FAST GENERATION OF MACHINE LEARNING MODELS IN MODEL-BASED FAULT DETECTION SYSTEMS

A Dissertation

by

GANG LI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Alexander G. Parlos
Co-Chair of Committee,	Reza Langari
Committee Members,	Won-Jong Kim
	Edgar Sanchez-Sinencio
Head of Department,	Andreas A. Polycarpou

August 2018

Major Subject: Mechanical Engineering

ABSTRACT

In model-based fault detection, processed input and output time-series data are used to generate models and then perform on-line predictions. Many practical considerations in model-based fault detection systems rule out most of the available approaches in generating on-line predictive models. One approach that satisfies the needs of on-line predictive model generation is the piece-wise linear robust regression (PWLRR) method. The PWLRR requires too much data for initial training, it has limited ability to perform accurate predictions beyond the initial training region (extrapolation), it does not perform accurate predictions without significant follow-on or on-line training, and it constantly runs into blind spots losing track of the asset condition. The performance of a fault detection system based on the PWLRR suffers in many instances, mostly in the form of delayed detection, missed faults and/or false alarms.

This work presents an alternative learning method to model generation that overcomes many of the existing disadvantages of the PWLRR approach. Gaussian process (GP) based regression, which exhibits good approximation properties with minimal training data points and has very good extrapolation properties, is selected as an alternate learning method in the fault detection system. The fraction of blind spots, number of data points in the training and validation sets, extent of extrapolation, average prediction error, true negatives (missed faults), false positives (false alarms), and detection time are all considered performance indicators when testing and comparing the model generation methods. Five (5) cases with artificial data and ten (10) cases with real world data from both staged experiments in the laboratory and fielded production sites are used to benchmark the performance of the GP approach, and compare it against the PWLRR approach.

Empirical research results comparing GP to PWLRR demonstrate that for comparable training levels, less data is required to train a GP model and with fewer blind spots. Extrapolation by the GP model improves significantly. Among the real test cases with faults, GP results in 100% detection rate, while PWLRR results in 50 % detection rate. For the test cases where both approaches result in true positives, detection time is improved by roughly 2.5 times when using the GP. In test cases without faults, GP results in no false positives, while PWLRR results in three (3) false positives. The proposed method for training based on GP can generate models with less computational resources than the PWLRR and requires less human intervention. Further testing is needed to verify the performance of the proposed approach on data sets with a wider variety of statistical properties.

DEDICATION

To My Wife

ACKNOWLEDGEMENTS

I would like to first and foremost thank to my committee chair, Dr. Alexander G. Parlos, for his guidance throughout the course of this research, this work would not have been possible without his support. Thanks to my committee co-chair Dr. Reza Langari, committee members, Dr. Won-Jong Kim, Dr. Edgar Sanchez-Sinencio for their helpful and invaluable comments. Also, thanks to Dr. Xingyong Song for his attendance and suggestions at my preliminary and final exam.

I would like to extend my gratitude and thanks to Dr. Amir Atiya and Mr. Jianxi Fu for sharing and discussing their opinions on the related topics. I would also like to take this opportunity to thank the faculty and staff at Office of Graduate and Professional Studies and Department of Mechanical Engineering for making my time at Texas A&M University a great experience.

Finally, I would like to thank my wife for her love, and my parents for their patience and encouragement.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a dissertation committee consisting of Dr. Alexander G. Parlos as chair, Dr. Reza Langari of the Department of Engineering Technology & Industrial Distribution as co-chair, Dr. Won-Jong Kim of the Department of Mechanical Engineering and Dr. Edgar Sanchez-Sinencio of the Department of Electrical Engineering as members.

All work conducted for the dissertation was completed by the student independently. The student is self-supported, there is no outside funding source.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

In modern industry, induction motors are rugged, low cost, low maintenance, reasonably small sized, reasonably high efficient, and operating with an easily available power supply, they have wide applications in electromechanical energy conversion. A complete system (asset) that converts the electrical power to mechanical power may include motors, shafts, couplings, belts, chains, gears, bearings, compressors, pumps, conveyors, etc. As time passes, the health of the asset degrades, some components fail resulting in unexpected shutdown. These shutdowns are usually very costly since the industry's production has heavy reliance on these assets [1].

This large cost associated with the unexpected shutdown can be avoided if degradation is detected in its early stage, therefore the proper maintenance can be scheduled accordingly to avoid catastrophic failure, or replacement parts and work can be prepared in advance to reduce the downtime [2].

Several methods have been developed to detect the asset's health degradation (fault detection). Such as estimation of motor parameters, or measurement of noise, temperature and vibration [3]. All these methods have their own limitations, therefore, a more cost-effective sensorless approach was pursued and developed previously through electrical signal analysis (ESA) [3].

However, although signal-based fault detection method is much simpler, lack of disturbance resistance would result in the generation of frequent false alarm and missing failure detection. So, a more complicated model-based fault detection method is required to achieve acceptable performance [4].

Due to the differences in assets' characteristics and the specific application environment, the "healthy" response to an asset is unique to each asset. So, cold start learning is essential, the models used to obtain the baseline must be generated from scratch not only once at commissioning by a predetermined period. To obtain the health baseline, the current model generation method requires an off-line optimization process and a significant window of training data, up to one year of training window of data.

Ideally, the data acquired should always be compared to the baseline model to detect fault. However, when there is no corresponding baseline for the data acquired, the inability to perform the detection at that time creates blind spots.

1.1 Motivation

The existence of long duration periods without prediction due to lack of training have motivated us to find model generation alternatives rather than the current approach. In particular, implementing a "one shot learning" approach to baseline model generation. The alternative model generation algorithm should require far fewer data to generate reliably than currently required to reduce the initial training. And further, it should generate a reliable model which is able to provide accurate predictions outside the narrow operating and power quality region found in the initial training to avoid the follow-on training. Accurate extrapolation beyond the input space of the training set becomes a key criterion. This is in the context of model generalization, which addresses the ability to obtain accurate predictions when interpolating and extrapolating from the initial training set.

In addition, any candidate solution must not suffer from the complexities that require the solution to nonlinear optimization problems or require lots of training data to obtain models. Such approaches are not amenable to fully autonomous model generation without any human intervention, which is a key requirement for the fault detection system to be applicable in the real industry in large scale.

1.2 Problem Definition

The use of the models is formulated in the solution of the fault detection problem. Then the specific learning problem to be solved is defined.

1.2.1 The Fault Detection Problem

The fault detection problem addressed consists of generating a static predictive model of the target asset as the baseline for use in detection. It is assumed that the inputs and outputs of the model are available from ESA. These measurements are used in generating the predictive model. Once the model is developed, the inputs are used to generate the predicted outputs from the baseline model. Then these predictions are then used in generating residuals (differences) from the corresponding output measurements. Finally, the residuals are then used in the detection and alarming logic to detect the onset of an incipient fault over time.

1.2.2 Model Generation Problem

Given the vector of input time series and a scalar output time series $(\vec{u}_k, y_k)$ for $k = 1, 2, 3, \dots N$, where k is the time index. A static model that relates all the inputs to the output is required to be generated such that once the model is ready, upon arrival of a new data set $(\vec{u}_p, y_p)$, the input vector $\vec{u}_p$ can be used with the model to obtain the scalar prediction $\hat{y}_p$. The residual is then defined as$e_p = y_p - \hat{y}_p$. As shown in Figure 1.

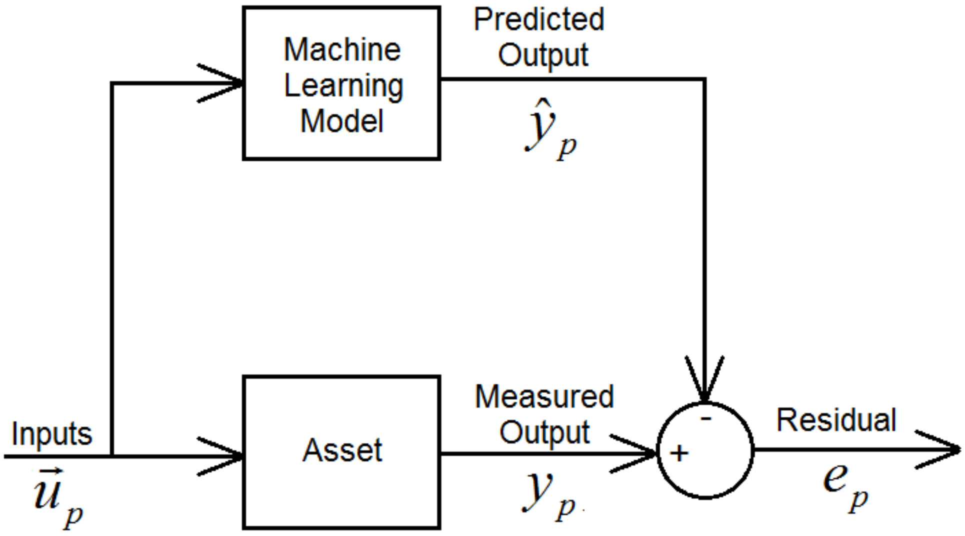

Figure 1 Model based fault detection

The method that generates the inputs and output time series $(\vec{u}, y)$ are developed previously by applying signal processing algorithms to the raw electrical measurements which continuously acquired by the data acquisition system. The details of these steps will not be investigated in this research.

1.3 Literature Review

1.3.1 Fault Detection by Electrical Signal Analysis (ESA)

The demand for predictive maintenance keeps increasing, this becomes the main motivation of the development of better fault diagnosis algorithms. Multiple methods with different measurements were developed and are kept being developed, including using acoustic signal [5], vibration measurements [6], electrical signature [7], etc. and even the combination of two or more measurements to achieve better detection and isolation performance as being shown in Figure 2 [8].

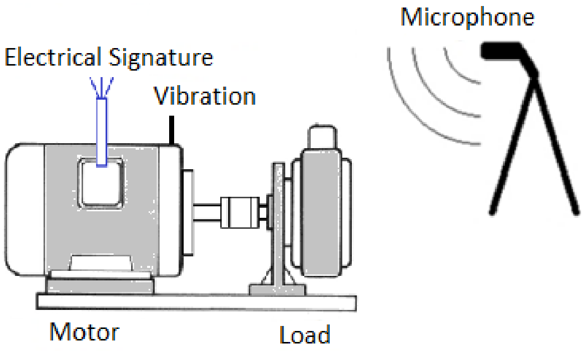

Figure 2 Setup of induction motor fault diagnosis with multiple measurement

Among all these methods, using the electrical waveform measurements were selected since the data acquisition for electrical measurements is non-intrusive and cost effective [1][4]. The three-phase voltage and current waveforms are sampled by a data acquisition endpoint and transmitted to a central server with more computing power, where the waveform signal get analyzed to produce input-output time series. Those time series are fed into the fault detection algorithm which includes solving the model generation problem as one of the steps.

1.3.2 Fault Types of an Asset

The complete system of an asset coverts the electrical power to mechanical power, therefore the fault of an asset can generally be separated into electrical fault and mechanical fault [9]. There are two major components of an asset, driver (induction motor) and driven load (pump, fan, compressor, etc.). A fault of the driver could happen on the stator, the rotor, the bearing or other places [10] [11], and depends the root cause of the fault, it could be classified as mechanical or electrical. A fault on the driven load has more varieties, since there exist tens of different types of load that can be driven by the induction motor, every each of them has different physical functionality and characteristics. But generally, fault on the driven load is always mechanical fault because the driven load does not contain any electrical components.

Type of Faults	Number of Faults	Mechanical Faults
Bearing	152	Yes
Winding	75	
Rotor	8	
Shaft	19	Yes
External Device	10	
Others	40	

Table 1 Statistics on induction motor faults/failures [1]

Both mechanical and electrical faults could be detected by ESA, but they require different signal processing method on the electrical waveform measurements, i.e. the model generation problem of the fault detection algorithm is solved with different input-output data for different types of fault. Table 1 [12]. provides a survey on induction motor failure cases, it shows just on the driver side, mechanical faults are more than electrical faults, in this research, only the input-output data for mechanical fault will be investigated, since the most of the asset failures are due to the mechanical fault, and previous work of the existed fault detection system is more developed and matured for the mechanical fault.

1.3.3 Gaussian Process (GP) For Machine Learning

There are many model generation methods available, such as statistical models (linear regression, robust regression, ridge regression, transformation methods, nonlinear regression, time-series prediction, non-parametric methods), econometric (System ID) models (ARX, ARMAX, NARX), machine learning models (SVM, kernel methods, Gaussian process).

The model based on GP prior and a kernel function can be used to fit nonlinear data with multidimensional inputs. It has been used as a flexible non-parametric approach for curve fitting, classification, clustering, and other statistical problems [13] [14]. And it has been widely applied to deal with complex nonlinear systems in many different areas particularly in machine learning [15][16].

The GP has several advantages, such as the prediction interpolates the observations, and different kernels even custom kernels can be specified. Figure 3 [17] shows samples drawn from GP with the same data and different kernel functions. Typical samples from the posterior of GP with different kernel functions have different characteristics. The periodic kernel function's primary characteristic is self-explanatory. The other kernel functions affect the smoothness of the samples in different ways.

Although GP require the whole samples information to perform the prediction and it is not very efficient in high dimensional spaces [18]. The current fault detection algorithm needs no more than five dimensions for the input space, and motivation of altering the model generation methods is to use significant less data to solve the model generation problem. Compared to other learning methods, some require more data to train therefore more training time and some may need some non-trivial hyper-parameters tuning which is not suitable for online automation, GP is selected to be investigated as the alternative model generation method in this research.

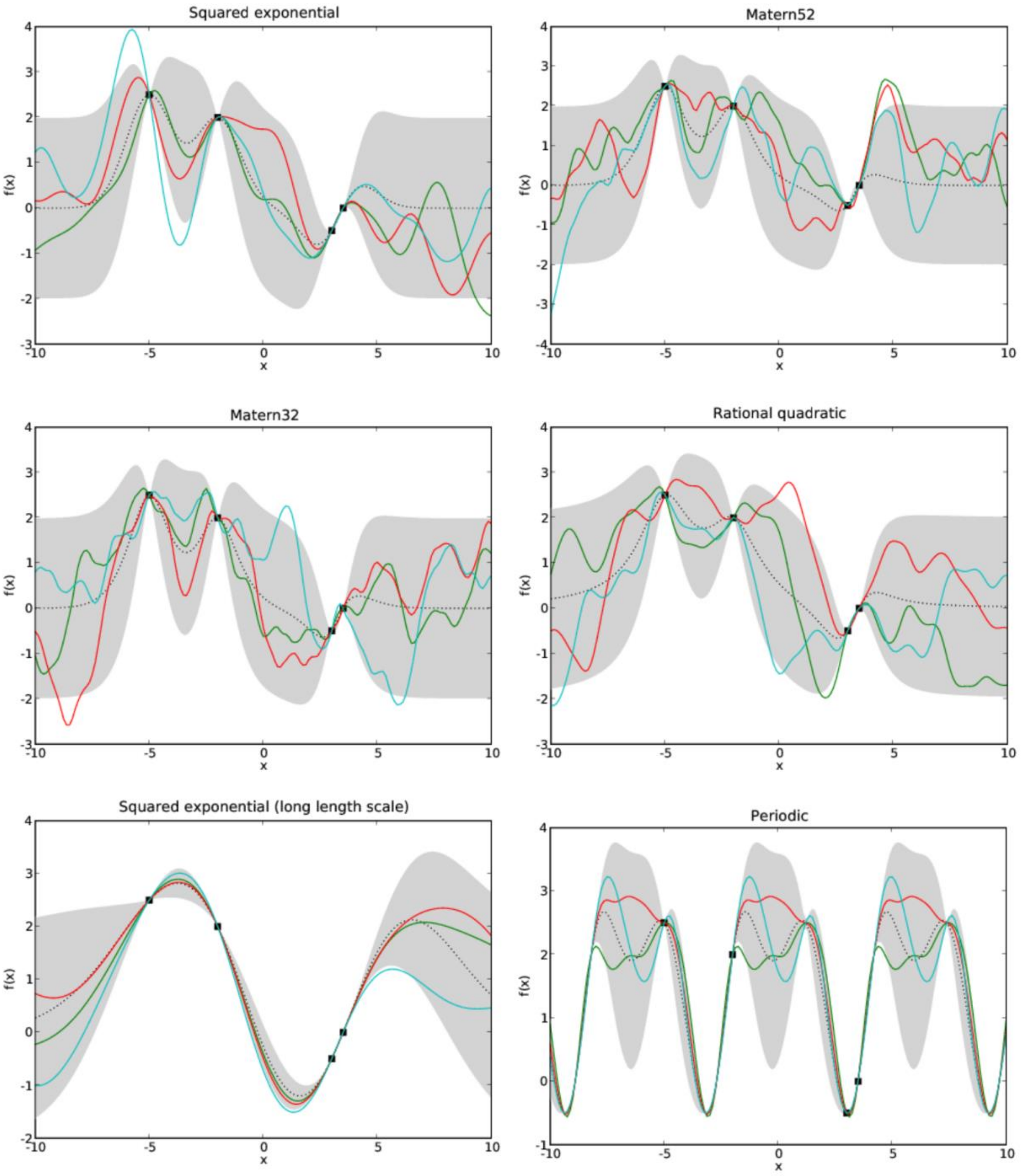

Figure 3 Different kernel functions for GP. Reprinted from [17]

1.4 Research Objectives

The objective of this work is presented as: generate an input-output model from the fewest on-line measurements such that the prediction errors, i.e. the residuals, on new unseen data are below a certain acceptable threshold. Given the vector of input time series and a scalar output time series generate a static model that relates all the inputs to the output with

- Fewest possible training points
- Not requiring the solution to nonlinear optimization
- Fully automated for structure and parameter selection
- Good extrapolation properties

1.5 Proposed Approach

Supervised-learning is a problem of learning input-output mappings from empirical data. If the output is discrete, the problem is known as classification. For our fault detection problem on asset, since the output from the processed electrical measurement is continuous, this learning problem is defined as regression problem. The regression problem has wide applications, such as in robot arm control, where the torque of the joint is continuous.

There are two common approaches to find the function that maps the input to the output. The first approach restricts the class of functions that are being considered. The disadvantage of this approach is if the target function is not well modeled by the restricted function selection, the prediction will be poor, and if the flexibility of the function selection is increased, it is possible to get into the overfitting situation in which the predicted function can perform good on the training data but poor on the testing data.

The second approach to find the mapping function is to give a prior probability to every possible function. Although this approach appears to be not feasible with finite computation since there are infinite possible functions, the GP can be used in this approach to avoid mathematical sophistication and only govern the properties of functions [19].

1.6 Research Contributions

The model generation by proposed GP method will be compared to the existing model learning method. The performance indicators which can characterize model learning method performance in this model-based fault detection system by ESA for induction motor driving asset will be defined. The performances of the proposed alternative method and existing method will be quantified and compared to each other.

If the attempt of using GP to get static model is successful, the time needed for the initial cold start data acquisition can be reduced, and the regions in the inputs space that are not covered by the initial data can be covered by the statistical interpolation and extrapolation with the initial data, Therefore, more prediction can be are performed. The overall performance of the fault detection system is also expected to be improved.

1.7 Organization of Dissertation

Section 2 will review the existing model learning method in the model-based fault detection system. The limitation and disadvantage of the existing model learning method will be further described.

Section 3 will interpret the algorithm of the proposed GP method and the procedure for applying GP on the model-based fault detection system. The overall definitions of the performance indicators for model learning method benchmark will also be introduced in this section.

Section 4 will list all detailed results from existing method and proposed method. The data for testing the model learning method are consisted by both artificial data and data collected from real world induction motor driving assets.

Section 5 will summarize the results, compare and conclude the performance of the two methods and also list the possible future work for the proposed GP method.

2. EXISTING METHOD FOR GENERATING MODELS

One of the few model building approaches that meets our criteria is the piece-wise linear robust regression (PWLRR) [20][21].

2.1 Overview of Piece-Wise Linear Robust Regression

The PWLRR is a model learning approach that restricts the class of functions that are being considered. It is selected to be the model learning method in the existing model-based fault detection system since it is an automated approach, it can handle nonlinearities and it has direct solution without requiring optimization.

The PWLRR method approaches the target model by selecting only the linear function in each restricted region. While if the size of the region is properly chosen, the group of linear functions can always capture the target function well enough. Instead of using simple linear regression by solving for the least square solution, the robust linear regression is considered to get the regression model to improve the accuracy and the robustness of the regression.

Given input x and output y of the data samples, the robust linear regression solves for the least square solution of a linear function and computes the estimated $\hat{y}$ from the regression solution, then gets the residual between y and $\hat{y}$ for each data sample. Based on the strength of residuals, a certain weight is computed for each data sample and applied to the input x and output y, then the weighted input x and output y of all data samples are used to get another linear least square solution. The new solution will be used to do another set of $\hat{y}$, residuals and weights, and the data samples will be re-weighted. This procedure is being executed iteratively until the linear least square solution converges.

The robust linear regression will be executed similarly in each piece (region), so every region needs to have a certain amount of data before the model of that region can be trained. The amount of the training data required to fill up the region is usually very huge.

Figure 4 and Figure 5 show some examples of PWLRR model, the data and model from any number of input dimensions larger than 2 cannot be easily visualized, so the examples only contain 1 and 2 input dimensions.

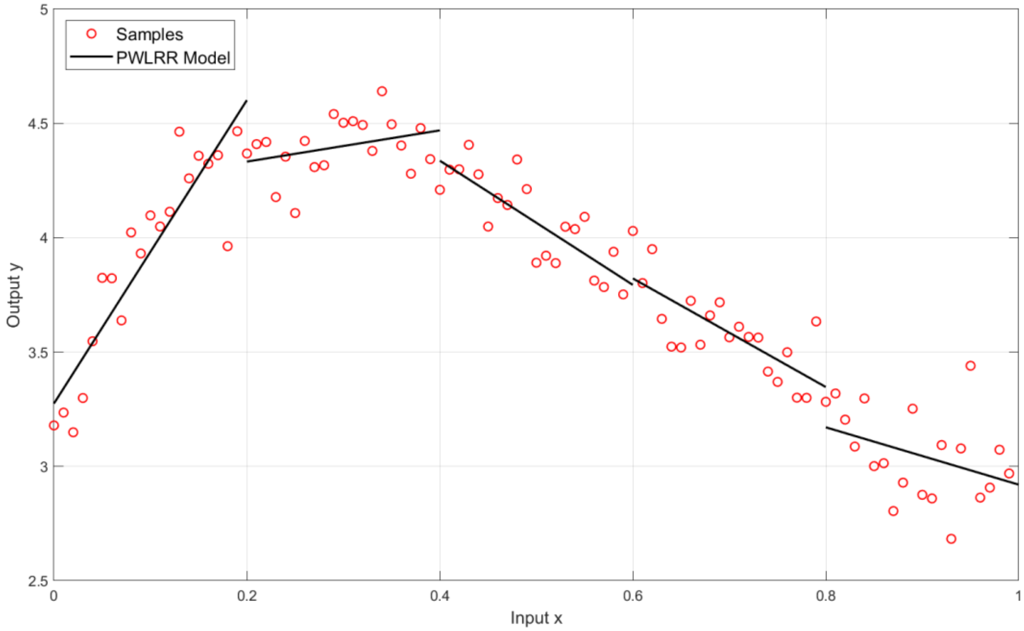

Figure 4 Example of 1-D PWLRR model

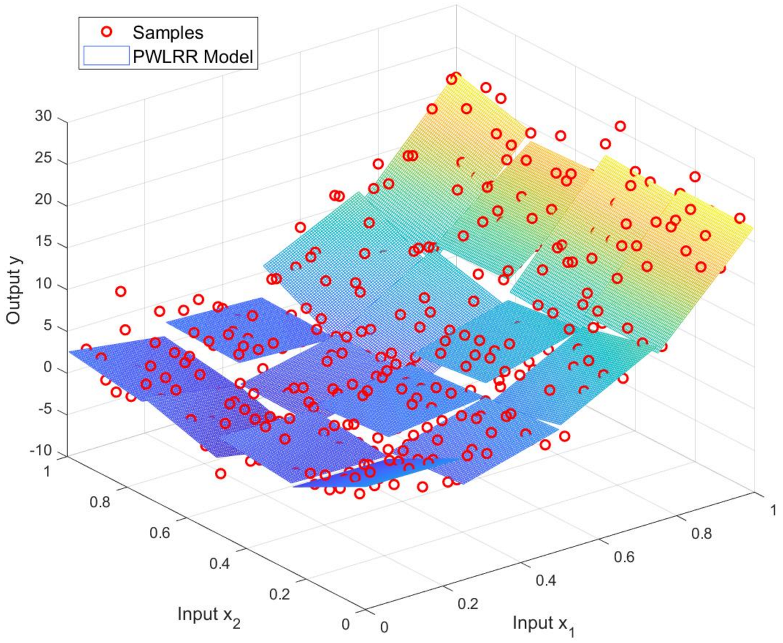

Figure 5 Example of 2-D PWLRR model

2.2 Model Learning and Blind Spots

When a fault detection system is initially activated on an induction motor driving asset, since the cold start learning is required for each asset, no model is available at the beginning the data collection. The data collected initially must be accumulated to a certain amount before some level of model learning can be initiated. During this period, the data cannot be used for predicting the health of the asset, this lack of prediction creates initial blind spots. Similarly, under any circumstance, when a data collected by the fault detection system cannot be used for prediction for any reason, this data is defined as blind spot data. And there are two types of blind spots associated with the PWLRR model learning.

2.2.1 Initial Blind Spots

The preferred amount of data used for baseline model generation cannot be acquired all at once, doing so would require the fault detection system not being functional for up to a year. As a result, the initial model generation is obtained using a smaller window. The smallest window that can be currently used for model generation is approximately one month and even this window duration comes at a significant risk of an unreliable model. A window of at least three months is considered sufficient in obtaining high fidelity models. During the first one to three months, all the acquired data are used to generate the baseline and perform necessary parameter selection (initial training), therefore, an initial blind spot lasts for one to three months is created.

2.2.2 Follow-on Blind Spots

Another peculiarity of the specific model generation problem we solve is the lack of diversity in the input space of the training data used. The selection of the training set is governed by the variation in the operating conditions of the physical system being monitored and it cannot be arbitrarily selected. Typically, physical assets, and especially industrial assets, are operated at near the same steady-state conditions with stable power quality most of the time. As a result, the training set, even within a few months, covers a very small region of the possible input space. And after the initial training is over, the fault detection system is commissioned and starts to monitor the asset's health, but whenever the asset operating condition or the power quality are changing, the inputs of the acquired data will move out of the region which is covered by the initial training, for the current model-based fault detection system, these data have to be either ignored from the system (without follow-on training) or used for baseline learning in that region so that once the region is covered by enough data, the baseline can be established, then the future coming data in the region can be used for fault detection (with follow-on training). In the real world, the fault detection system without follow-on training was never considered applicable, since amount the acquired data which are going into the regions not covered by the initial training is significant, ignoring them completely will make the system unable to perform any valuable fault detection for the asset. And even with follow-on training, the first batch of data required to train the baseline model in any non-covered region will have no ability to provide fault detection, resulting in the creation of follow-on blind spot. A follow-on blind spot may last a few hours to a few weeks and can happen as frequent as possible. An example of operating condition and power quality change is shown in Figure 6 and Figure 7.

