located to the upper right of the circle, and the weeks associated

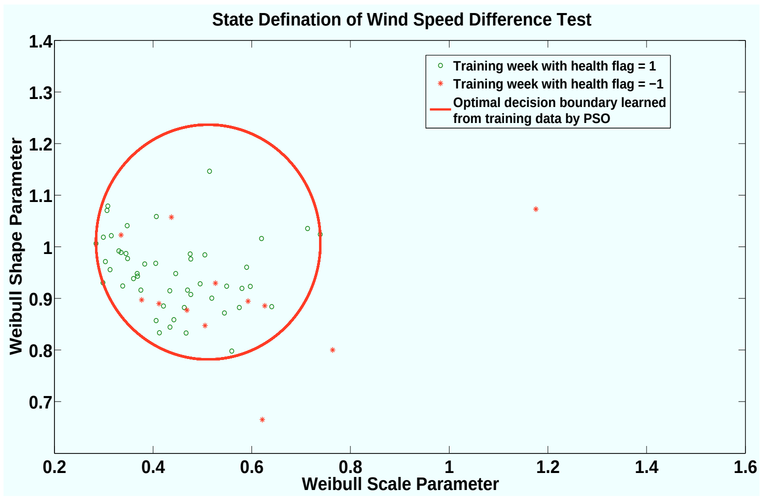


Figure 4: 2D plot of weekly estimated Weibull parameters of wind speed difference between turbine A and B. Apply the first 70 weeks as training data. Learn the optimal decision boundary of normal and abnormal states by PSO.

with hard failures are located to the lower right of the circle.

We compare PSO algorithm with the other two optimization algorithms, Differential Evolution (DE) (Gao & Tong, 2006) and Evolution Strategy (ES) (Hansen, March 7, 2010). Differential Evolution algorithm is an optimization method in evolutionary algorithms (EAs), capable of handling non-differentiable, nonlinear and multi-model objective problems. The crucial idea behind DE is a scheme for generating new candidate solutions by combining existing ones according to its formulae of vector crossover and mutation. Then, DE adds the weighted difference between two solution vectors to a third one, which makes the scheme completely self-organizing. In each iteration, DE updates the solution which has the best fitness value on the optimization problem. On the other hand, Evolution Strategy (ES) algorithm is a stochastic, derivative-free numerical optimization method for nonlinear or non-convex problems. In ES algorithm, new candidate solutions are sampled according to a multivariate normal distribution. The pair-wise dependencies between the variables in this distribution are described by a covariance matrix, which is updated by the covariance matrix adaptation (CMA) method. So, this algorithm is also called CMA-ES.

Figure 5 compares the progression of g_{best} versus iterations among three mentioned algorithms. As shown in Figure 5, the PSO algorithm achieves the lowest fitness value in a shorter converging time than both DE and CMA-ES. The fitness value by PSO converges below 10 around the 10th iteration, but the other two algorithms need about 20 iterations. Eventually, all methods hit the lowest fitness value after 40th iteration.

After the location and size of the decision boundary circle are determined for each turbine, we use this decision rule to test on future data for the same turbine. We use the data from week 71 to week 130 in testing. In Figure 6, the red circle is the optimal decision boundary learned from the first 70 weeks. All the weeks outside this circle are classified as abnormal weeks. Except for week 118, which falls into the *idle* state due to the lightning strike, all the other abnormal weeks are caused by the fact that the anemometer is degraded as time goes on. As shown in Figure 6, the anemometer barely functions since week 79 until it is replaced in week 95. With the red decision boundary learned from the earlier data, even through there are false alarms, miss detections are avoided.

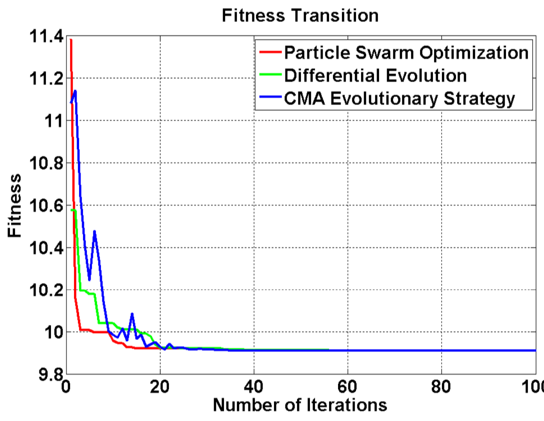


Figure 5: Comparison of fitness transition among PSO, DE and CMA-ES

To demonstrate the system degradation due to anemometer aging, we apply the same decision boundary procedure on the testing data from week 71 to week 130 to obtain the yellow circle. This procedure on the latter data is not used to train or test the system, but only to show the average system performance. The center of later data set is shifted to the right, indicating that the anemometer shifts towards worse performance.