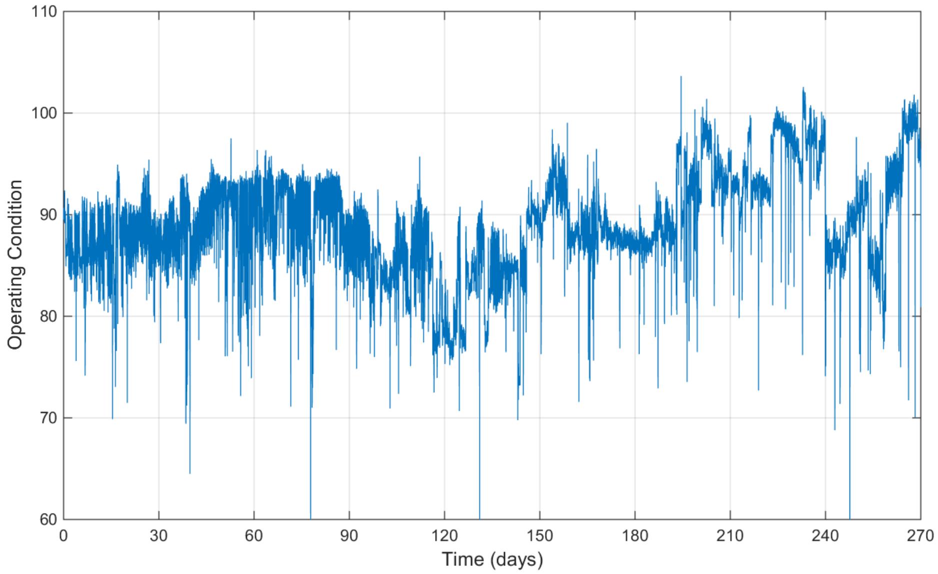

Figure 6 Change in operation condition over time

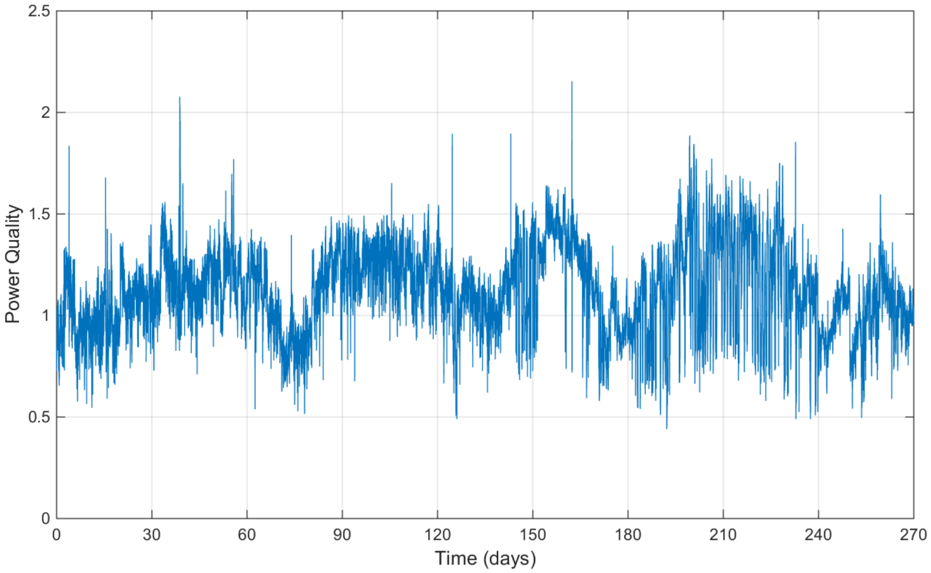

Figure 7 Change in power quality over time

As shown in the figures, if the initial training of the PWLRR model uses 30 days of data, the initial trained model can only cover the region of operating condition from roughly 83 to 91, power quality from 0.7 to 1.3. After the asset is being operated for a while, between the day 180 and the day 240, the operating condition goes to the region between 91 and 100, while power quality goes to the region between 1.3 and 1.5, most of these data cannot use initial trained model to do prediction, and the blind spots are inevitable. Noted that the above situation is only described one dimension at a time, for a real fault detection system with 5-dimensional inputs space, the situation can only be worse.

2.3 Fault Detection Logic

A simple fault detection logic is introduced to the fault detection system in this research.

After some level of model training is accomplished and the system can start making predictions, the predicted output is compared to the measured output to compute the absolute prediction error (PE):

$$PE = |y - \hat{y}|$$

For every data with prediction available, since the inputs-output data are time series data, the PE is also a time series. The time series PE is compared to a fixed threshold thereafter, the fault detection logic in this work is: on the PE vs time plot, whenever the PE is above the threshold, a fault decision will be made from the system and the decision will be further compared to the ground truth acquired from other source (the designed testing plan of a lab test, or the failure report from the production site) to determine its accuracy. If the fault decision is made and there is a fault introduced in the lab, or there is a failure on the production asset after a while and the fault decision sustained during this period, then the fault decision is a successful decision. Otherwise, the fault decision is a false alarm.

The determination of fault detection threshold for PE will be explained in the next section, together with other parameters. Once the value of the threshold is determined, for easy comparison, the threshold will be normalized to 1 by multiplying a scaler to it, correspondingly, the same scaler will also be applied on the time series PE to perform direct comparison. Figure 8 shows an example of time series PE goes over the threshold, for this situation, based on the logic described above, a fault will be made at data with time index around 3600.

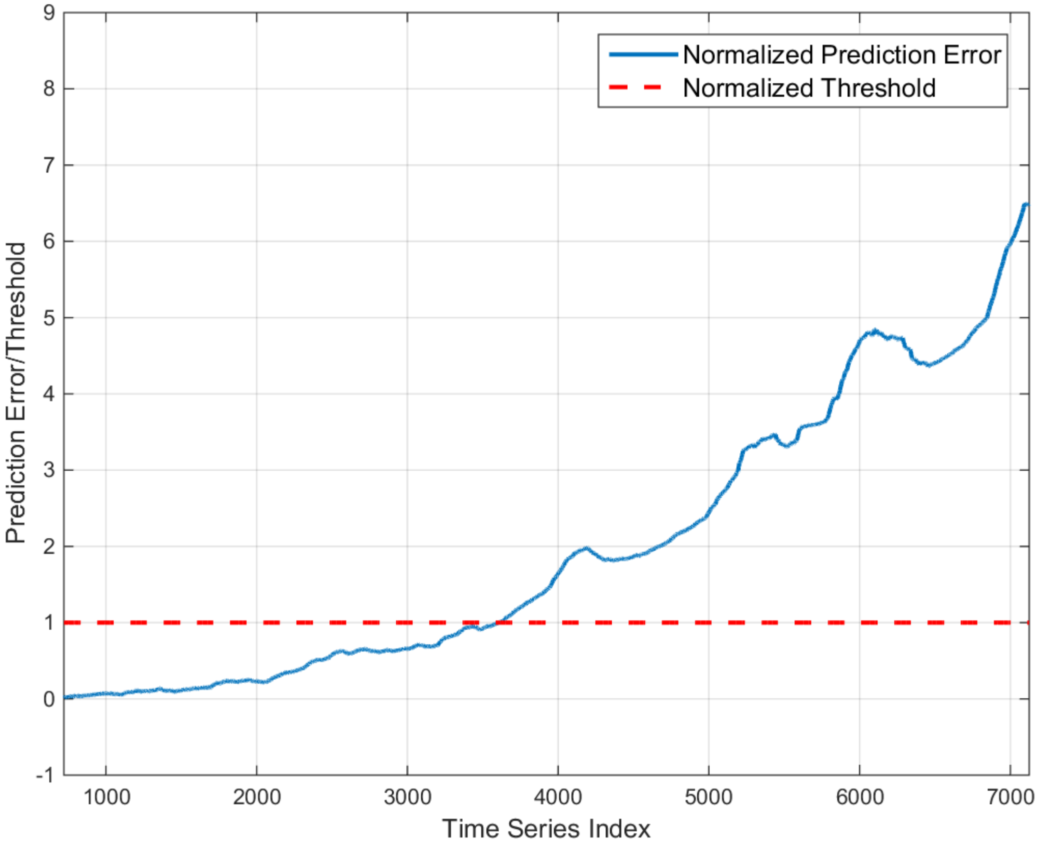

Figure 8 Example of time series PE goes above threshold

3. PROPOSED METHOD FOR LEARNING MODELS

GP for regression is proposed to be the alternative model learning method in the model-based fault detection system to substitute PWLRR if it can achieve better performance.

3.1 Gaussian Process

GP is a generalization of the Gaussian probability distribution which was first started in the statistics field, and it was considerably developed in the seventies in the geo-statistics. In mid-nineties of last century, GP was introduced in the machine learning field, especially by Rasmussen and Williams [16].

3.1.1 Overview

GP is based on Bayesian analysis, it assumes a prior distribution on class of functions to be modeled. This is determined based on our knowledge of the problem, before observing any training set. Then the observation of the training set is combined with the prior to get the prediction. Let $p(f)$ denote the prior probability for function f. The posterior probability is obtained by the Bayes rule as:

$$p(f|D) = \frac{p(D|f)p(f)}{p(D)}$$

where D denotes the training set.

Figure 9 shows the definition of GP for regression. In the figure, (x_1, y_1) and (x_2, y_2) are training input/output pairs, x_* is the test data input, f are inherent function values and e are zero mean Gaussian noises. The task of GP for regression is to estimate f on both training data and test data.

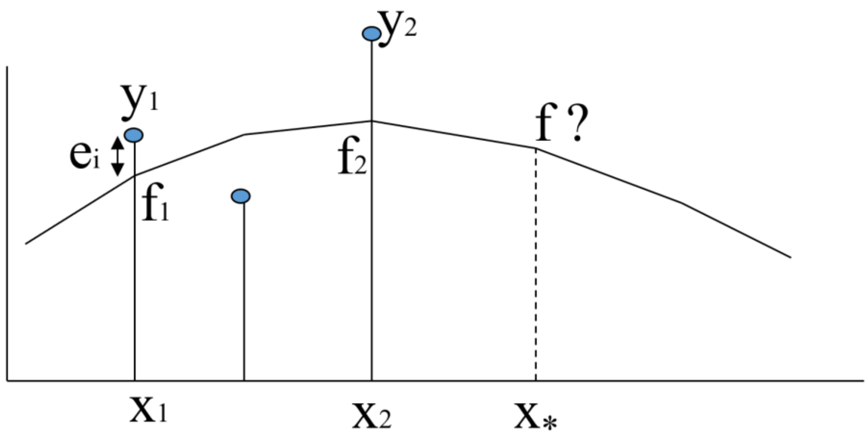

Figure 9 GP for regression definition

The smoothness of the f curve is enforced by assuming f are jointly Gaussian distributed (as a prior). Patterns that are close (in the input space, e.g. very similar feature vectors), must have highly correlated f. And the covariance is high if inputs are nearby, otherwise it is low.

Let K$\left(x_i,\ x_j\right)$ be the covariance function, and define the covariance matrix between the training and prediction points as:

$$k_{**} = K(x_*, x_*)$$

$$k_* = K(\bar{X}, x_*)$$

$$K = K(\bar{X}, \bar{X})$$

Where $\bar{X}$ is the training data inputs, x* is the prediction data input, then the prediction f* is given by the following equation:

$$f_* = k_*^T(K + \sigma_n^2 I)^{-1}\bar{y}$$

And variance of prediction is given by:

$$Var(f_*) = k_{**} - k_*^T(K + \sigma_n^2 I)^{-1}k_*$$

It can be observed that the variance only depends on the inputs space.

Figure 10[22] shows an example of Gaussian process on a regression problem with a squared exponential kernel. Left plot are draws from the prior function distribution. Middle are draws from the posterior. Right is mean prediction with one standard deviation shaded.

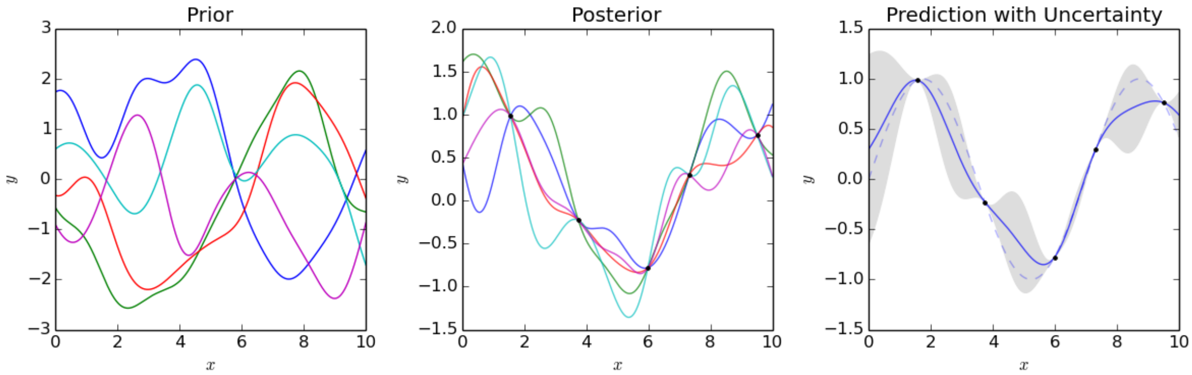

Figure 10 Gaussian process for regression. Reprinted from [22]

Besides the squared exponential covariance function, there are many other common kernel functions can be selected:

- Constant: $\mathrm{K}(x, x') = C$
- Linear: $\mathrm{K}(x, x') = x^T x'$
- Gaussian Noise: $\mathrm{K}(x, x') = \sigma^2 \delta_{x,x'}$
- Squared Exponential [23]: $\mathrm{K}(x, x') = exp\left(-\frac{\|d\|^2}{2l^2}\right)$
- Ornstein-Uhlenbeck [24]: $\mathrm{K}(x, x') = exp\left(-\frac{|d|}{l}\right)$
- Matern [25]: $\mathrm{K}(x, x') = \frac{2^{1-v}}{\Gamma(v)}\left(\frac{\sqrt{2v}|d|}{v}\right)^v K_v\left(\frac{\sqrt{2v}|d|}{v}\right)$
- Periodic: $\mathrm{K}(x, x') = exp\left(-\frac{2\sin^2\left(\frac{d}{2}\right)}{l^2}\right)$

- Rational Quadratic [26]: $K(x, x') = (1 + |d|^2)^{-\alpha}$

Figure 11 shows the flowchart of GP for regression. It can be observed from the flowchart that when GP is used for prediction, the entire training data set is needed to be used to compute a large size covariance matrix. This demonstrate the most critical disadvantage of GP method: it does not scale well to large data sets.

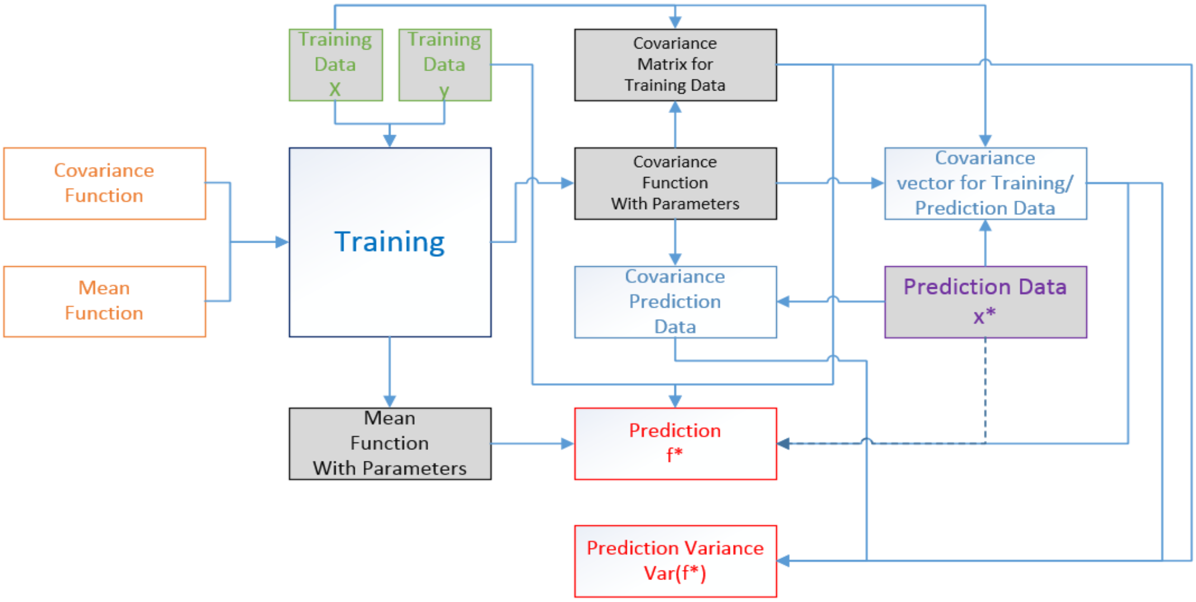
Training Data X
Training Data y
Covariance Matrix for Training Data
Covariance Function
Mean Function
Training
Covariance Function With Parameters
Covariance vector for Training/ Prediction Data
Covariance Prediction Data
Prediction Data x*
Mean Function With Parameters
Prediction f*
Prediction Variance Var(f*)

Figure 11 Gaussian process training and prediction

3.1.2 GP on Fault Detection System

To apply the model generation by GP to the fault detection system, the time series data collected by the fault detection system will be split into four groups. The data splits are similar between GP and PWLRR. For both methods, the initial training and validation data cannot be used to generate prediction, so the existence of initial blind spot applies to both method. However, the follow-on training only exists in PWLRR method, so the blind spots due to follow-on training only applies to the PWLRR method.

3.1.3 Parameters Selection

In this work, the squared exponential covariance function with isotropic distance measure is selected as the kernel function for learning the model of the fault detection system. And Gaussian likelihood function is selected to be the likelihood function for GP regression. There are in total three hyper-parameters in these two functions need to be selected

The squared exponential covariance function used in the GP tool library is given by the following equation [27]:

$$\mathrm{k}(x,z) = sf^2 \cdot exp\left(\frac{-(x-z)'(ell^2 \cdot I)(x-z)}{2}\right)$$

where sf and ell are the two hyper-parameters.

The Gaussian likelihood function used in the GP tool library is given by the following equation:

$$\mathrm{likGauss}(t) = \frac{exp\left(-\frac{(t-y)^2}{2 \cdot sn^2}\right)}{\sqrt{2\pi \cdot sn^2}}$$

where sn is the hyper-parameter.

These three hyper-parameters together with the number of initial training points are selected by maximizing the prediction accuracy on the validation data. To achieve better training quality, since the isotropic distance measure is used, the inputs are normalized to have comparable ranges among different dimensions.

For the PWLRR method investigated in this work, there are three hyper-parameters need to be selected: the size of the piece, the minimum number of data points to perform linear robust regression for each piece and the number of initial training points. Where the size of the piece is vector, the length of the vector is same as the inputs space dimensionality.

Same selection method is applied to the PWLRR hyper-parameters selection. The hyper-parameters are selected to maximize the prediction accuracy on the validation data.

For both GP and PWLRR methods, when selecting number of training data, the number of validation data and number of testing data are also fixed by a pre-determined constant ratio:

$$\text{Training}:\text{Validation}:\text{Testing} = 2:1:3$$

Besides the hyper-parameters selected for both methods with training and validation data, there is one more parameter needs to be determined from testing data, which is the threshold for PE in the detection logic. For both methods, the average PE is computed from the testing data, then the threshold is set to twice of the average PE.

There are two windows used in the fault detection system. One average window is applied on inputs-output time series. And one moving average window is applied on the PE time series. The sizes of the two windows are not optimized. They are just selected to be same for both GP and PWLRR methods.

3.2 Extrapolation Measure

To compare the extrapolation ability between the two methods, the volume of the convex hull of the input space is considered to be used as the performance indicator, the percentage extrapolation for one or a group of prediction data points is defined as:

$$\text{Extrapolation} = \frac{Vol_{ConvexHull}(training + \ prediction)}{Vol_{ConvexHull}(training)} - 1$$

where $Vol_{ConvexHull}(*)$ is the volume of the convex hull. Figure 12 shows an artificial example on using convex hull to estimate the input space extrapolation. For this 2-dimensional inputs example, the volume is just the area surrounded by the lines on the convex hull edge.

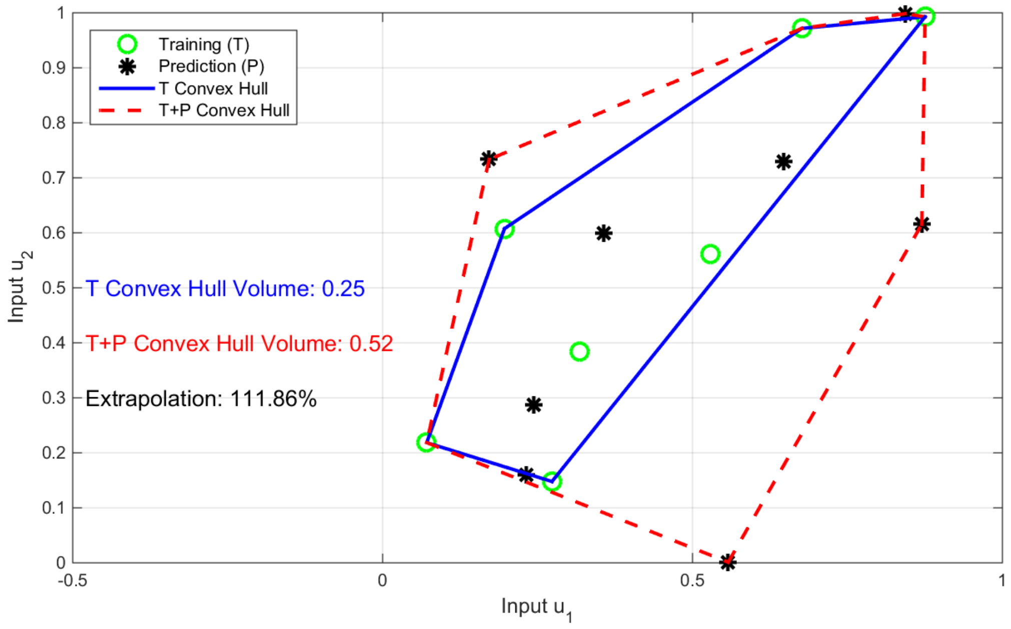

Figure 12 2-D convex hull and extrapolation example

In this research Qhull is used to compute the convex hull, Qhull implements the Quickhull algorithm [28] for computing the convex hull. Basically, for each possible facet of the given data points, if all other points (not on the facet) are located at one side of the facet, then this facet is a facet on the convex hull. Rather than searching all the possible facets to determine whether it is on the convex hull or not, the Quickhull starts from an initial facet, and expand the convex hull by selecting the farthest outside point of a certain facet on the convex hull, then exclude the points which are inside the expanded convex hull, repeat the procedure until there is no point outside [29][30].

3.3 Prediction Limitation

The GP can train global models and can perform predictions in regions without prior training points. However, if the prediction points are “too far away” from the training points then the confidence of the predictions drops, and the predictions become unreliable.

Considering that the basic assumption in GP is that data close to each other in the input space must have highly correlated outputs. To limit the use of “too far away” points in prediction, the norm (or distance) between the prediction and training points is computed to determine prediction points that are “too far away” from training points. For each collected data after the initial GP model is trained, the norms between this data and all training data are computed, then the average of the smallest 10% norms is computed, this value is used to determine whether the data can be used for prediction or not.

If this average norm is high, then the data is rejected from being used in predictions; this also rejects data located in a sufficiently large “hole” within the training set. Figure 13 shows an example of using the average norm to reject the data from prediction. Since the average norm can be computed for each time series data after the initial training, the norm can also be constructed as a time series which is shown in the figure. In this example, when the average norm is larger than 2, the data will be rejected from prediction. It can be observed that between day 160 and 220, most of the collected data are rejected from the prediction.

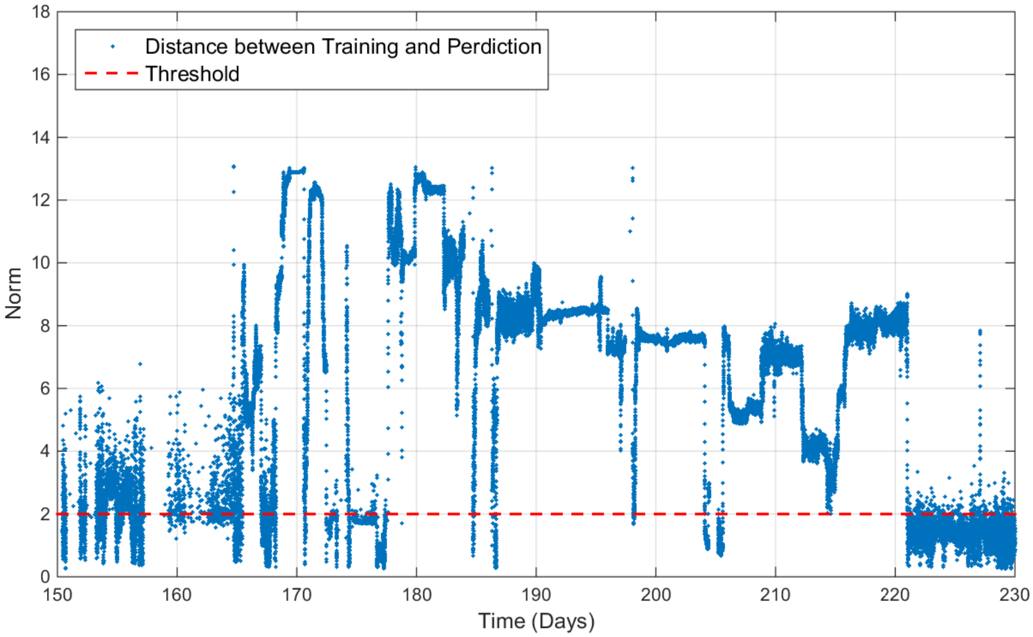

Figure 13 Example of using the norm to reject prediction points

When a data is rejected from prediction, this data will be marked as blind spot data. This definition is similar as the follow-on blind spot data from PWLRR, although the causes of not being able to do prediction of the two methods are different. The follow-on blind spot data is processed differently for these two methods, for GP, the data is discarded, for PWLRR, the data must be used as follow-on training data.

Ideally, the comparison between GP and PWLRR should be done when both methods are having similar learning pattern. Since in this work, there is no follow-on training in the GP method, the performance of PWLRR method should also be evaluated when the follow-on training for PWLRR is disabled. But further investigation indicates that PWLRR is not be able to provide any effective prediction without follow-on training. And this is the reason for the decision of enabling follow-on training in PWLRR is made.

To demonstrate that the PWLRR is not be able to provide any effective prediction without follow-on training, one case from real world is selected (this the case #1, details will be described in the next section). Three model learning options, GP without follow-on, PWLRR with follow-on and PWLRR without follow-on are applied to this case. Figure 14 shows the PE time series from the three learning options on the same plot. To display the lack of the prediction availability, markers with different color are used in plotting. The “red square” markers which represent the prediction from PWLRR without follow-on training are vary sparse in the plot.

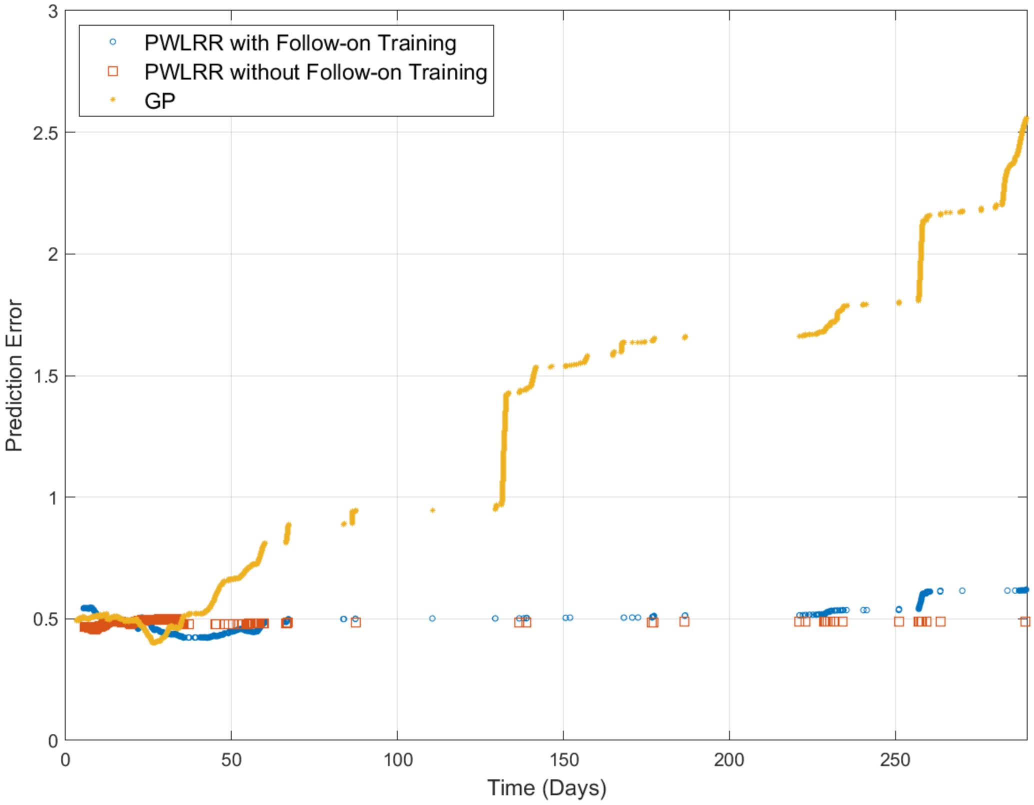

Figure 14 PE time series of the three options

The blind spot is used to further demonstrate the lack of the prediction availability for PWLRR without follow-on training. The blind spot is computed using a monthly moving average window. Therefore, the blind spot time series can be constructed. Figure 15 shows the blind spots vs time for the tree options. It is showing that within 2 months after the initial training, the system trained by PWLRR without follow-on training has more than 95% blind spots. It is not possible to perform effective prediction with such a high blind spot ratio.

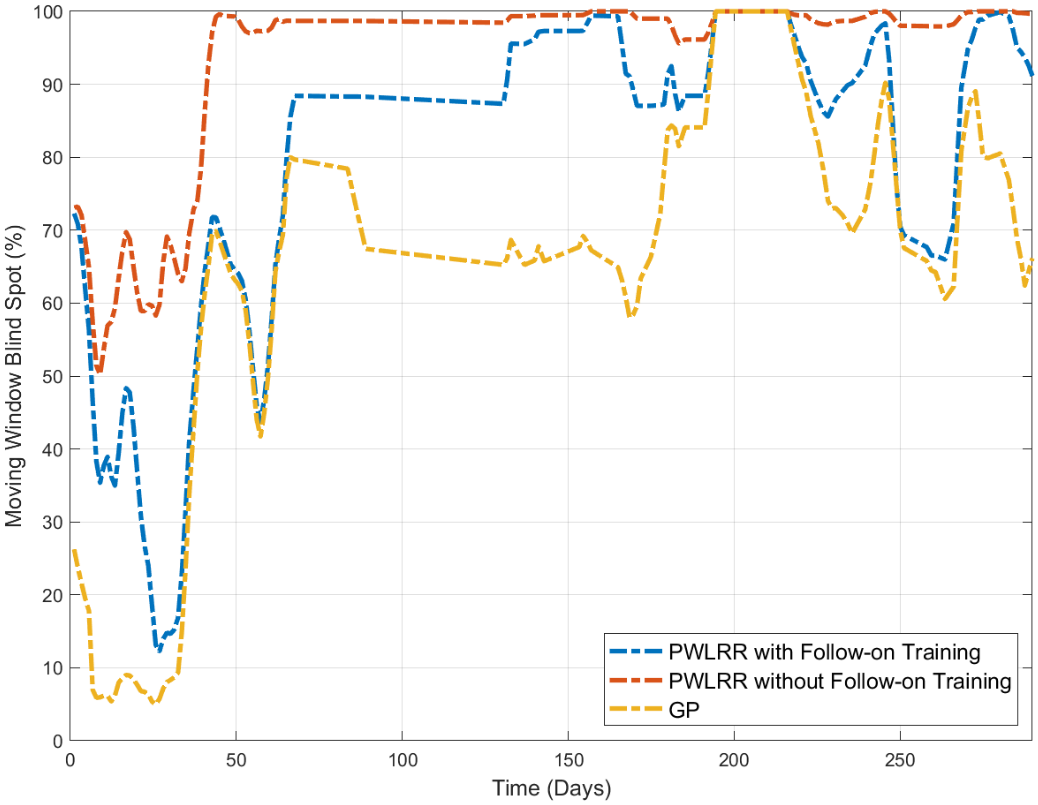

Figure 15 Blind spot time series of the three options

3.4 Performance Indicators

There are in total seven performance indicators are defined for learning methods benchmarking purpose, some of them are already discussed previously. They are summarized and listed as follows:

- **Number of Training & Validation Points**: The amount of data used to train the model (lower is better).
- **Blind Spots**: If a data point cannot be used for prediction due to its distance from training, it is counted as a blind spot. For PWLRR, this includes the data for initial training and the data for follow-on training. For GP, this includes the data for initial training and the data rejected by the norm criterion. The blind spot is presented as a percentage ratio of the blind spot data and the total data (lower is better).
- **Extent of Extrapolation**: As defined in previous slide (higher is better).
- **Testing Set Average PE**: The average PE from the testing data, which is used for determination the threshold (lower is better).
- **Detection Time**: If there is a confirmed fault, the detection time is measured by the number of data pointes from the time when PE is over the threshold until failure (Failure at Production Site/End of the Lab Test). If PE goes above the threshold at multiple times, only the last crossing is considered. If PE is below the threshold at the end, detection time is 0 (higher is better).
- **True Positive**: If there is a confirmed fault, and if the PE is sustained above the threshold, a true positive is counted (higher is better).
- **False Positive**: If there is confirmed no fault, every time the PE goes above the threshold, a false positive is counted (lower is better).

All these performance indicators will be accessed on the cases with real word data in the next section. Although due to the lack of complexity, some of the performance indicators cannot be applied to the cases with artificial data. The details will be discussed case by case in the next section.

4. RESULTS

Two types of cases are used in testing the learning methods: artificial cases and real cases. Both PWLRR and GP are applied to the cases.

4.1 Artificial Cases

There are in total 5 artificial cases prepared for testing. These cases include different number of input dimension: from 1-dimensional input to 4-dimensional inputs. Some cases only compare the leaned model to target without performing prediction while other cases also test the prediction part. One of the cases is used to test the accuracy of the learning when there is a “hole” in the input space of the training data. One of the cases is used to test the accuracy of the prediction when there is a simulated fault.

4.1.1 1-D Input

In this case, 100 single input single output data points are created, the inputs of the data are created by uniformly distributing them in between 0 and 1, and the outputs of the data are created by a static target model function:

$$y = 9x^2 + 5\cos(3x) + \log(x + 0.05) + 1.0 + noise$$

where x is the input, y is the output, and the *noise* is random number generated from normal distribution with 0 mean and 0.15 standard deviation.

The GP models are learned with randomly selected 4 data points, 8 data points, 25 data points and all 100 data points with parameters selected to minimize the prediction error on the training data. The PWLRR model is learned with all 100 points and 5 equally sized pieces. Figure 16 shows the model functions trained with GP by using 4, 8, 25 and 100 points. Figure 17 shows the model function trained with PWLRR by using 100 points. Figure 18 and Figure 19 show the difference in between model function trained by GP and PWLRR and target model function.

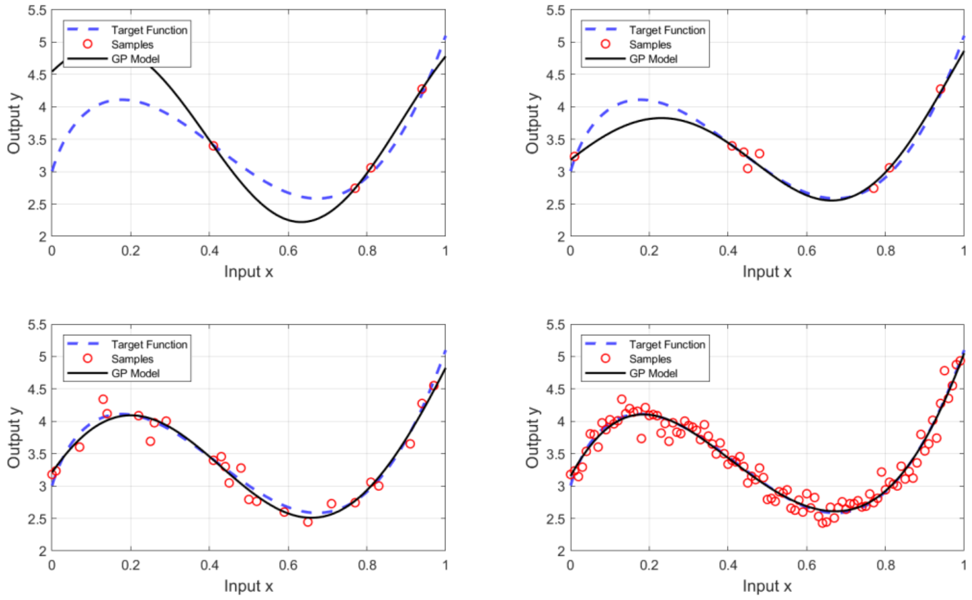

Figure 16 GP models trained by different number of data points

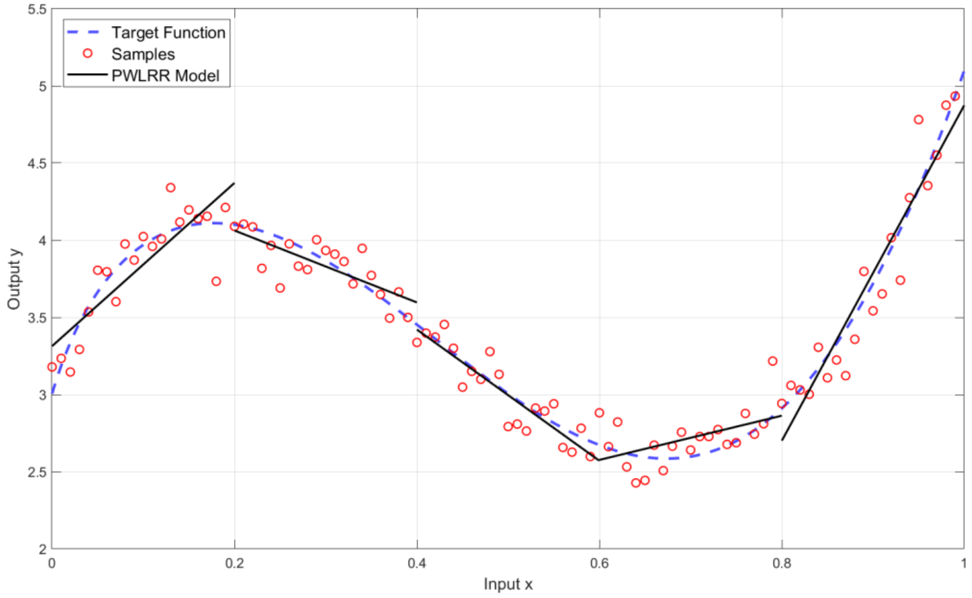

Figure 17 PWLRR model trained by all 100 data points

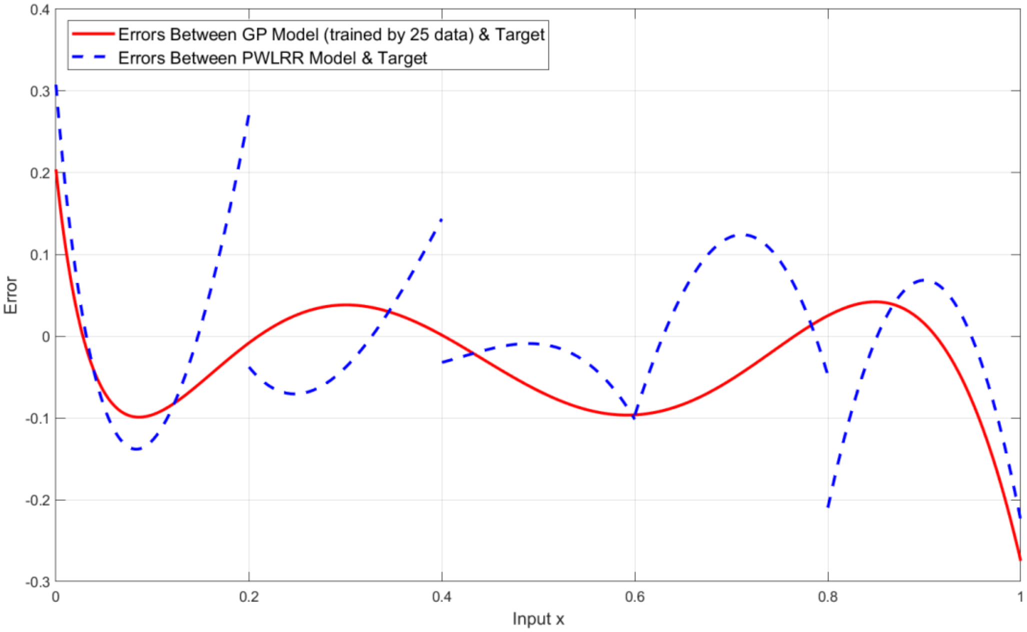

Figure 18 PWLRR model error and GP model (25 data) error

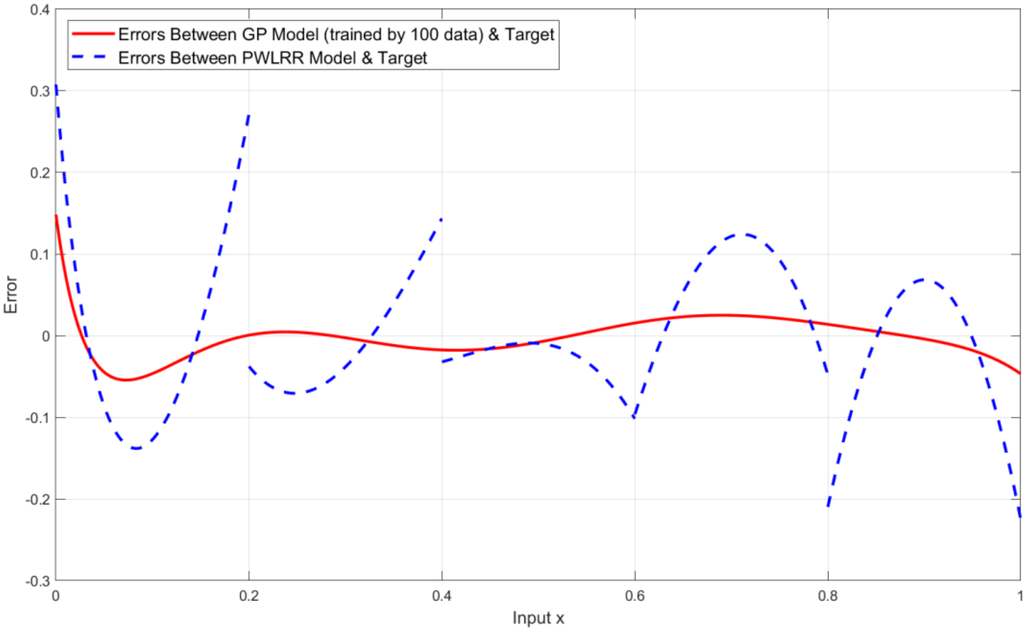

Figure 19 PWLRR model error and GP model (100 data) error

The trained models and target model is plotted in a 2-D figure as input vs output, so the models can be visually reviewed. To compare the accuracy of the learned model, the error between leaned model and target model is computed by taking the mean of the absolute difference of the two functions. Since the difference can only be taken with discrete values, the discrete outputs from the learned model and the discrete outputs from the target model are generated by inputs with 0.0001 step between 0 and 1. And the two outputs are used to get the absolute difference. The results show that with same number of training data, the error between GP model and target (0.0173) is much smaller than the error between PWLRR model and target (0.0677). And even the GP model trained with one fourth of the data (25 data) is achieving a smaller error (0.0539) than the PWLRR model. This is an indication that to achieve same level of training (get the comparable prediction error), the GP may be able to use less data therefore do faster training.

4.1.2 2-D Inputs

In this case, 400 double inputs single output data points are created, the inputs of the data are created by uniformly distributing them in between 0 and 1 for both input dimensions, they are equally spaced from each other in the input space. Figure 20 shows the distribution of the data in input space. And the output of the data is created by a static target model function:

$$y = 20x_1^3 + 5\cos(x_1 x_2) + 7\cos(5x_2) + \log(x_2 + 0.05) + 1.0 + noise$$

where x_1 and x_2 are the inputs, y is the output, and the noise is random number generated from normal distribution with 0 mean and 2.5 standard deviation.

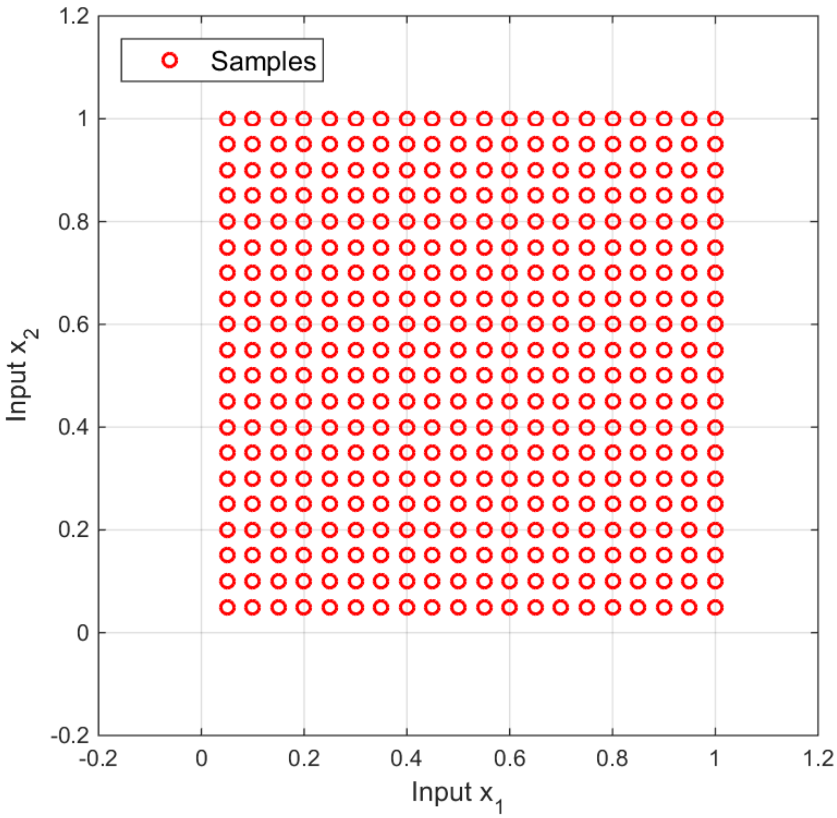

Figure 20 The distribution of the data in input space

The GP models are learned with randomly selected 200 data points and all 400 data points with parameters selected to minimize the prediction error on the training data. The PWLRR model is also learned with all 400 points and 16 equally sized pieces in total (4 equally sized pieces for each dimension). Figure 21 shows the target model function and the artificial data samples. Figure 22 shows the data samples the trained PWLRR model. Figure 23 shows the data samples and the trained GP model with 200 data. Figure 24 shows the data samples and the trained GP model with 400 data. Figure 25 shows the error between the PWLRR model and target model. Figure 26 shows the error between the GP model trained by 200 data and target model. Figure 27 shows the error between the GP model by 400 data and target model.