5. CONCLUSIONS

We propose a PSO based approach to determine the decision boundary between the normal and abnormal states in the wind speed difference failure detector. The training data are labelled based on known events from wind

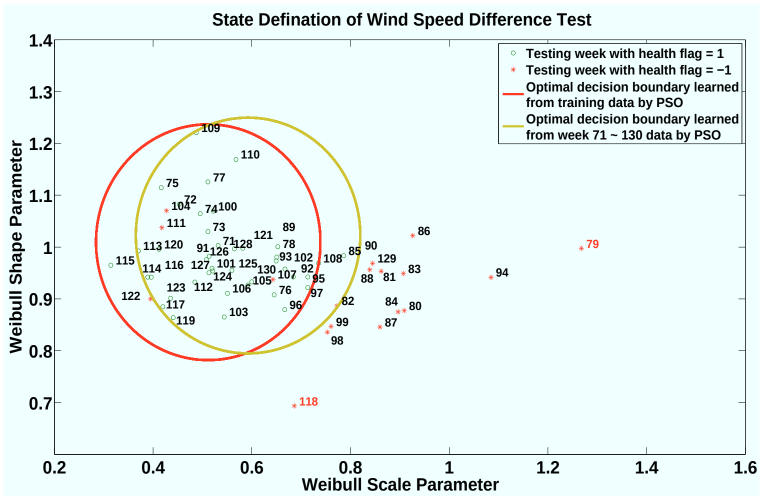


Figure 6: 2D plot of weekly estimated Weibull parameters with optimal decision boundary. Apply the week 71 to week 130 as testing data. The average system performance gets worse due to the aging anemometer.

farm operators and the fault detectors designed by us. The fitness function is the summation of miss detections and false alarms, which is minimized to yield the optimal location and size of the boundary circle. The testing result verifies the effectiveness of fault detection with minimal false alarms. We also observed anemometers aging as the circles shifts toward the right if the circle is obtained using latter data.

ACKNOWLEDGMENTS

We thank AWS-Truewind for their technical and financial support to this research.

REFERENCES

Bennouna, O., Heraud, N., Chafouk, H., & Notton, G. (2009, jul.). Influence of model parameters on the diagnosis of the wind turbine generator. In (p. 1 -5).

Burton, T., Sharpe, D., Jenkins, N., & Bossanyi, E. (2001). *Wind Energy Handbook*. Wiley.

Chen, Z., & Blaabjerg, F. (2006). Wind Energythe worlds fastest growing energy source. *IEEE Power Electron Soc Newslett*, *18*(3).

Chen, Z., Lian, X., Yu, H., & Bao, Z. (2009, nov.). Algorithm of Data Mining and its Application in Fault Diagnosis for Wind Turbine. In (Vol. 2, p. 240 -243).

Gao, F., & Tong, H. (2006). Differential Evolution: An Efficient Method in Optimal PID Tuning and on–line Tuning. In *Proceedings of the First International Conference on Complex Systems and Applications.* Wuxi, China.

Hansen, N. (March 7, 2010). *The CMA Evolution Strategy: A Tutorial* (Tech. Rep.). Available from `http://www.lri.fr/~hansen/cmatutorial.pdf`

Hisada, K., & Arizino, F. (2002, sep.). Reliability tests for Weibull distribution with varying shape-parameter, based on complete data. *Reliability, IEEE Transactions on*, *51*(3), 331 - 336.

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. In *Proc. IEEE Int'l. Conf. on Neural Networks (Perth, Australia)* (Vol. IV, p. 1942-1948). IEEE Service Center, Piscataway, NJ.

Lu, B., Li, Y., Wu, X., & Yang, Z. (2009, jun.). A review of recent advances in wind turbine condition monitoring and fault diagnosis. In (p. 1 -7).

Ribrant, J. (2006). *Reliability performance and maintenance - a survey of failures in wind power systems.* Unpublished doctoral dissertation, XR-EE-EEK.