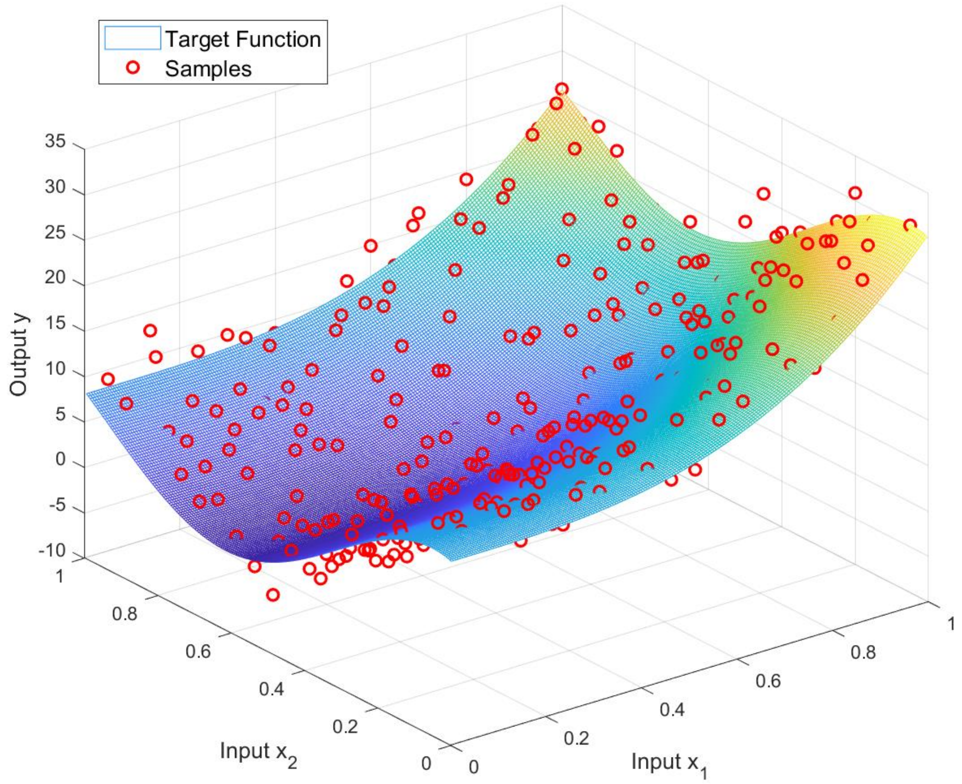

Figure 21 The target model function and the artificial data samples

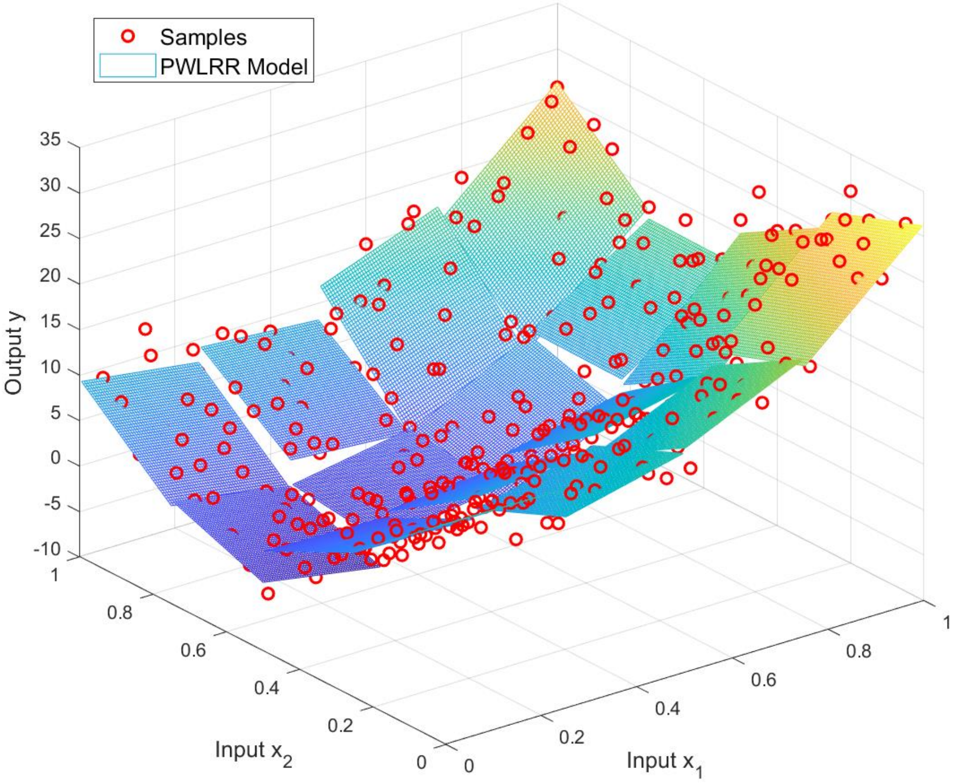

Figure 22 The PWLRR model

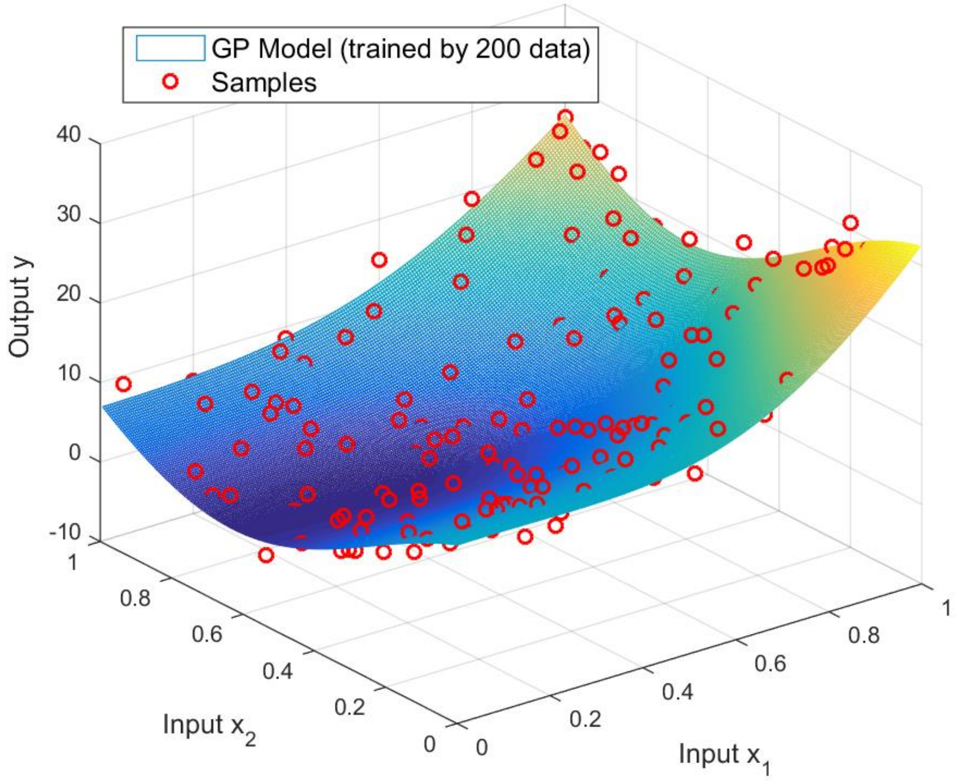

Figure 23 The GP model trained by 200 data

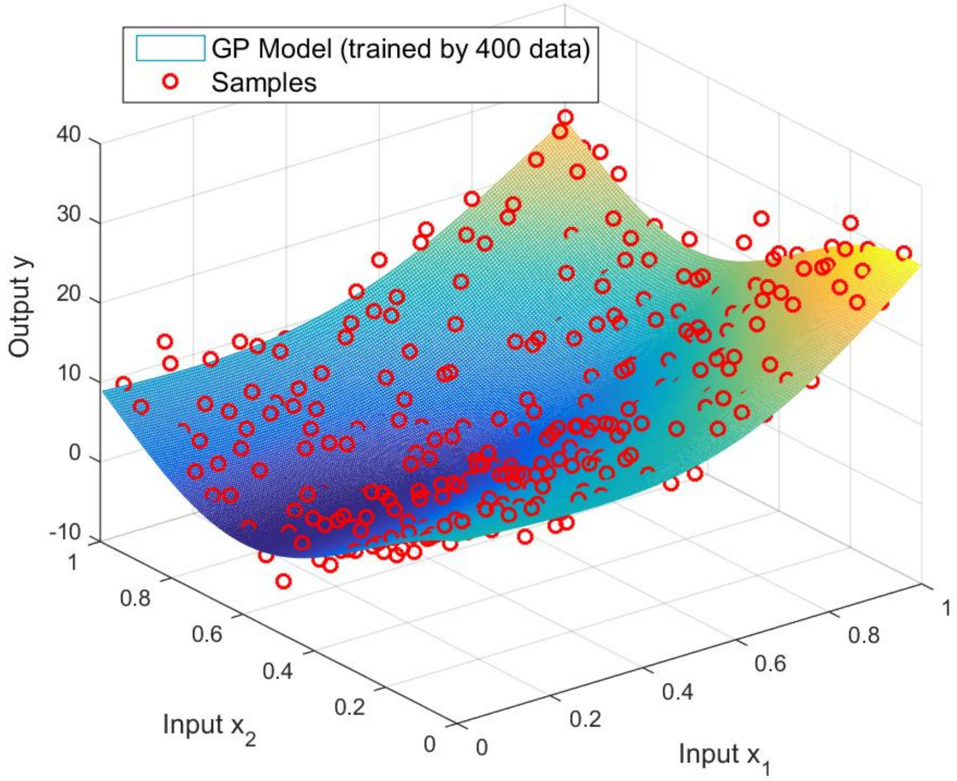

Figure 24 The GP model trained by all 400 data

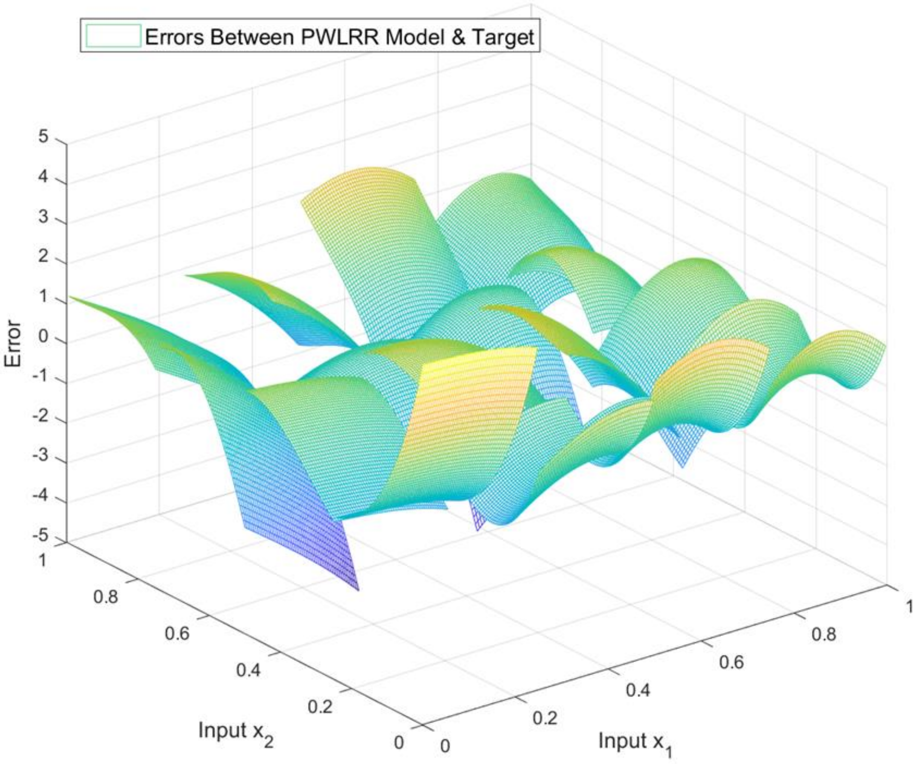

Figure 25 The error of PWLRR model

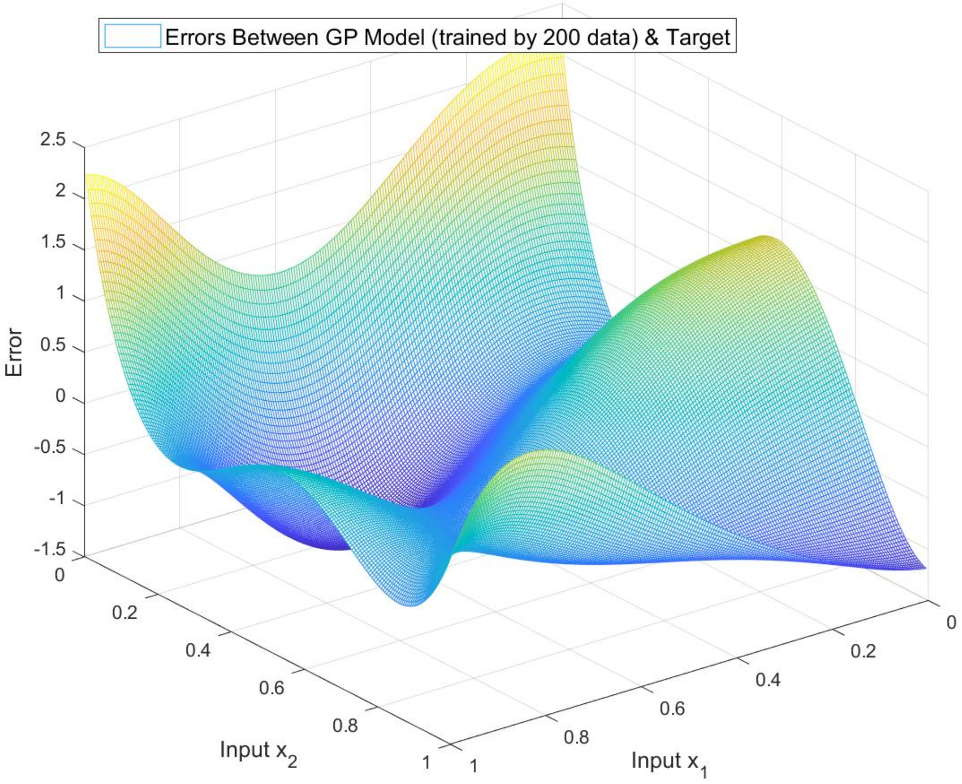

Figure 26 The error of GP model trained by 200 data

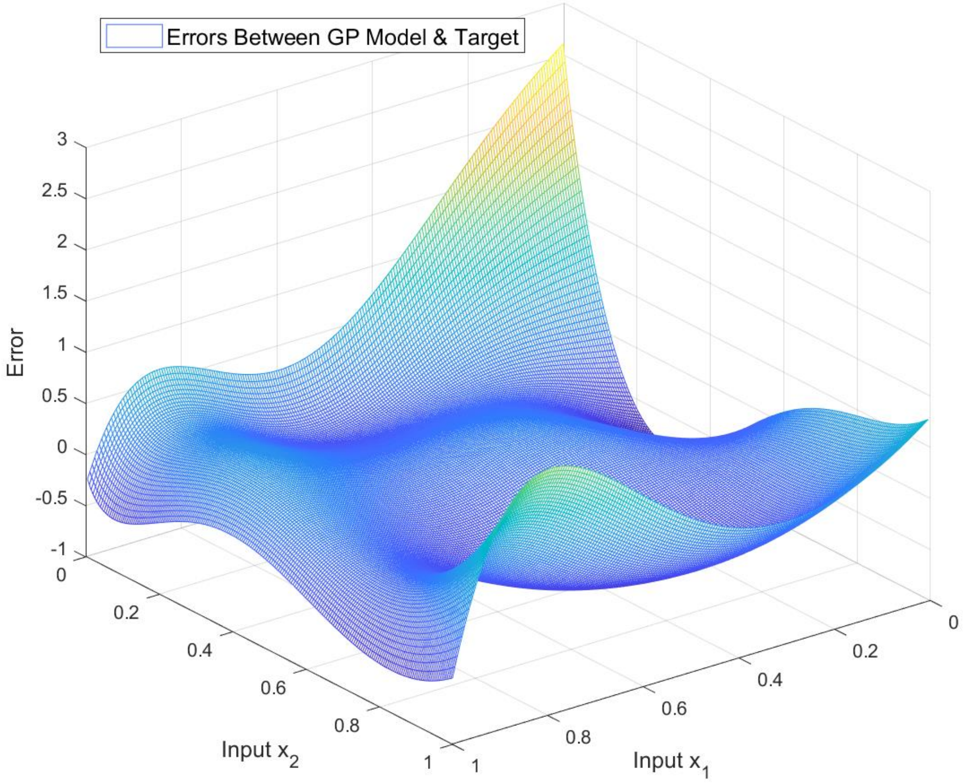

Figure 27 The error of GP model trained by all 400 data

The trained models and target model is plotted in a 3-D figure as 1st-input vs 2nd-input vs output. So, the models can still be visually reviewed. To compare the accuracy of the learned model, the error between leaned model and target model is computed by taking the mean of the absolute difference of the two functions. Since the difference can only be taken with discrete values, the discrete outputs from the learned model and the discrete outputs from the target model are generated by 40,000 inputs with 0.005 between 0 and 1 for each input dimension (200 inputs for each dimension). And the outputs are used to get the absolute difference. The results show that with same number of training data, the error between GP model and target (0.2434) is much smaller than the error between PWLRR model and target (0.7658). And even the GP model trained with half of the data (200 data) is achieving a smaller error (0.5376) than the PWLRR model. This is a further indication that the GP may be able to use less data to achieve same level of training.

4.1.3 2-D Inputs with Hole in Input Space

In this case, 400 double inputs single output data points are created in the same way as the previous 2-D case. Also, the output is generated with the exact same target model function as the previous 2-D case.

One GP model is trained with all 400 points as reference, same as the previous case. Then to show the impact of the situation when there is a “hole” in the training data input space (i.e. there is region in the input space is not covered by the training data, while the surrounding regions of the uncovered region are all covered by the training data), 121 data located in a 0.5 by 0.5 region in the center are removed from training, leaving 279 data for training. Figure 28 shows the input space location of the remaining 279 data. To make reasonable comparison, another GP model is also trained with 279 data by randomly removing 121 data from 400 data. All GP parameters are selected to minimize the prediction error on the training data.

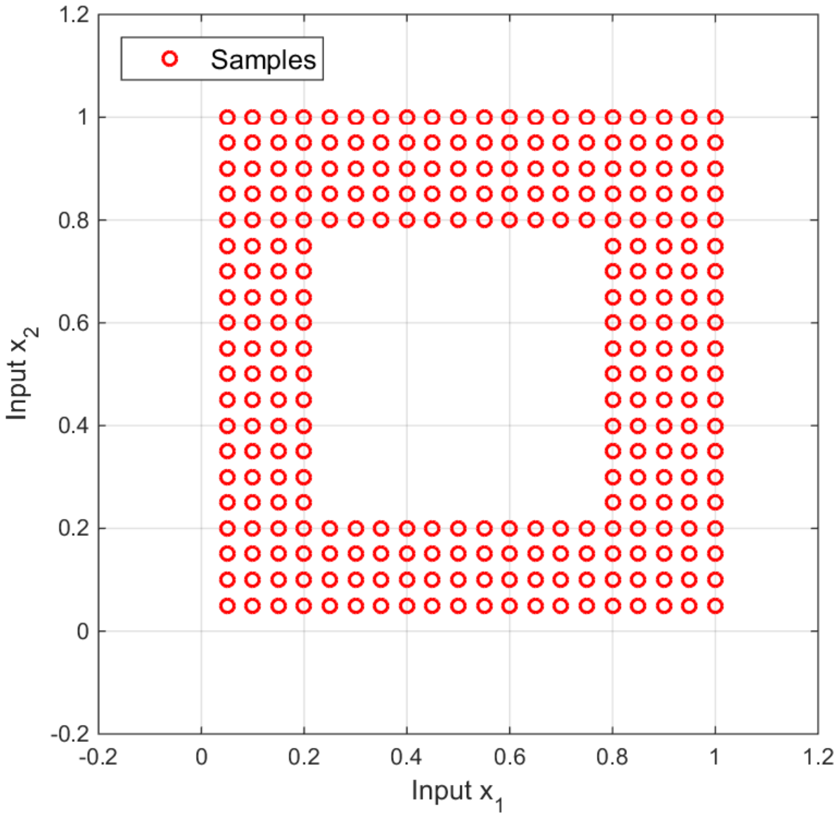

Figure 28 The distribution of the data in input space with a hole

Since this is still a 2-D cases, the trained GP models can be visually reviewed by the 3-D figure as 1st-input vs 2nd-input vs output similarly as the previous case. Figure 29 shows the data samples and the trained GP model by 279 data with an input space hole in the center. Figure 30 shows the data samples and the trained GP model by 279 randomly selected data. Figure 31 shows the error between the GP model trained by 279 data with an input space hole in the center and target model. Figure 32 shows the error between the GP model trained by 279 randomly selected data and target model.

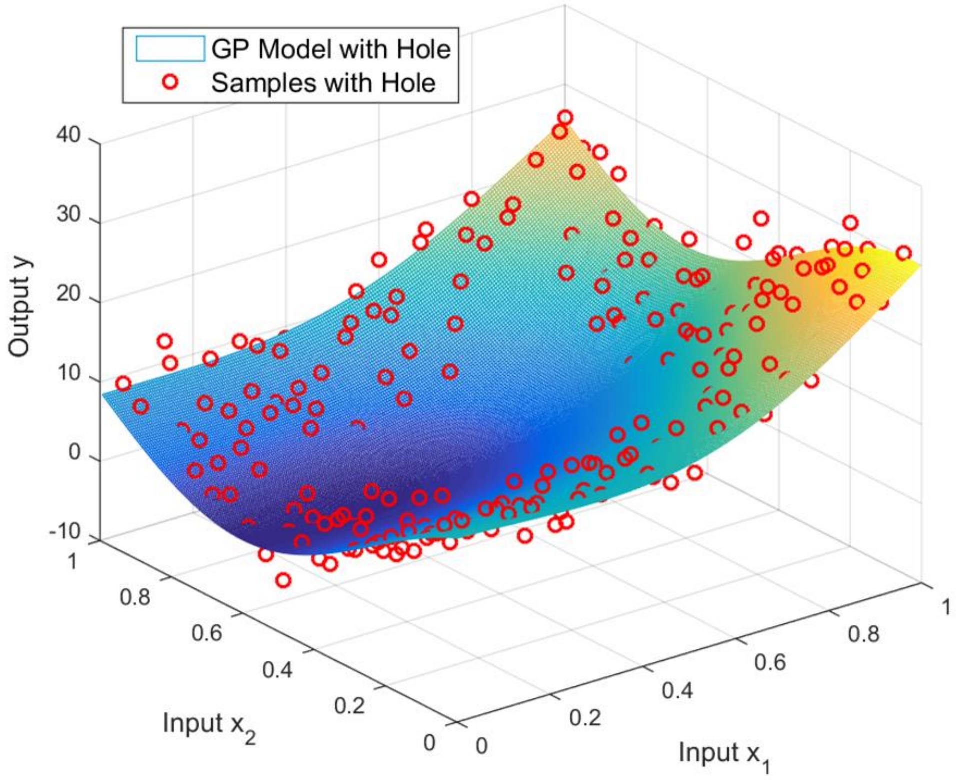

Figure 29 GP model trained by 279 data with an input space hole in the center

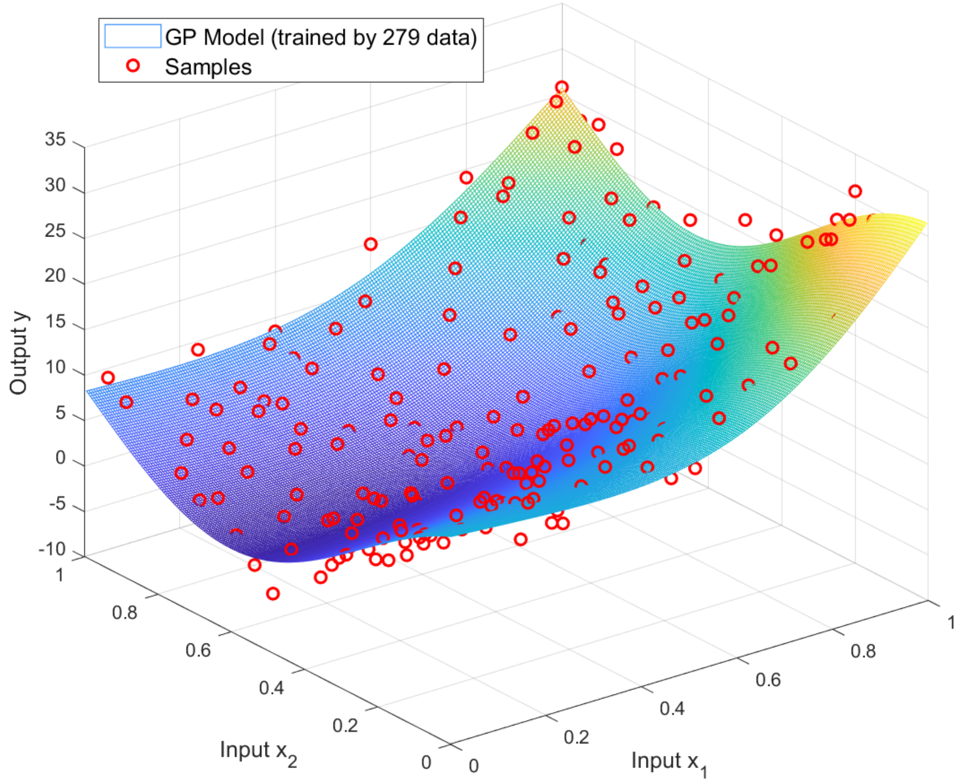

Figure 30 GP model trained by 279 random selected data

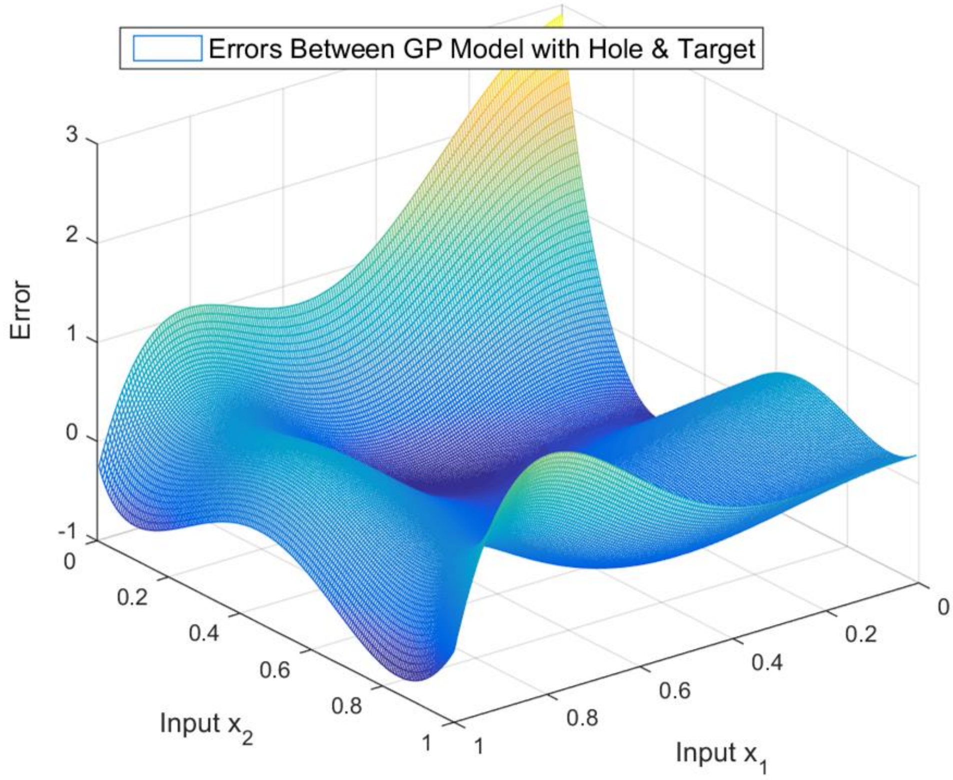

Figure 31 The error of GP model trained by 279 data with an input space hole in the center

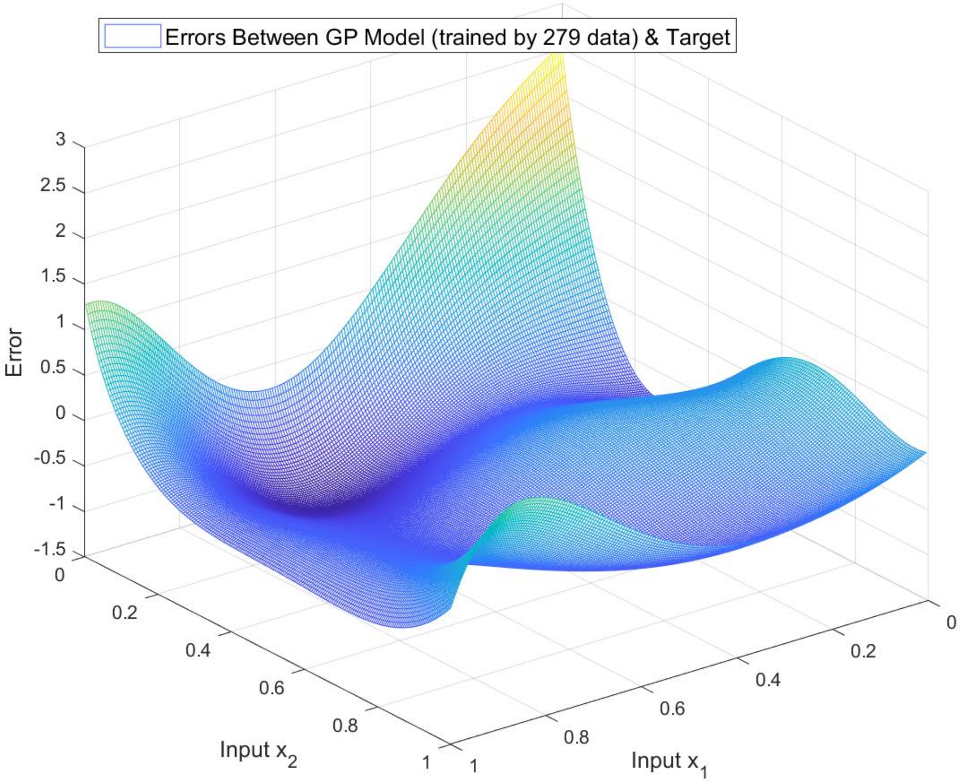

Figure 32 The error of GP model trained by 279 random selected data

The errors are also computed similarly as the previous case with 40,000 discrete values generated from 40,000 uniformly distributed data in the input space. The results show that while the model trained by all 400 points is giving the smallest error (0.2434). Both models trained by 279 data with a input space hole in the center and by 279 randomly selected data are getting larger error, while the error (0.3519) of the model trained with a input space hole in the center is a bit smaller than the other model (0.4330). This indicates that a reasonable size of a input space hole will likely not impact the accuracy of the GP model. But rejection by norm as mentioned in Section

3 will always be applied irrespective of interpolation or extrapolation to safe guard again the situation that the uncovered center region is large enough to impact the accuracy of the model.

4.1.4 4-D inputs with Static Model

In this case, 7125 data with four inputs one output are tested. in addition to the inputs and output, a time index from 1 to 7125 is also associated with each data, since this is still an artificial case, only the time index is used rather than the timestamp. So that when the data which are not used for training are used for prediction, the prediction from each data, if available, will also has a time index since each data has a time index. Therefore, the prediction can be sorted by the time index to construct a prediction vs time trend to simulate the behavior of a real fault detection system.

The four inputs, instead of being created from any artificial distribution, are generated from real data. They are produced from normalized inputs used in a real fault detection, the temporal sequence of the inputs is maintained, just the timestamp associated each input is replaced by the time index. Figure 33 contains the plots of the four inputs vs time index. All inputs are normalized between 0 and 1.

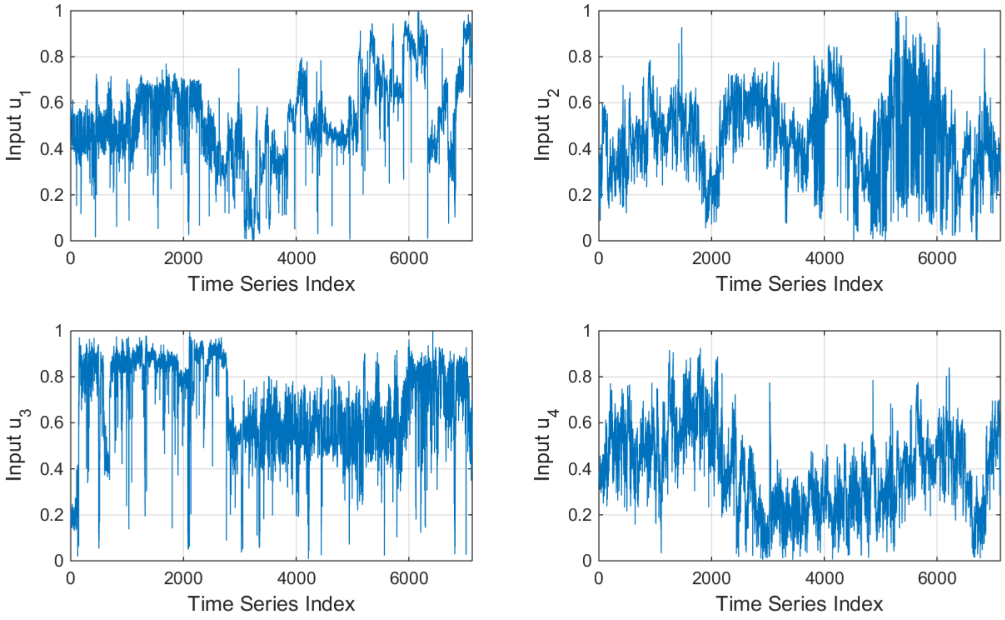

Figure 33 The inputs of the 4-D artificial problem

The outputs for these 7125 data, are still created by an artificial target model function which is the following:

$$y = 10x_1^3 + x_1x_2^2 + 5\sin(5x_3) + x_2x_4 + x_4 + 1.0 + noise$$

where x_1, x_2, x_3, x_4 are the 4 inputs, y is the output, and the noise is random number generated from normal distribution with 0 mean and 0.1 standard deviation.

Parameters selection is performed on this case by varying the parameters and comparing the prediction accuracy. For GP, except the GP parameters are selected by minimize the training data error, number of initial data points used for training is also altered. For PWLRR, number of pieces in each dimension, number of initial data points used for initial training and number of points required for follow-on training in each piece are all varied.

The 7125 data are processed differently for GP and for PWLRR. For GP, the initial set of data with the smallest time indices are used for training the GP model, after the GP model is trained, the data after the initial set are processed one by one, from the smaller time index to larger time index, every data after the initial set are used for prediction, and prediction series is generated.

For PWLRR, the initial set of data with the smallest time indices are used for training the PWLRR model, the data after the initial set are processed one by one, from the smaller time index to larger time index. If a data is located in a trained piece, this data is used to generate prediction, otherwise, this data will be used as follow-on training data. Every data goes to an untrained piece will be considered as training data until this piece is trained, after that, the future coming data to this piece will be used for prediction. And all available predictions are used for prediction series generation.

Blind spot and extent of extrapolation are also computed, the definitions used on this case are same as it is defined in Section 2 and 3 For each GP/PWLRR model, for every available prediction, the absolute prediction error is derived as defined in Section 2, then the mean of all available absolute prediction error is recorded. Figure 34 shows the percentage of blind spot, extent of extrapolation and average absolute prediction error of each GP/PWLRR model.

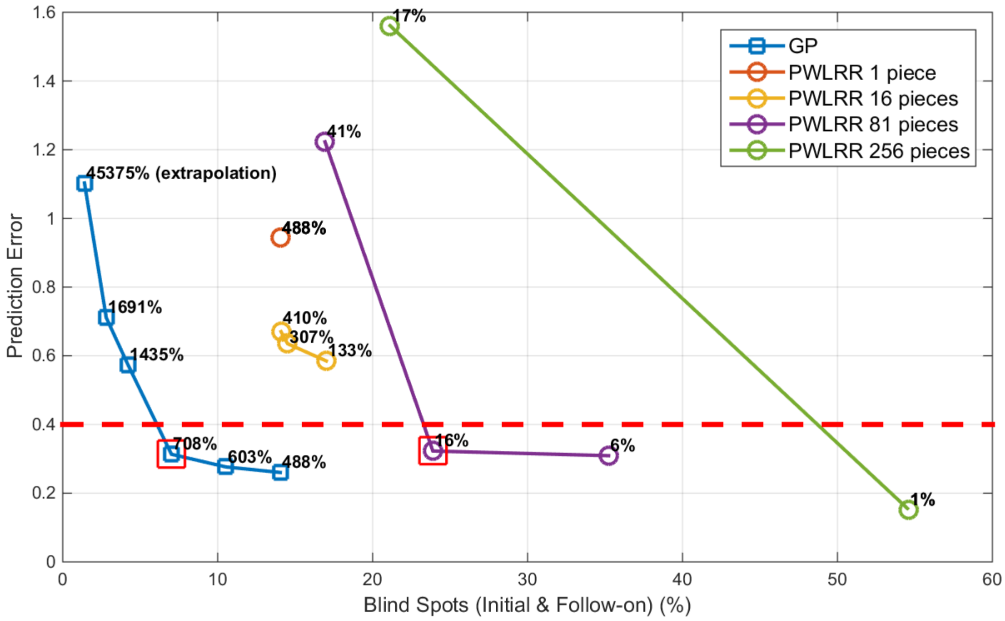

Figure 34 Blind spots vs PE of GP and PWLRR

It can be observed from the results that when more data are used for training, there will be more blind spot, less extrapolation, but also less prediction error. So generally, prediction error and blind spot / extent of extrapolation are tradeoffs. In a real fault detection system, the goal of model training is to acquire certain level of prediction error. For this artificial case, given a certain prediction error level as the goal. It can be observed that the leaning by GP is always needing less training data than learning by PWLRR, there model by GP has better performance in blind spots and extent of extrapolation. Figure 35 shows two prediction error series generated by GP model and PWLRR model, and when the prediction series have comparable prediction error, PWLRR model is having more blind spot data. Figure 36 also shows prediction series generated by GP model and PWLRR model, and this PWLRR model can achieve no blind spot after initial training similar as the GP model, but the prediction error of the PWLRR model is obviously worse than the GP model as the PWLRR prediction error series are more spread from the target which is zero for this case.

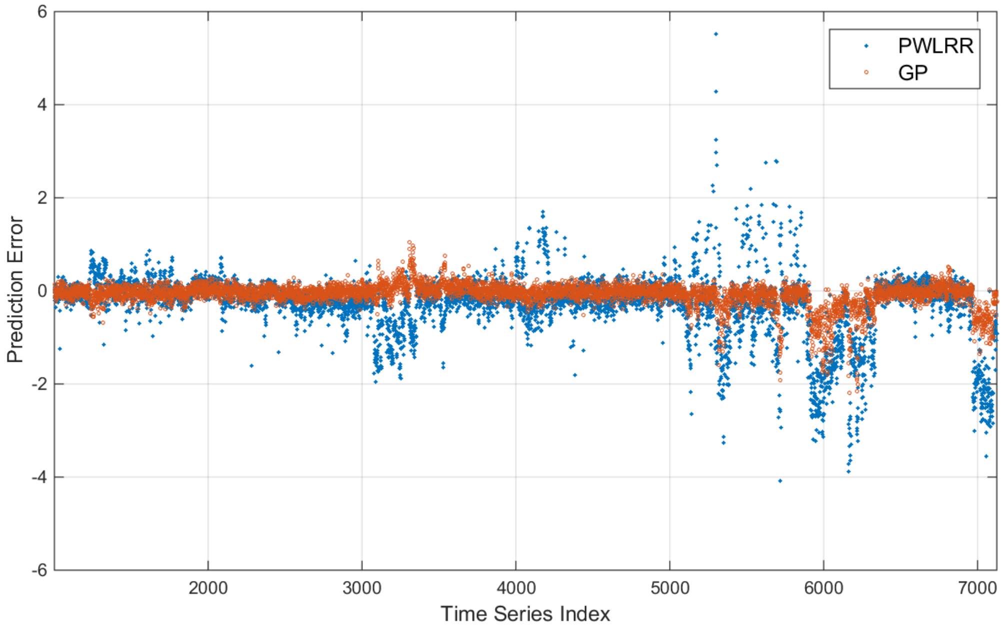

Figure 35 PE comparison when blind spots are similar

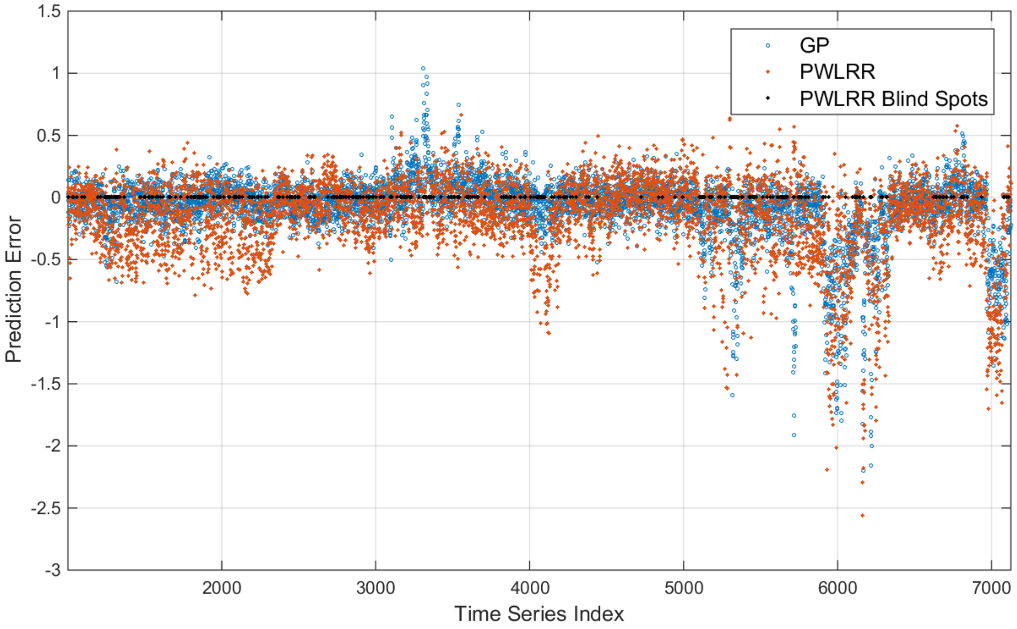

Figure 36 Blind spots comparison when PE are simila

Assume the objective of the average prediction error is less than 0.4 for this case, one solution from GP and one from PWLRR are selected as indicated in Figure 36 And their performances are compared in Table 2 The GP is performing much better than PWLRR.

	PWLRR	GP	Notes
PE	0.4	0.4	Absolute Error
Data for Training	3.5x	1x	Relative data use
Extent of Extrapolation	16%	700%	Beyond the training set

Table 2 Performance comparison for artificial 4-D Case

4.1.5 4-D Inputs with Fault Simulation

Since the prediction error can always be lowered by increasing the training data, as demonstrated in the precious case, PWLRR may potentially achieve lower prediction error than GP if both the initial and follow-on training are allowed to have more training data. However, letting more data to be used as training data, especially the data collected in the later follow-on stage, has a risk to impact the effectiveness of the fault detection system when there is healthy situation developing. This last artificial case is used to demonstrate the risk of the follow-on training which has to exist in the PWLRR method.

This case with fault simulation is based on the previous 4-D case with static model, the number of data is same, the 4-dimensional inputs of the data are same, also the time index associated with each data is same. The target model function for this case is different, while keeping the target model function from the previous case, which only includes the terms associated with 4 dimensional inputs, constant and noise, one additional term which contains the time index is added to the target model function:

$$y = 10x_1^3 + x_1x_2^2 + 5\sin(5x_3) + x_2x_4 + x_4 + 1.0 + noise + f(t)$$

$$f(t) = 3\left(max\left(0, \frac{t-720}{6495}\right)\right)^2$$

where x_1, x_2, x_3, x_4, y and noise are having the same definition as the previous case, and t is the time index ranged from 1 to 7215.

The time index term is used to simulate the fault signature in the output when there is a fault developing. Usually, the health of the asset is getting worse as time goes, therefore the fault signature is getting stronger. The same solutions from GP and PWLRR selected in the previous case are also selected in the case. Figure 37 shows the GP and PWLRR prediction error series when there is no fault simulated, it is same as Figure 36 except a moving average window is applied to the series to get a clearer view.

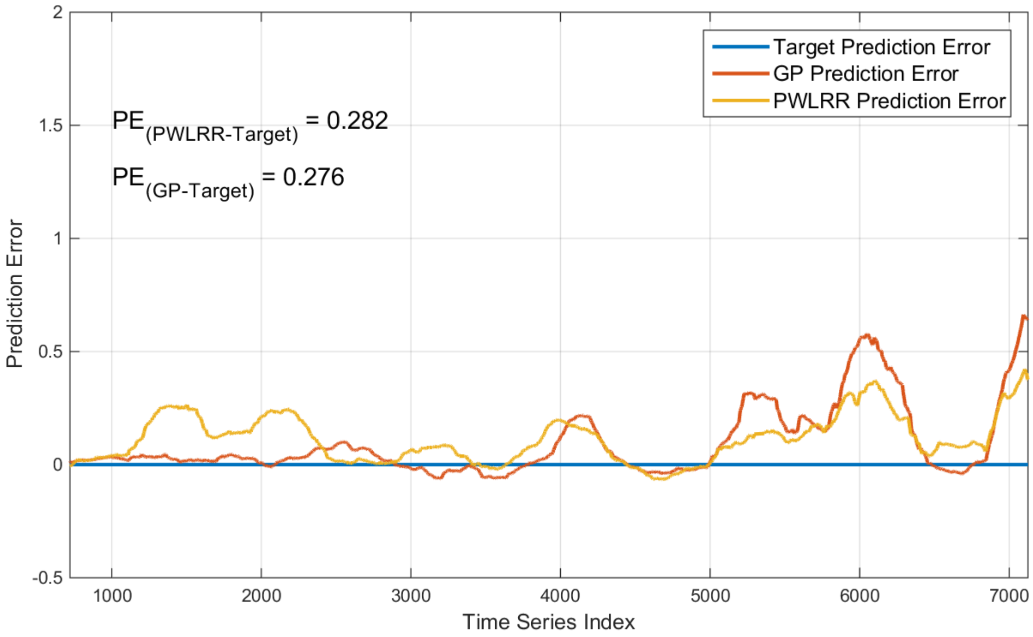

Figure 37 Smoothed PE without simulated fault

Then the exact same parameters, training and prediction processes are applied to the data with fault simulation in this case. Figure 38 shows the raw prediction error series from GP and PWLRR model. Figure 39 shows the prediction error series from GP and PWLRR model with a moving average window and the target prediction error which is the term f(t).

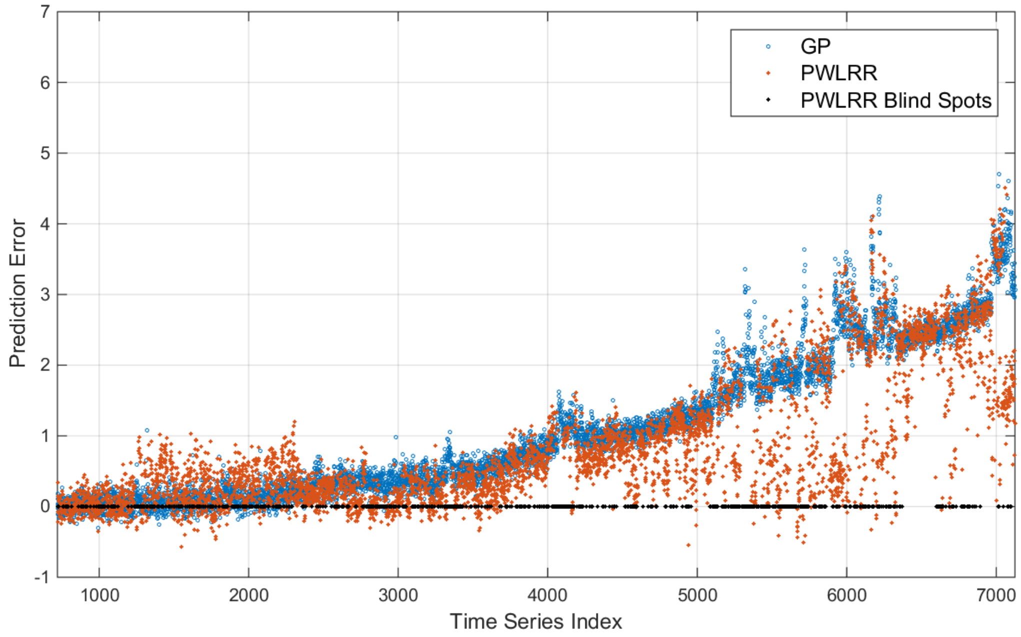

Figure 38 PE with fault simulated

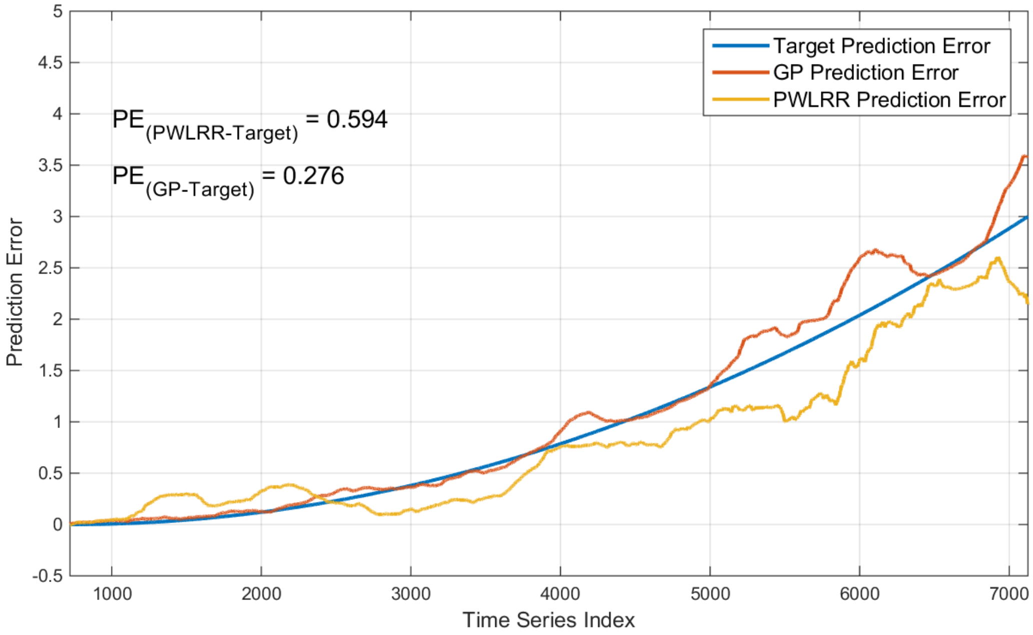

Figure 39 Smoothed PE with fault simulated

The figures show that while GP model can achieve similar performance in tracking the target error function with and without fault simulation, the PWLRR model is performing worse when there is developing fault than the situation of no fault. The reason is when this case is created with only the initial 10 percent of data are having no fault signature ($f(t)$ is zero), GP model can be trained with only healthy data, so it can capture all fault signature as desired. While PWLRR model must be trained with more initial and follow-on data which contains some developing fault signature, so the PWLRR model can only capture the fault signature difference between the training data and the prediction data, which is weaker than the target fault signature simulated.

This risk of using more data in initial and follow-on training will result in delay in the fault detection.

4.2 Real Cases

4.2.1 Fault Detection System Setup

The fault detection system is consisted by a data acquisition component which collects voltage and current measurements, a signal processing component which processes the measurements to generate five inputs one output time series data, and a final fault detection component which includes the model training, prediction generation and the fault decision logic based on prediction error vs time. Figure 40 shows an example of the complete fault detection system by ESA and an asset.

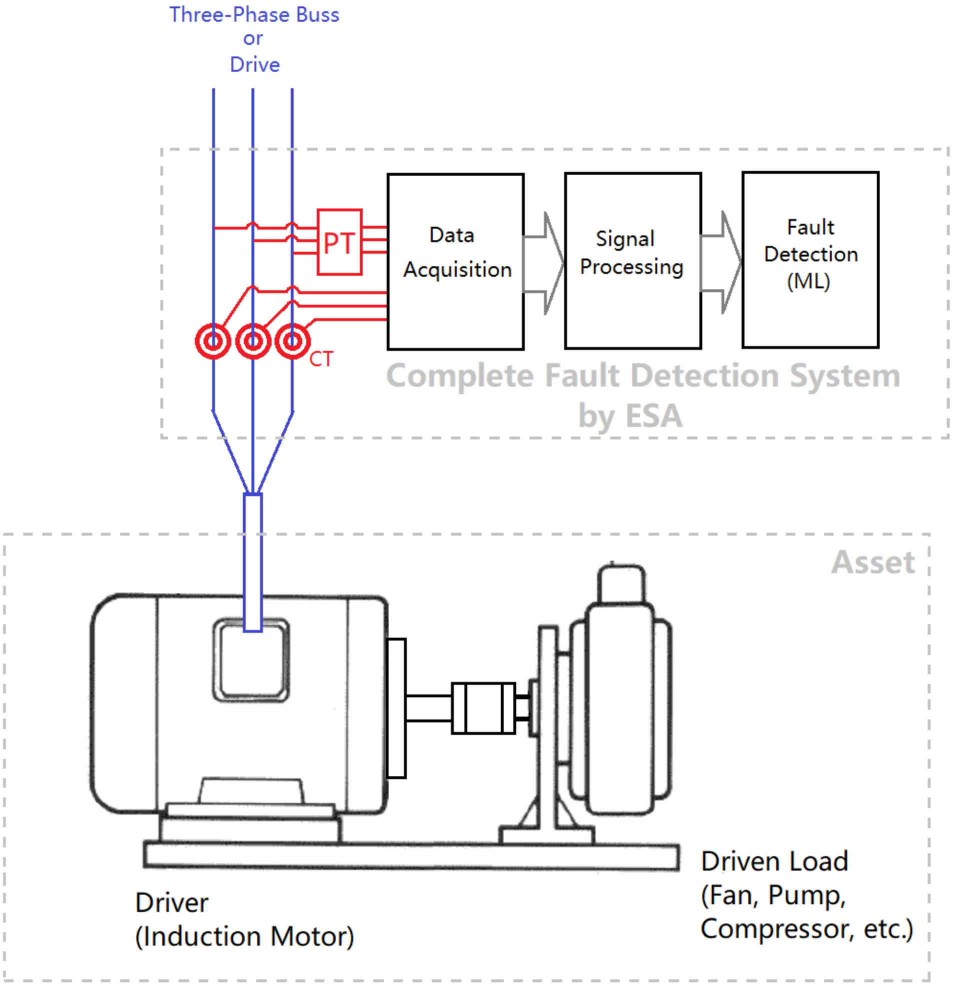

Figure 40 The complete fault detection system by ESA

4.2.2 Data Preparation and List of Cases

Both voltage and current measurements had been collected and processed to get the inputs-output time series, the details of the first two components of the fault detection system shown in the Figure 40 are not the major focus of this research. All of time series data are stamped with a relative timestamp, with the first data stamped as t = 0.

Real cases investigated in this research can be grouped into groups: cases with data collected from the assets which are tested in a lab environment and cases with data collected from the production site assets. Only the mechanical faults are investigated, both cases with and without fault are tested, for the cases with fault, maximizing the true positives and detection is the primary goal, and for the cases without fault, minimizing the false positives is important.

The cases are selected to cover a variety of voltage and load levels, driven load types, power sources (Utility Buss or Variable Frequency Drive) and fault types. Table 3 lists the detailed information of these real cases. There are 6 cases with fault and 4 cases without.

Case ID	Data Source	Asset Type	Power Source	Nameplate	Fault

#1	Production Site	ESP	VFD	Rated Voltage: 4160 volts;
				Rated Current: 233 amps;
				Rated Load: 1500 hp;
				Rated Speed: 3490 rpm
					Pump Failure

#2	Production Site	ESP	VFD	Rated Voltage: 4160 volts;
				Rated Current: 233 amps;
				Rated Load: 1500 hp;
				Rated Speed: 3490 rpm
					Pump Failure

#3	Production Site	Reciprocating Compressor	Buss	Rated Voltage: 4160 volts;
				Rated Current: 472 amps;
				Rated Load: 3750 hp;
				Rated Speed: 1193 rpm
					Driven Load Bearing

#4	Lab	ESP	VFD	Rated Voltage: 1157 volts;
				Rated Current: 42.3 amps;
				Rated Load: 75 hp;
				Rated Speed: 3377 rpm
					Driven Load Sand Corrosion

#5	Lab	ESP	VFD	Rated Voltage: 460 volts;
				Rated Current: 263 amps;
				Rated Load: 250 hp;
				Rated Speed: 3570 rpm
					Driven Load Sand Corrosion

#6	Lab	Pump	Buss	Rated Voltage: 400 volts;
				Rated Current: 27.8 amps;
				Rated Load: 20 hp;
				Rated Speed: 1474 rpm
					Driven Load Bearing Current Injection

#7	Production Site	ESP	VFD	Rated Voltage: 4160 volts;
				Rated Current: 233 amps;
				Rated Load: 1500 hp;
				Rated Speed: 3490 rpm
					N/A

#8	Production Site	Reciprocating Compressor	Buss	Rated Voltage: 4160 volts;
				Rated Current: 203 amps;
				Rated Load: 1650 hp;
				Rated Speed: 1188 rpm
					N/A

#9	Production Site	Pump	Buss	Rated Voltage: 4160 volts;
				Rated Current: 213 amps;
				Rated Load: 1500 hp;
				Rated Speed: 445 rpm
					N/A

#10	Production Site	Pump	Buss	Rated Voltage: 4160 volts;
				Rated Current: 125 amps;
				Rated Load: 1000 hp;
				Rated Speed: 1780 rpm
					N/A

Table 3 List of real cases

4.2.3 Results of Real Cases

For each individual case, the plots of prediction error / threshold vs time will be compared between GP model and PWLRR model. The true positives and false positives can be reviewed from the plots. Other performance indicators defined in Section 3 will also be computed and results will be summarized at the end of this section.

a) Cases #1, #2, #3

The data for these three cases are collected from the assets on production sites. The duration of the data is from 4 to 10 months. All assets ran into failure at the end of the data period.