Ribrant, J., & Bertling, L. M. (2007, mar.). Survey of Failures in Wind Power Systems With Focus on Swedish Wind Power Plants During 1997 ndash;2005. *Energy Conversion, IEEE Transactions on*, *22*(1), 167 -173.

Robb, D. (20045). Gearbox design for wind turbines improving but still face challenges. *Windstat Newsletter*, *18*(3).

Stefani, A., Bellini, A., & Filippetti, F. (2009, nov.). Diagnosis of Induction Machines' Rotor Faults in Time-Varying Conditions. *Industrial Electronics, IEEE Transactions on*, *56*(11), 4548 -4556.

Tindal, A., Johnson, C., LeBlanc, M., Harman, K., Rareshide, E., & Graves, A. (2008). Site-Specific Adjustments to Wind Turbine Power Curves. In *AWEA WINDPOWER Conference.* Houston, TX, USA.

Yan, Y., Kamath, G., Osadciw, L. A., Benson, G., Legac, P., Johnson, P., et al. (2009, July). Fusion for Modeling Wake Effects on Wind Turbines. In *Proceedings of 12th International Conference on Information Fusion.* Seattle,Washington, USA.

Yan, Y., Osadciw, L. A., Benson, G., & White, E. (2009, May). Inverse Data Transformation for Change Detection in Wind Turbine Diagnostics. In *Proceedings of 22nd IEEE Canadian Conference on Electrical and Computer Engineering.* Delta St. Johns, Newfoundland and Labrador, Canada.

Yang, W., Tavner, P., Crabtree, C., & Wilkinson, M. (2010, jan.). Cost-Effective Condition Monitoring for Wind Turbines. *Industrial Electronics, IEEE*

Transactions on, *57*(1), 263 -271.

Ye, X., Veeramachaneni, K., Yan, Y., & Osadciw, L. A. (2009, July). Unsupervised Learning and Fusion for Failure Detection in Wind Turbines. In *Proceedings of 12th International Conference on Information Fusion.* Seattle,Washington, USA.

Yeh, T.-H., & Wang, L. (2008, jun.). A Study on Generator Capacity for Wind Turbines Under Various Tower Heights and Rated Wind Speeds Using Weibull Distribution. *Energy Conversion, IEEE Transactions on*, *23*(2), 592 -602.

Zaher, A., & McArthur, S. (2007, jul.). A Multi-Agent Fault Detection System for Wind Turbine Defect Recognition and Diagnosis. In (p. 22 -27).

Xiang Ye is currently a Ph.D. candidate of Electrical and Computer Engineering at Syracuse University. She obtained her M.S. in Electrical Engineering from Syracuse University in 2009, and her M.S. in Engineering Technology from Pittsburg State University in 2006. Her research is focused on applying artificial intelligence and optimization algorithms for reasoning, planning, and learning with probabilistic models. Her current work is on the application of such algorithms to real-world industrial problems such as automatic schema matching, wind turbine diagnostics and prognostics, and air traffic control. Currently she is doing her internship at Mitsubishi Electric Research Laboratory. She is a member of IEEE and SPIE.

Yanjun Yan received her Ph.D. degree in Electrical Engineering in 2009 and M.S. degree in Applied Statistics in 2006, both from Syracuse University. She currently works at ARCON Corporation. She was at Osaka Kyoiku University (OKU) in Japan from 2000 to 2002 to work on a collaboration project between OKU and Harbin Institute of Technology, China. Her research interests include signal processing, pattern recognition, information fusion, optimization and their applications to real-world problems, such as face recognition, wind turbine diagnosis and prognosis, nonlinaer filtering and tracking, estimation, model order determination and differential geometry based nonlinearity evaluation. She is a member of IEEE, SPIE, and ISIF.

Lisa Osadciw earned her B.S. degree in Electrical Engineering from Iowa State University in 1986. She received her M.S.E.E. degree from Syracuse University in 1990 and her Ph.D. in Electrical Engineering from University of Rochester in 1998. She is currently a senior engineer at Sensis Corporation. She was a professor with the Department of Electrical Engineering at Syracuse University. She previously held a position as staff research engineer with Lockheed Martin in radar and sonar systems. Her research lies in the areas of sensor management, signal design (UWB and spread spectrum), optimizations, data fusion, communications and sensor networks. She is a senior member of IEEE.