Figure 41 and Figure 42 shows prediction error / threshold vs time of the PWLPP and GP model on Case #1. GP model successfully detects the fault while PWLRR failed in detection.

Figure 43 and Figure 44 shows prediction error / threshold vs time of the PWLPP and GP model on Case #2. Both GP model and PWLRR model detected the fault, while GP model achieves higher detection time.

Figure 45 and Figure 46 shows prediction error / threshold vs time of the PWLPP and GP model on Case #3. Both GP model and PWLRR model detected the fault, while GP model achieves higher detection time.

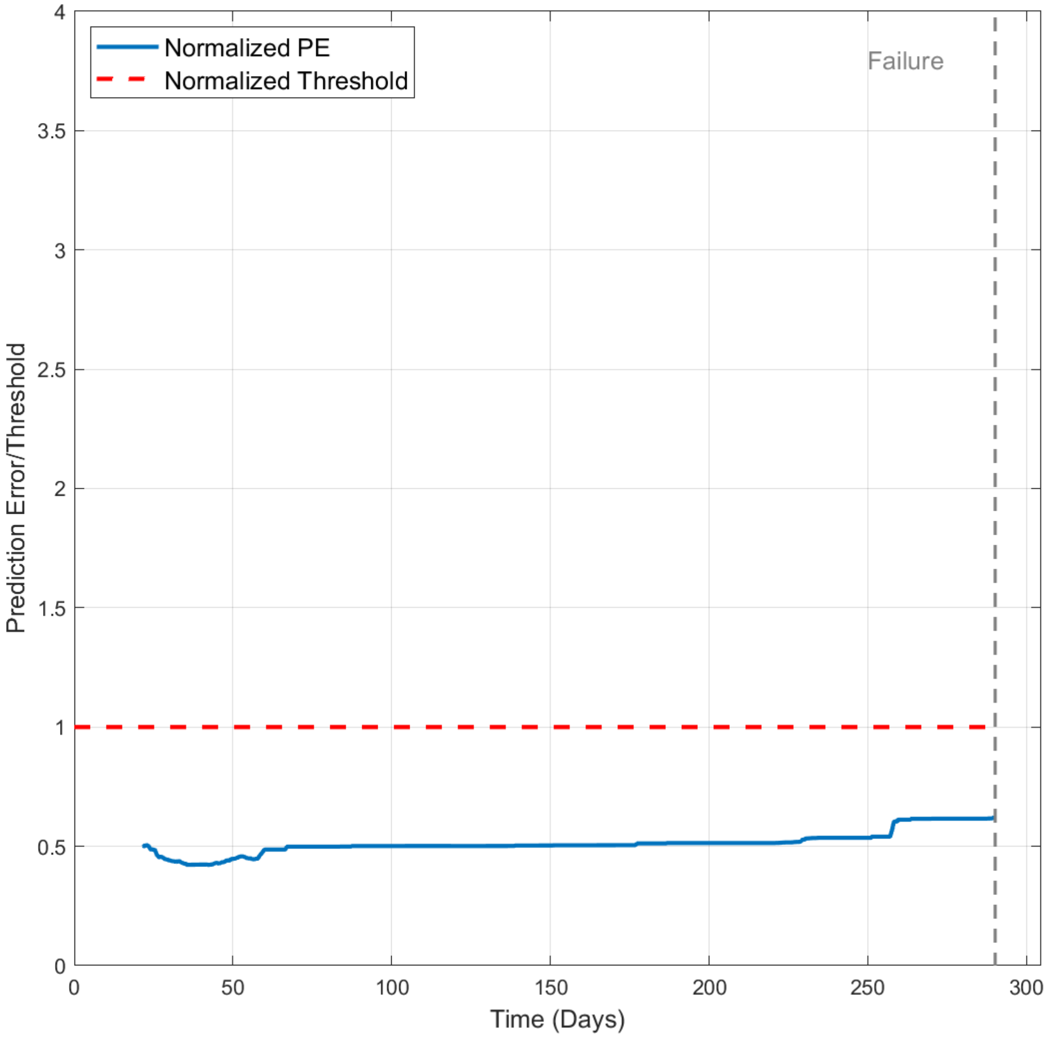

Figure 41 Case#1 PE of PWLRR

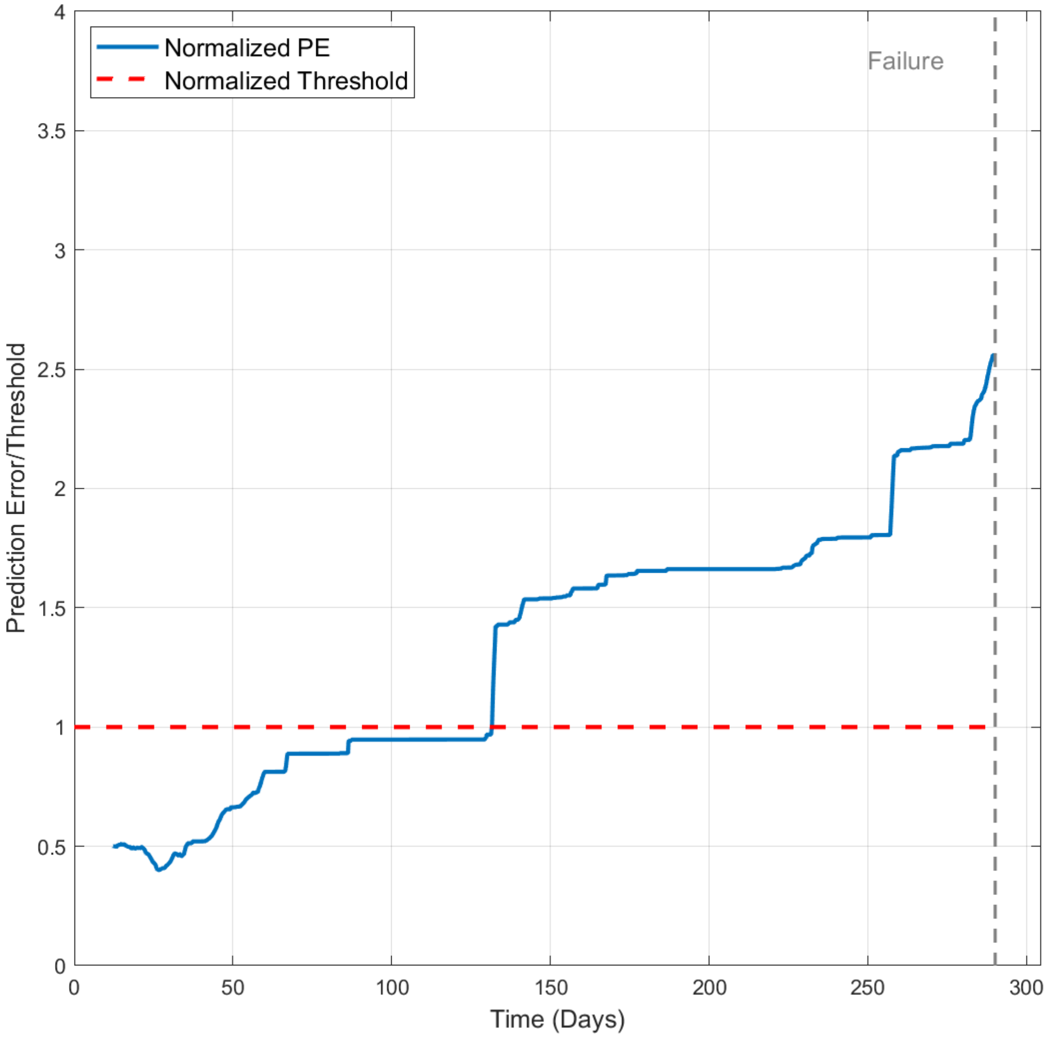

Figure 42 Case#1 PE of GP

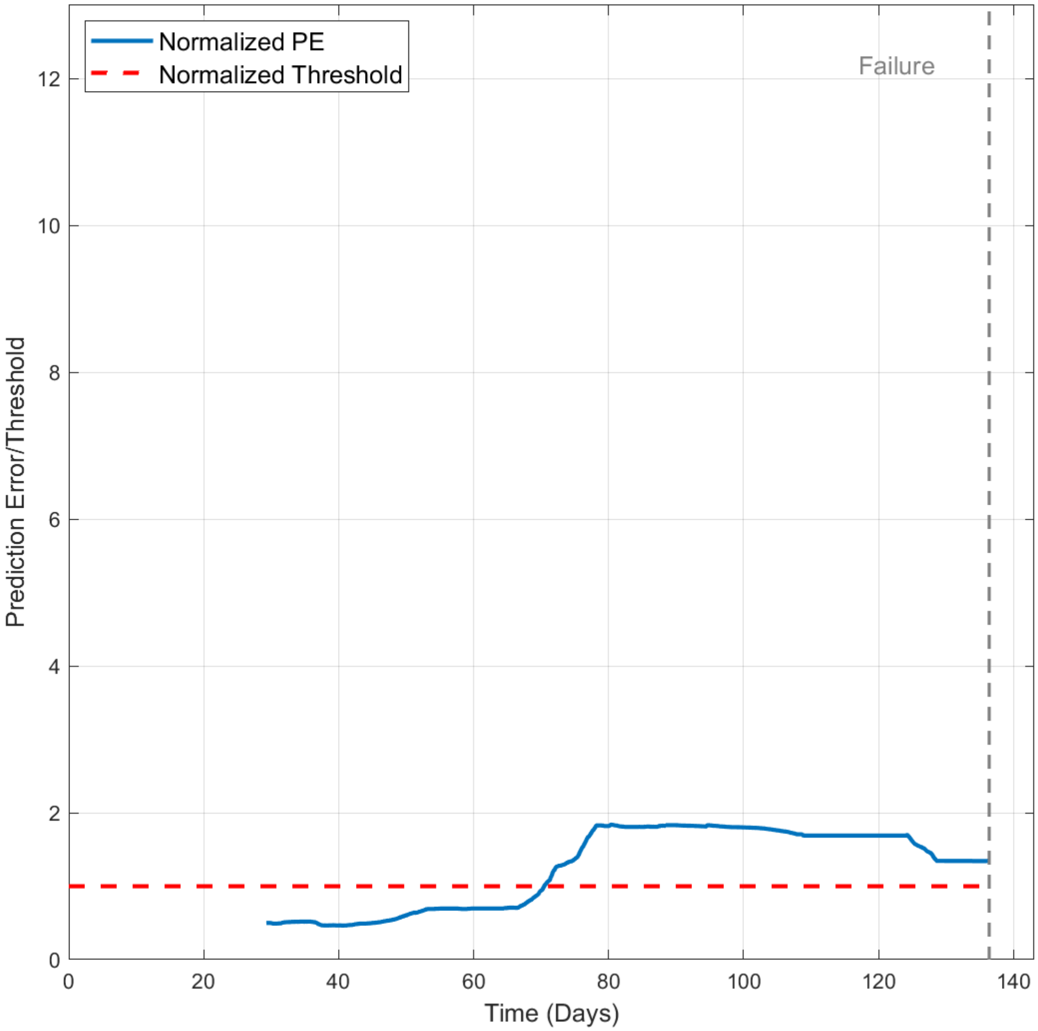

Figure 43 Case#2 PE of PWLRR

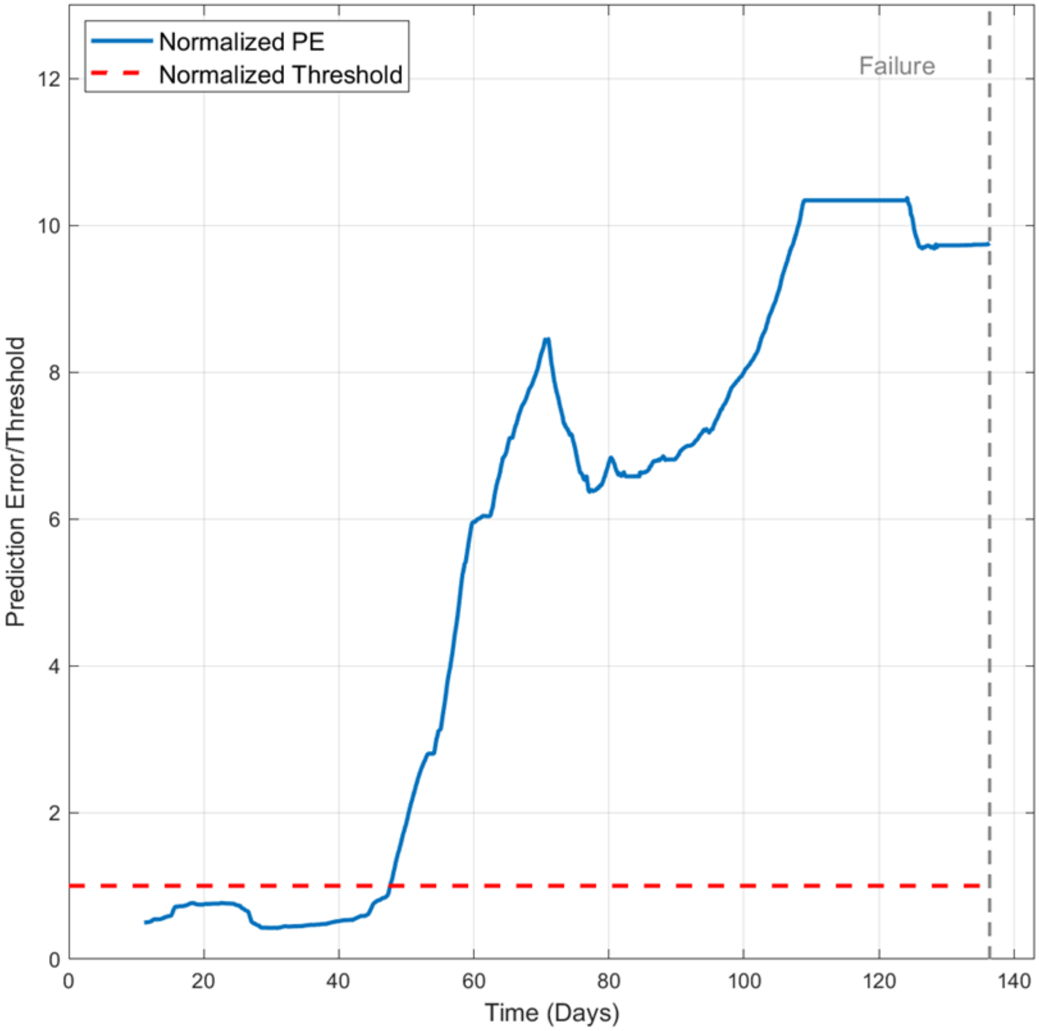

Figure 44 Case#2 PE of GP

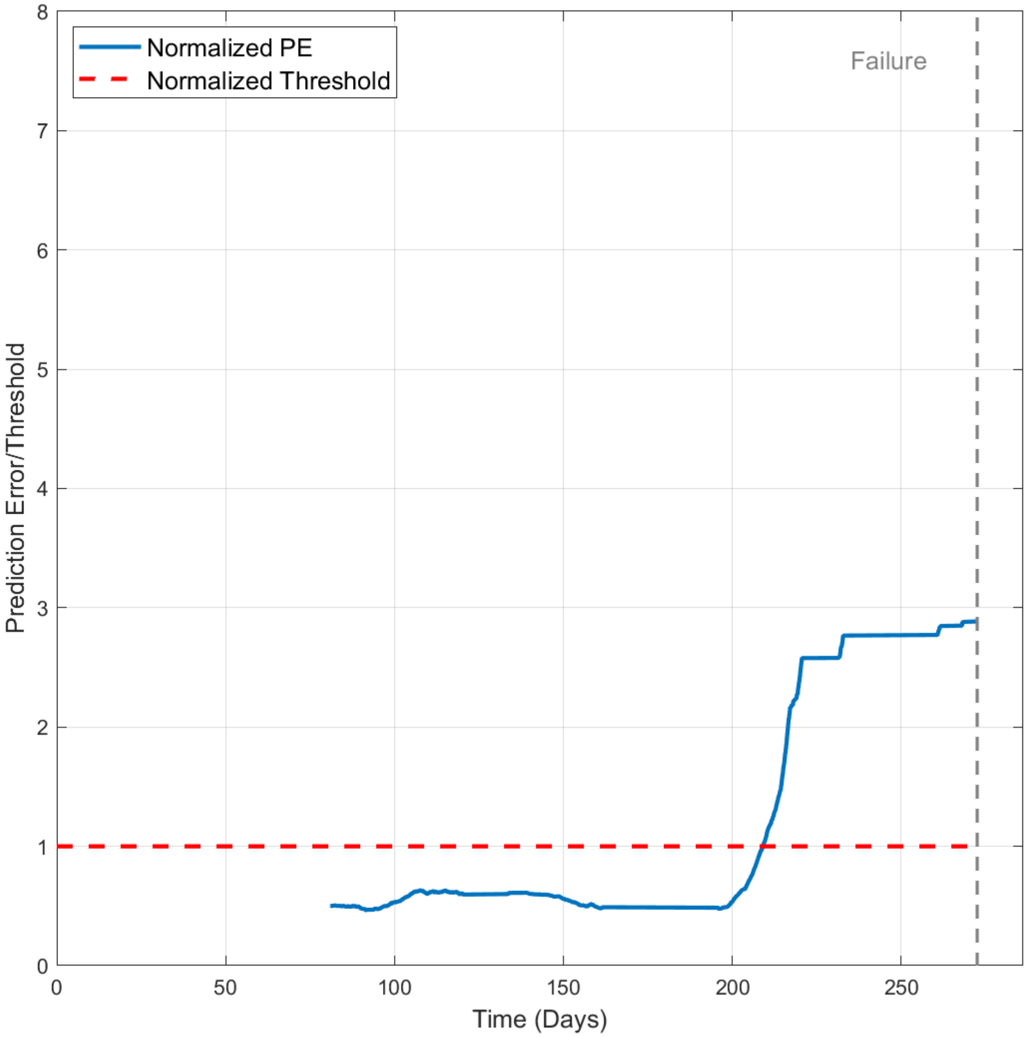

Figure 45 Case#3 PE of PWLRR

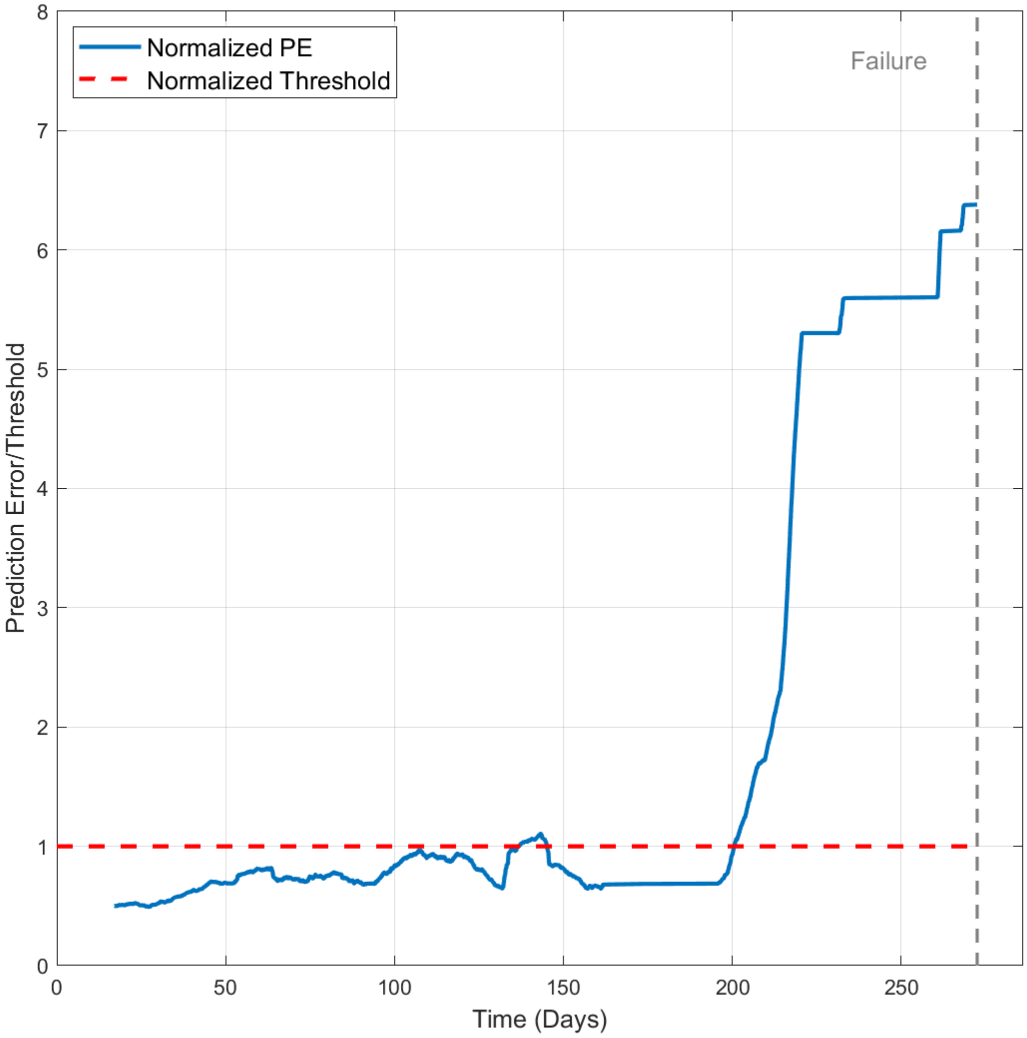

Figure 46 Case#3 PE of GP

b) Cases #4, #5, #6

The data for these three cases are collected from the assets in Lab. The duration of the data is less than 1.5 months with Case #5 only lasts from about 60 hours. The faults were introduced to the assets by artificial means.

Figure 47 and Figure 48 shows prediction error / threshold vs time of the PWLPP and GP model on Case #4. Both GP model and PWLRR model detected the fault, while GP model achieves higher detection time.

Figure 49 and Figure 50 shows prediction error / threshold vs time of the PWLPP and GP model on Case #5. GP model successfully detects the fault while PWLRR failed in detection. The data of this case are very few, the PWLRR cannot even perform effective prediction in the later stage of the test.

Figure 51 and Figure 52 shows prediction error / threshold vs time of the PWLPP and GP model on Case #6. GP model successfully detects the fault while PWLRR failed in detection.

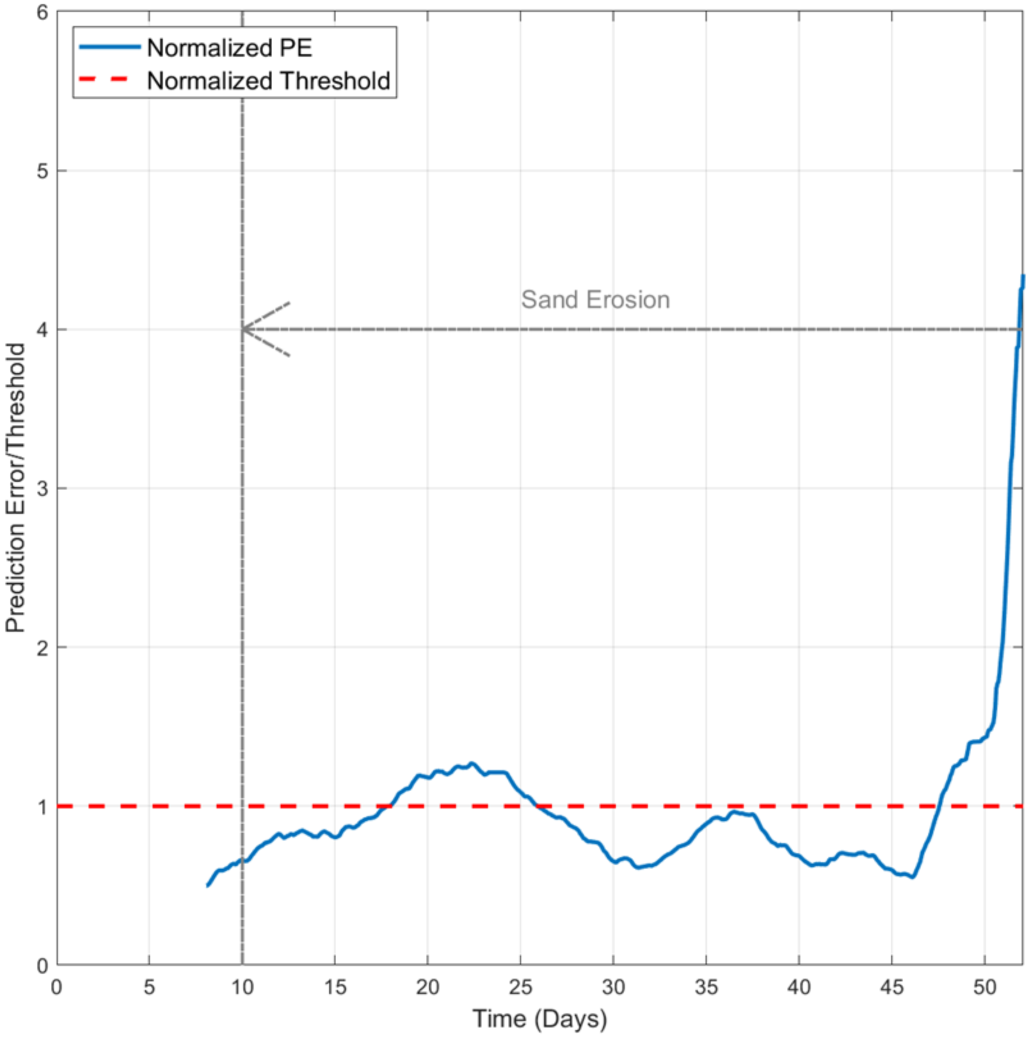

Figure 47 Case#4 PE of PWLRR

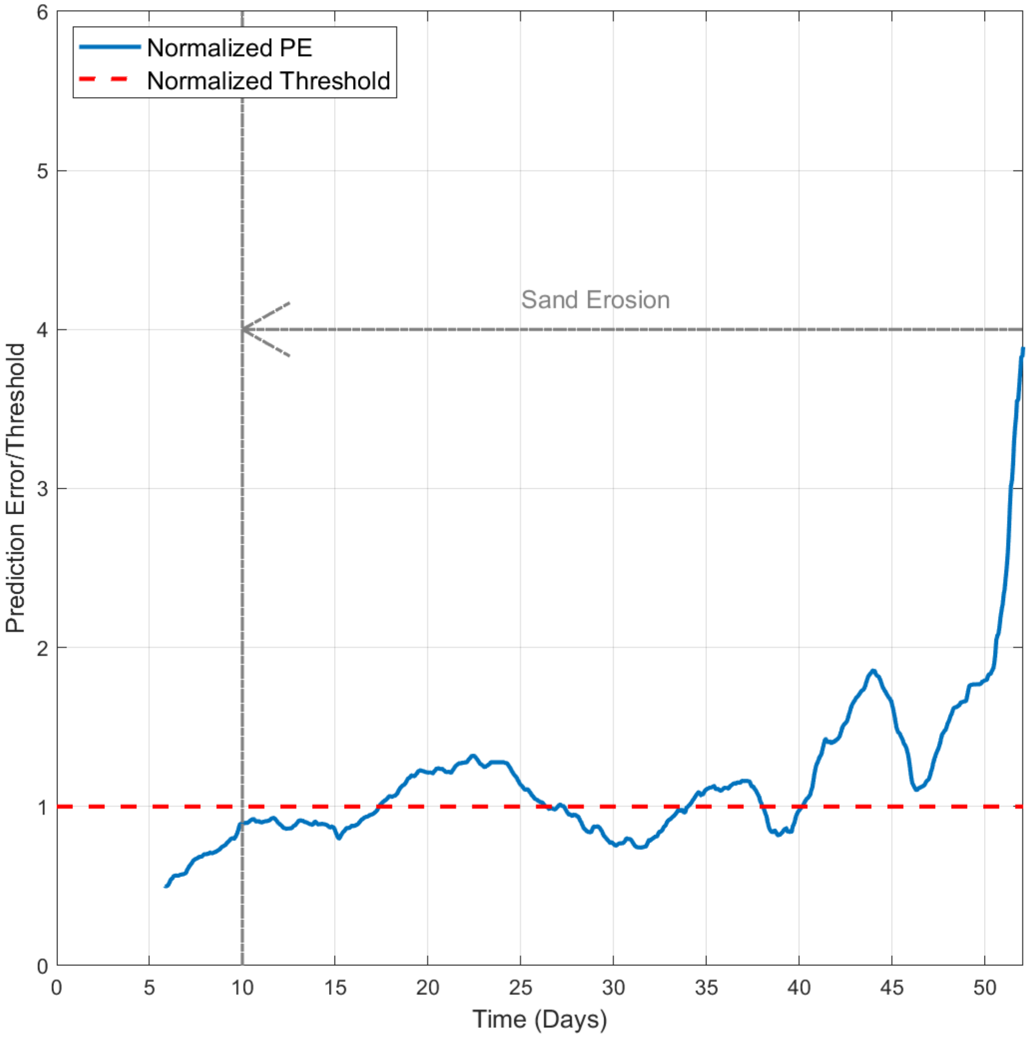

Figure 48 Case#4 PE of GP

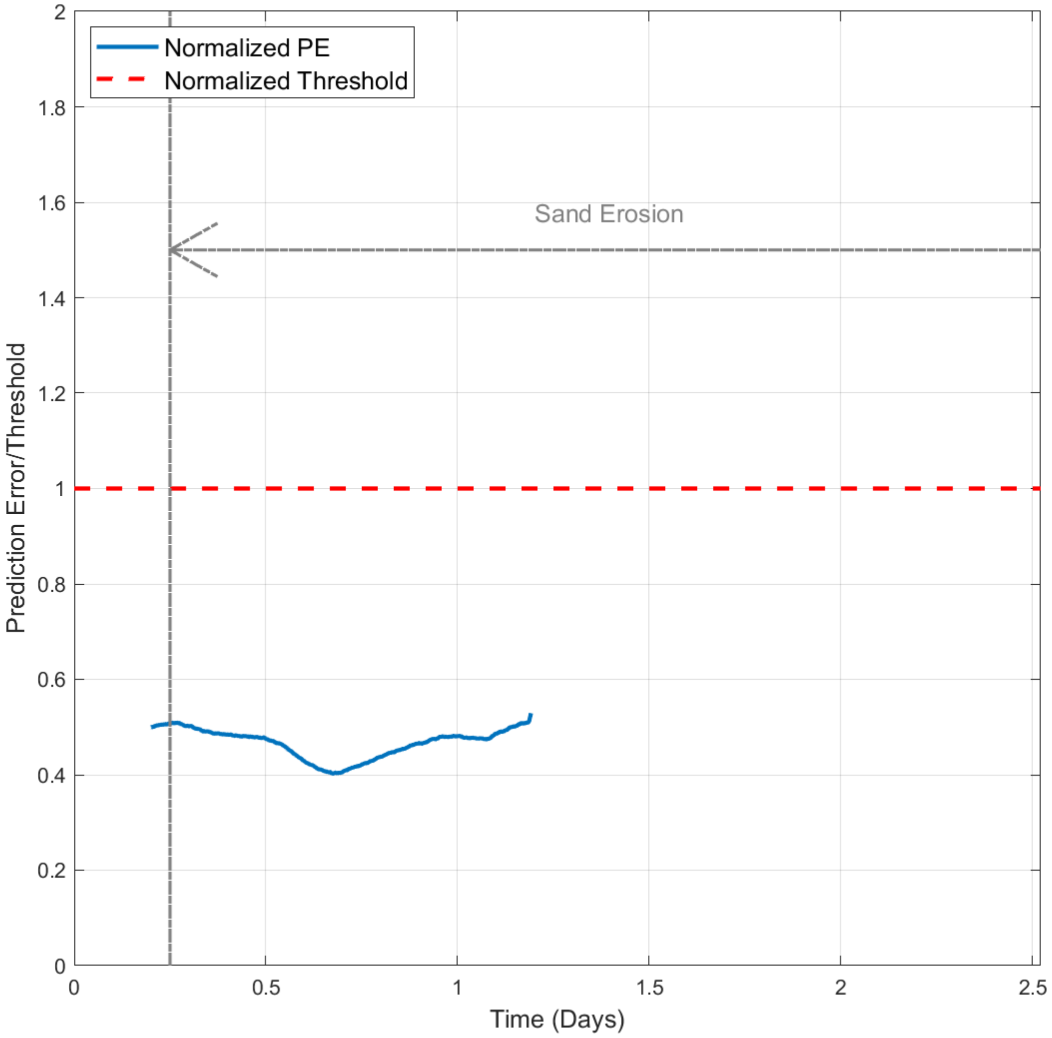

Figure 49 Case#5 PE of PWLRR

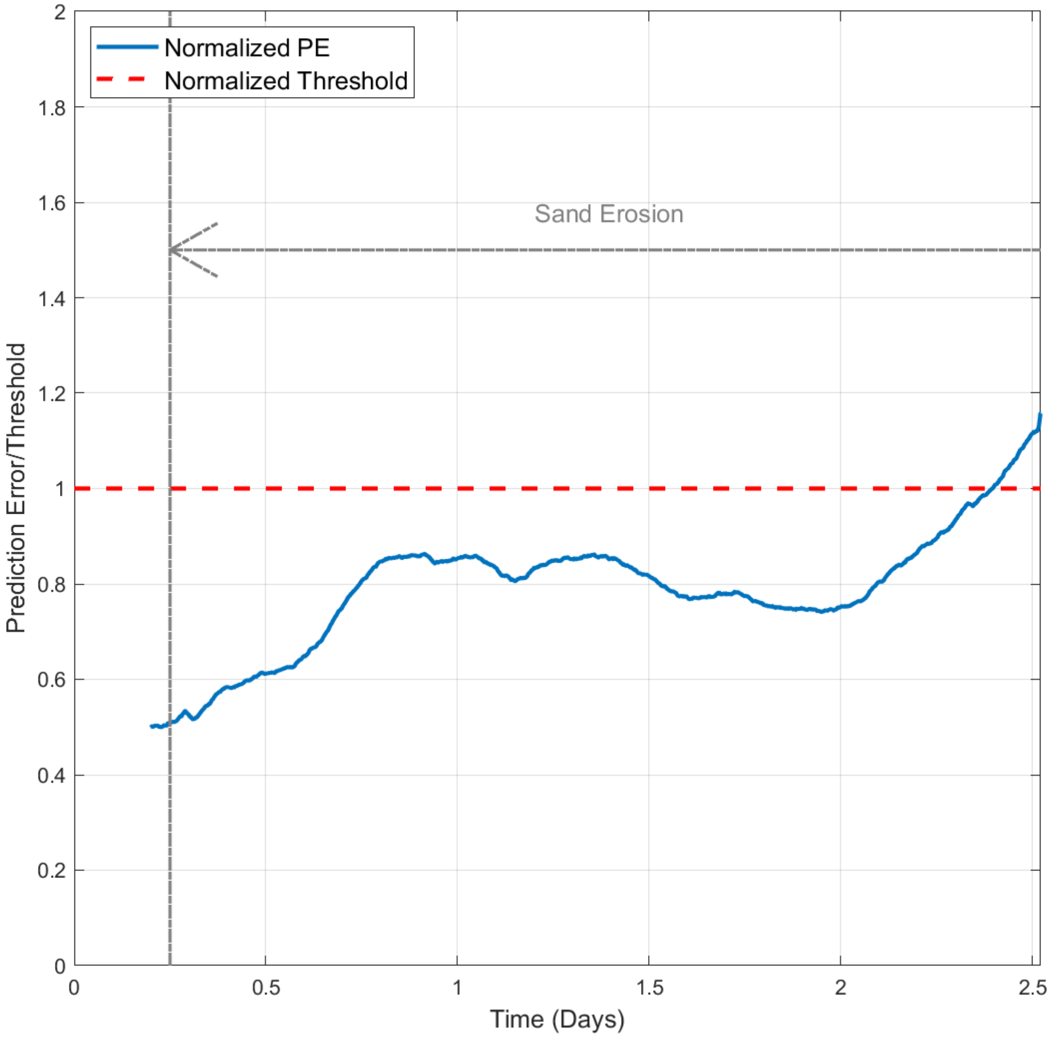

Figure 50 Case#5 PE of GP

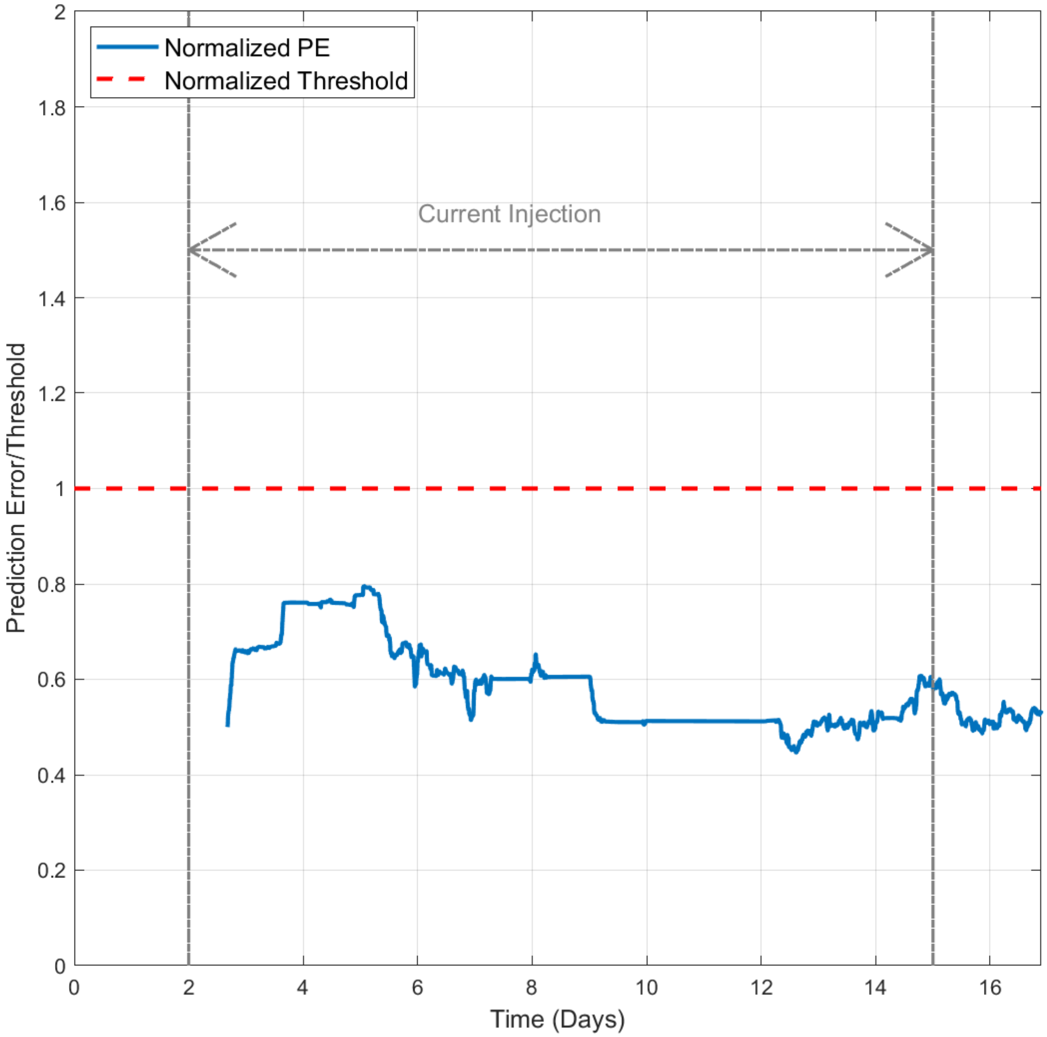

Figure 51 Case#6 PE of PWLRR

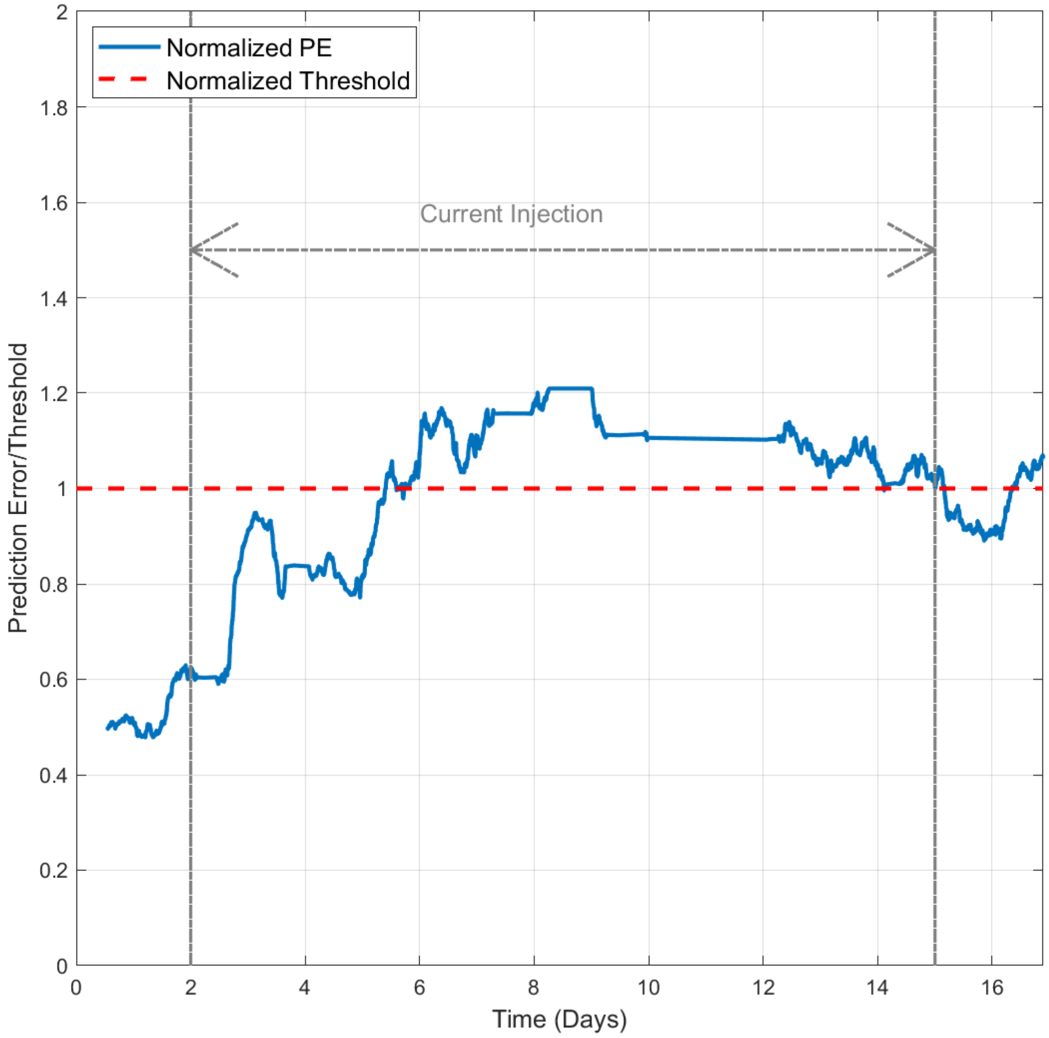

Figure 52 Case#6 PE of GP

c) Cases #7, #8, #9, #10

The data for these three cases are collected from the assets on production sites. The duration of the data is from 8 to 16 months. All assets were running normal during and after the data period.

Figure 53 and Figure 54 shows prediction error / threshold vs time of the PWLPP and GP model on Case #7. Both GP model and PWLRR model have no false alarm during the period

Figure 55 and Figure 56 shows prediction error / threshold vs time of the PWLPP and GP model on Case #8. GP model has no false alarm while prediction error from the PWLRR model goes above the threshold twice, resulting two false positives

Figure 57 and Figure 58 shows prediction error / threshold vs time of the PWLPP and GP model on Case #9. GP model has no false alarm while prediction error from the PWLRR model has one false positive.

Figure 59 and Figure 60 shows prediction error / threshold vs time of the PWLPP and GP model on Case #10. Both GP model and PWLRR model have no false alarm during the period.

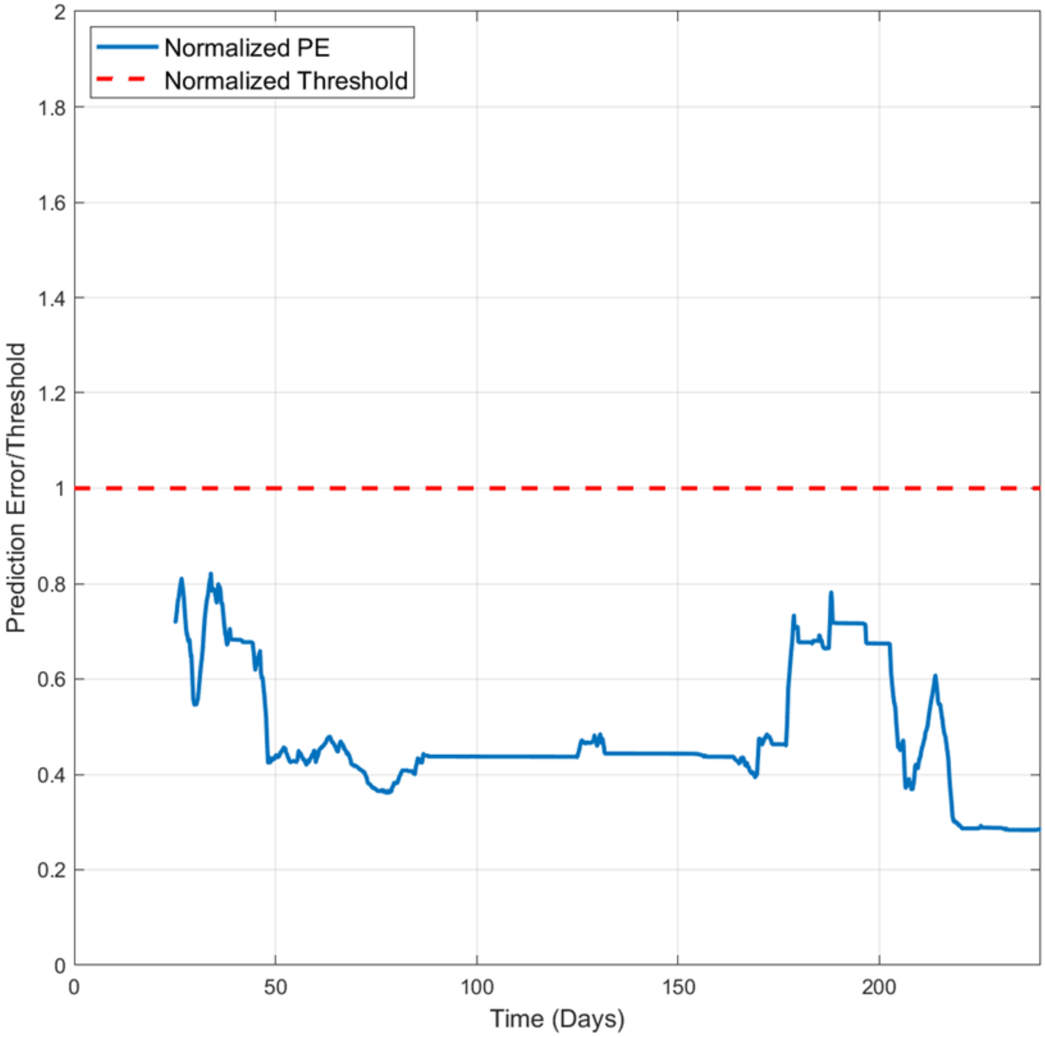

Figure 53 Case#7 PE of PWLRR

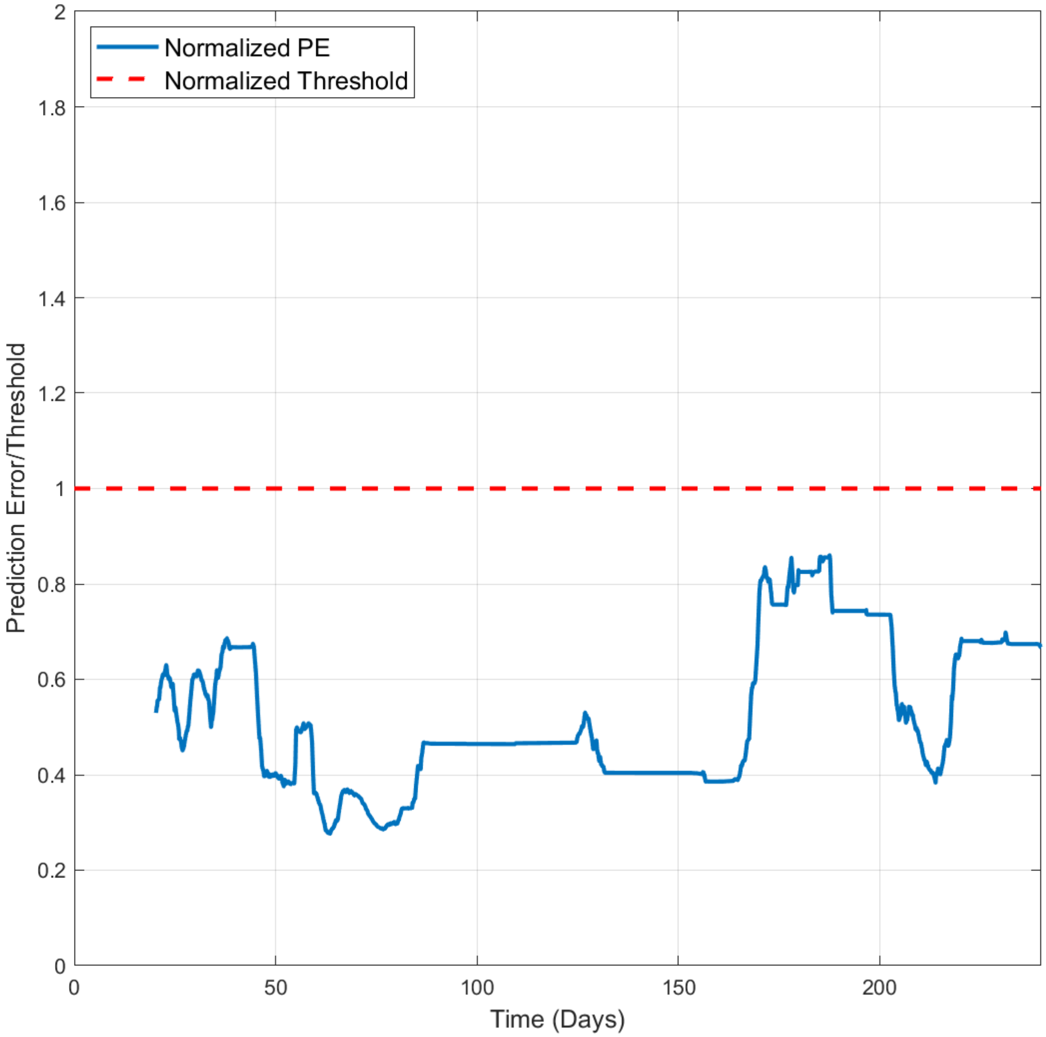

Figure 54 Case#7 PE of GP

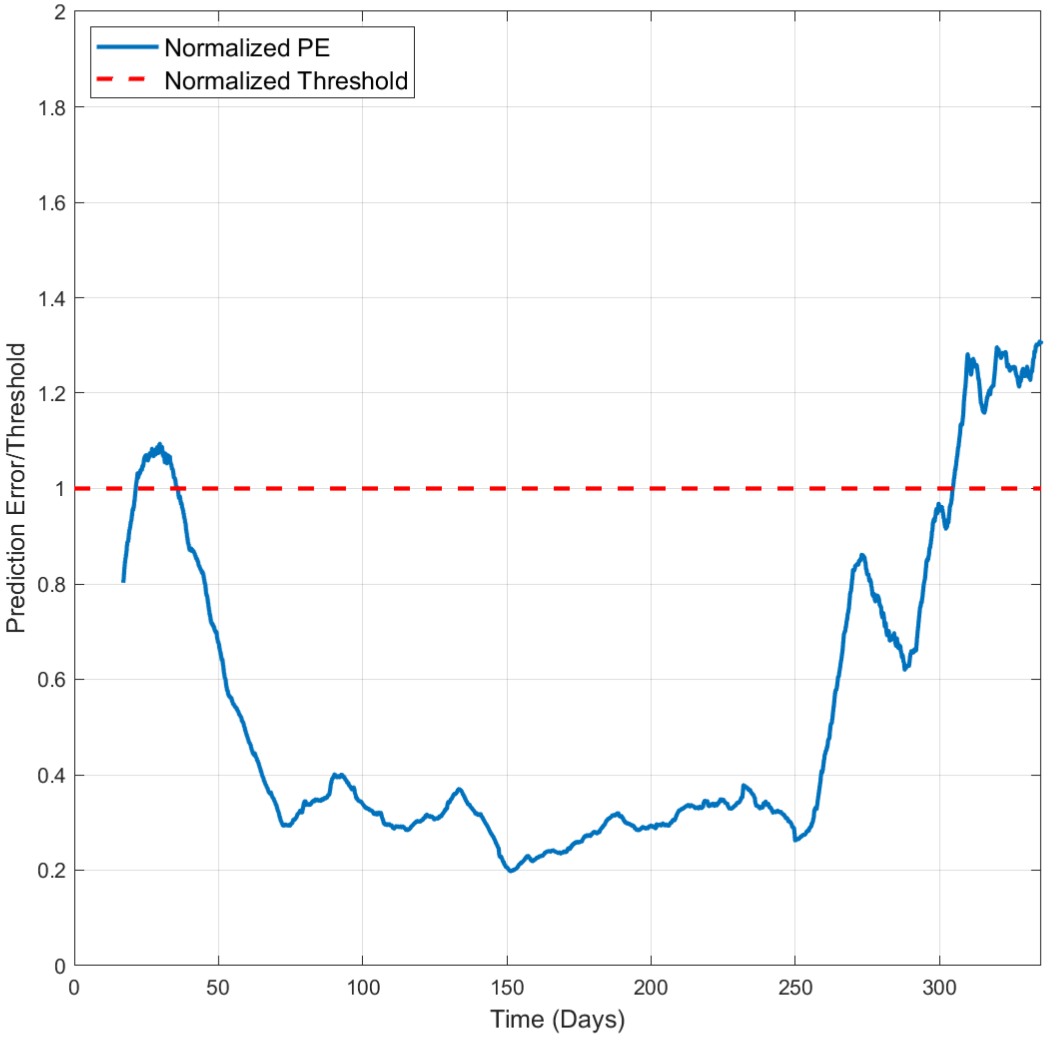

Figure 55 Case#8 PE of PWLRR

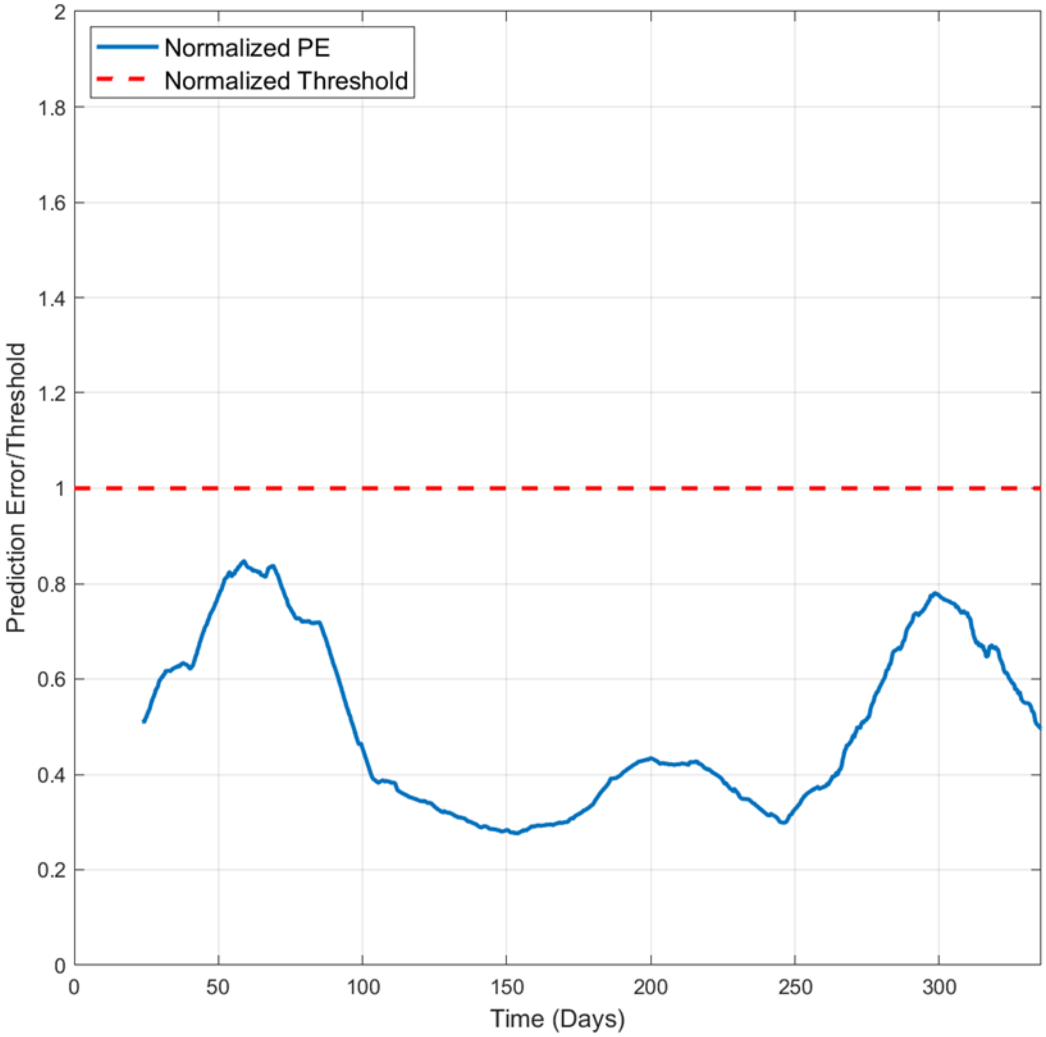

Figure 56 Case#8 PE of GP

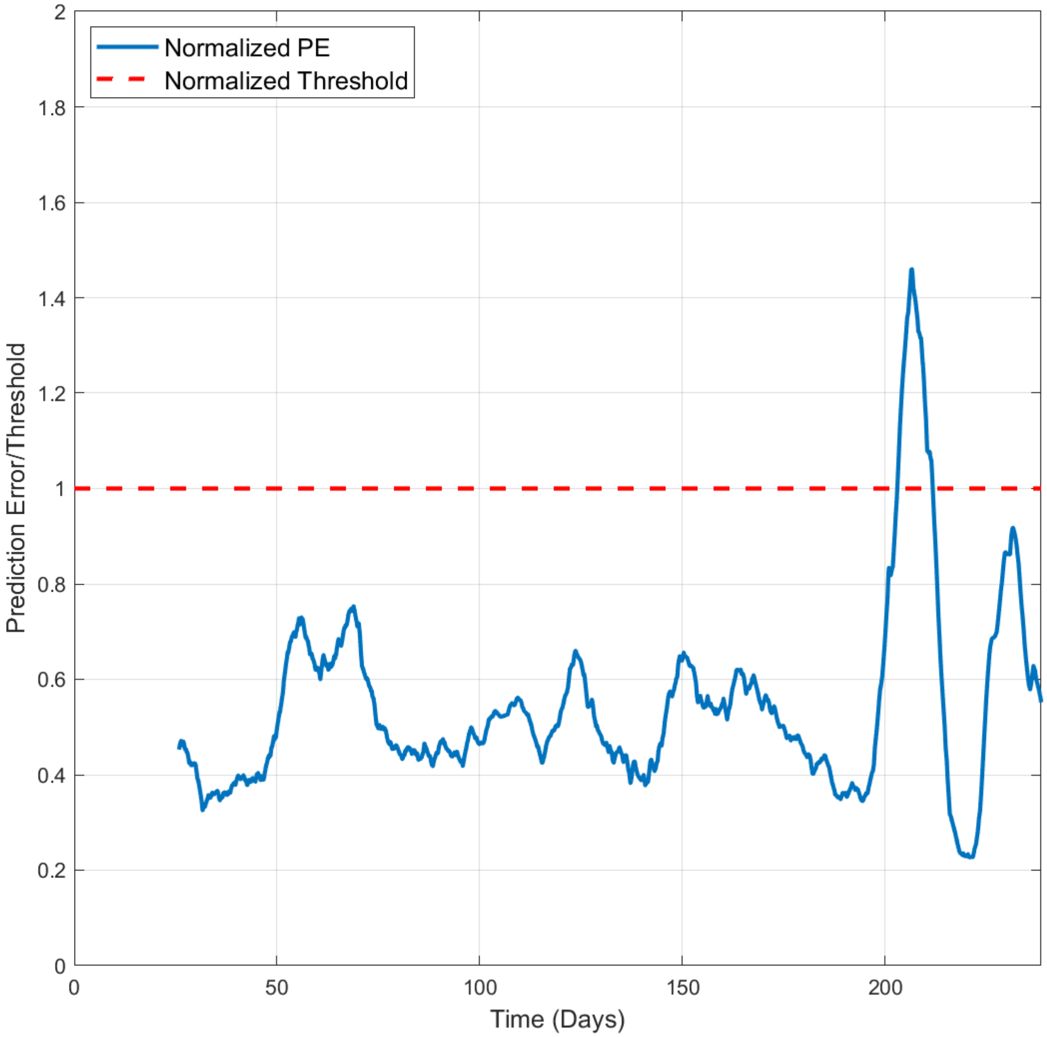

Figure 57 Case#9 PE of PWLRR

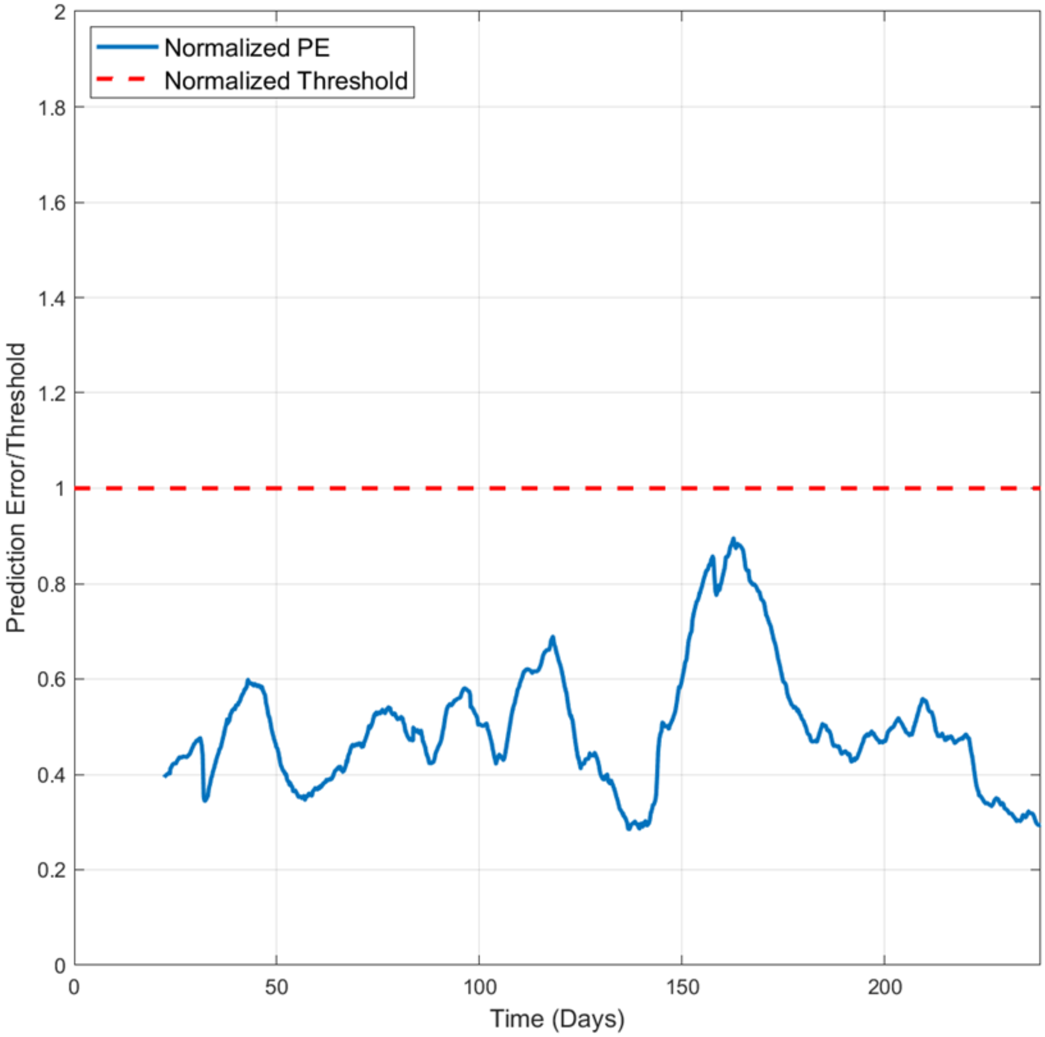

Figure 58 Case#9 PE of GP

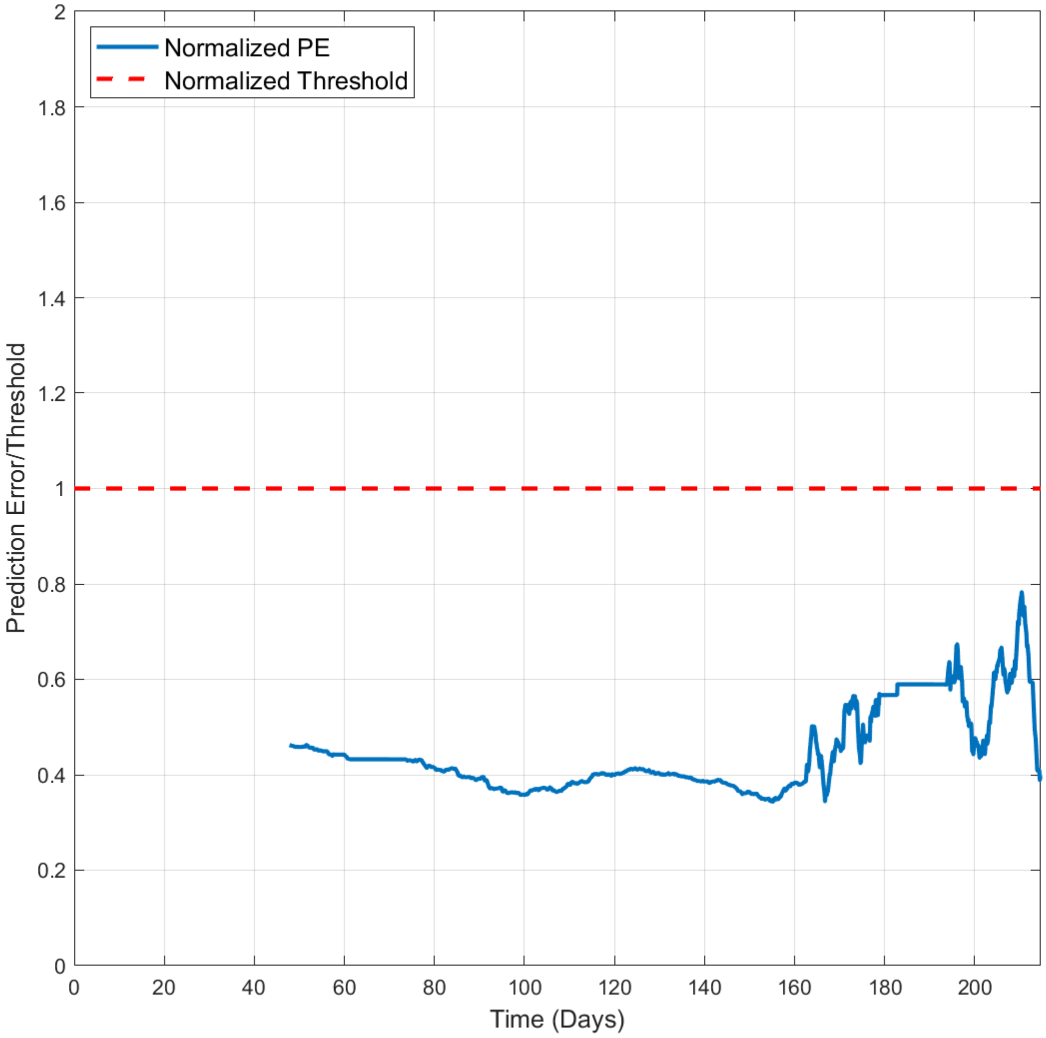

Figure 59 Case#10 PE of PWLRR

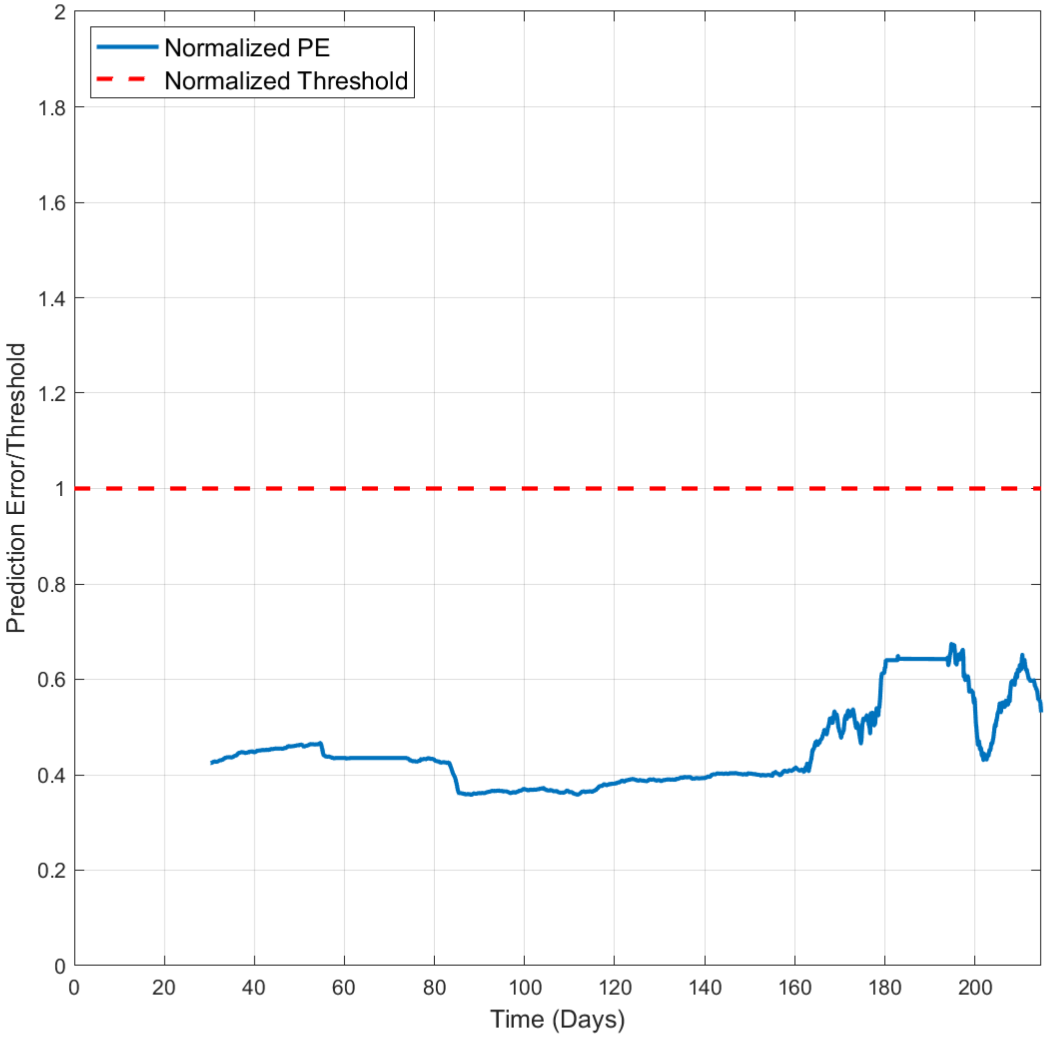

Figure 60 Case#10 PE of GP

d) Summary of the Results

The number of training and validation points, percentage of blind spot, extent of extrapolation, detection time, true positives and testing prediction error average are compared between the GP and PWLRR methods for cases with fault (#1 to #6). And The number of training and validation points, percentage of blind spot, extent of extrapolation, false positives and testing prediction error average are compared between the GP and PWLRR methods for cases without fault (#7 to #10). The aggregates of these performance indicators are also computed, when aggregating the results, the cases are separated into two groups: group with data from production site and group with data from lab. Since the data from lab are collected in the better controlled environment, the variability in the inputs space is less, and they are usually briefer and having less data. Table 4 lists detailed results comparison for number of training and validation points. Table 5 lists the detailed results comparison for detection time. Table 6 lists detailed results comparison for blind spot. Table 7 lists the detailed results comparison for extent of extrapolation and testing PE average. Table 8 shows the aggregated performance comparison between GP and PWLRR on cases from production sites. Table 9 shows the aggregated performance comparison between GP and PWLRR on cases from lab.

Case ID	Training & Validation Points		Total Points
	GP	PWLRR	
#1	2500	19571	41524
#2	1250	12962	29591
#3	800	5710	12513
#4	670	1345	11914
#5	170	840	1654
#6	330	2160	6732
#7	800	4891	41523
#8	2000	8223	29848
#9	1900	4818	49080
#10	2000	5997	47125

Table 4 Number of training and validation points

Case ID	Detection Time			
	GP		PWLRR	
	Number of Points	Percentage of Total Points	Number of Points	Percentage of Total Points
#1	24794	59.71%	0	0.00%
#2	16766	56.66%	10509	35.51%
#3	3400	27.17%	2795	22.33%
#4	2841	23.84%	1077	9.04%
#5	119	7.17%	0	0.00%
#6	278	4.13%	0	0.00%

Table 5 Detection time

Case ID	Blind Spot			
	GP			PWLRR
	Leaning	Norm Rejection	Total	
#1	6.00%	32.60%	38.60%	79.70%
#2	4.20%	16.10%	20.30%	59.90%
#3	6.40%	11.50%	17.90%	57.20%
#4	5.60%	20.10%	25.70%	31.40%
#5	10.10%	8.10%	18.20%	58.90%
#6	4.90%	7.30%	12.20%	39.40%
#7	1.90%	26.10%	28.10%	37.90%
#8	6.70%	17.70%	24.40%	45.30%
#9	3.90%	5.80%	9.60%	15.60%
#10	4.20%	8.00%	12.20%	27.70%

Table 6 Blind spot

Case ID	Extrapolation		Testing PE Average	
	GP	PWLRR	GP	PWLRR
#1	1051.70%	18.50%	3.46E-05	4.92E-05
#2	666.80%	1.50%	1.35E-04	1.35E-04
#3	636.60%	8.70%	7.96E-05	1.15E-04
#4	95.30%	35.70%	9.92E-05	1.03E-04
#5	1244.00%	40.00%	8.52E-04	1.32E-03
#6	1452.70%	89.80%	2.25E-05	2.58E-05
#7	350.70%	0.50%	2.11E-04	2.53E-04
#8	2050.40%	14.50%	6.03E-04	8.16E-04
#9	1522.60%	28.60%	1.20E-04	1.55E-04
#10	1188.10%	21.50%	9.25E-05	1.03E-04

Table 7 Extent of extrapolation and testing PE average

	PWLRR	GP
Prediction Error (PE)	Baseline	-19%
Average # of Train/Validate Points	8,882	1,607
Blind Spots	46%	22%
Level of Extrapolation	13%	1067%
True Positives	2/3	3/3
Detection Time (Percentage of Total Points)	Avg: 19.28% Std: 17.95%	Avg: 47.85% Std: 17.97%
False Positives	3	0
Relative Model Development Effort	*Baseline*	*-60%*

Table 8 Aggregated comparison for cases from production site

	PWLRR	GP
Prediction Error (PE)	Baseline	-17%
Average # of Train/Validate Points	1,448	390
Blind Spots	43%	19%
Level of Extrapolation	55%	931%
True Positives	1/3	3/3
Average Detection Time (Percentage of Total Points)	Avg: 3.01% Std: 5.22%	Avg: 11.71% Std: 10.61%
Relative Model Development Effort	*Baseline*	*-60%*

Table 9 Aggregated comparison for cases from lab

5. SUMMARY AND CONCLUSIONS

A new approach for learning the model of the fault detection system by performing GP for regression is proposed.

5.1 Summary of Research

A model-based fault detection system by electrical signature analysis is previously designed and implemented to monitor the health of induction motor driving assets. The objective of this research is to find a learning method which can learn the model of the fault detection system faster and more effective than the existing PWLRR method.

In Section 1 of the dissertation, a brief introduction on the induction motor driving assets and their failure types is discussed. The fault detection methods with different type of sensors are also review. Then, several learning techniques to train the model for the model-based fault detection system are investigated and compared. Finally, the contributions of this work are listed.

Section 2 describes the PWLRR method in the existing fault detection system. The format of the inputs and output of the data for training and prediction is explained. The procedure of data usage in the model learning and the existence of the blind spot during prediction are discussed thereafter. A simple fault detection logic is reviewed at the end of the section.

In Section 3, the algorithm of GP and its application is reviewed. Then the implementation of using GP for regression on the existing fault detection system is discussed. The parameters of the GP and the selection of the parameters are also explained in this section. Then the data usage for both GP and PWLRR is introduced, using convex hull to measure the extent of extrapolation is discussed. In addition, the necessity of having a way of limiting the prediction for GP is investigated. Finally, a list of performance indicators is summarized for GP and PWLRR comparison.

The first part of the Section 4 lists five cases with artificial data, starting from low dimensional inputs to high dimensional inputs. 1-D cases and 2-D cases are tested with both GP and PWLRR with figures which are displaying the trained model visually. A case with 2-D data which contains a hole in the input space is also tested. At the end, two 4-D artificial cases are created to demonstrate the potential performance improvement for blind spot, extent of extrapolation and prediction error with GP may achieve, and the risk of follow-on training in delaying the detection.

The second part of the Section 4 lists the results from 10 real cases tested by GP and PWLRR. These 10 cases cover variety of applications and health situations. The detailed benchmark was made between GP and PWLRR by comparing all the performance indicators defined in Section 3 The aggregated performance is also listed by combing the results from each individual case.

5.2 Conclusions

Based on empirical evidence over a limited number of artificial and real-world test problems, GP is a more effective machine learning model generation method than PWLRR.

- Based on comparable training & validation PE in both GP and PWLRR, GP uses far fewer data to train as compared to PWLRR: For cases from production site, on average, GP uses 1,607 data to train, while PWLRR uses 8,882 data to train. For cases from lab, 390 data for GP and 1,448 data for PWLRR.

- Even when considering rejection of prediction points, GP has fewer overall blind spots as compare to PWLRR: For cases from production site, on average, GP has 22% blind spot, while PWLRR has 46%. For cases from lab, 19% for GP and 43% for PWLRR.
- For the same time horizon, GP prediction is performed with just initial training while PWLRR requires follow-on training to generate comparable prediction accuracy; follow-on training presents significant complexities.
- GP extrapolates more effectively beyond the training domain as compared to PWLRR: For cases from production site, on average, GP has achieved 1067% extrapolation, while PWLRR achieves 13%. For cases from lab, 931% for GP and 55% for PWLRR.
- GP predicts health condition of the assets better than PWLRR: For all 6 cases with fault (3 production + 3 lab), GP performs successful detection on all of them while PWLRR only detects 3 out 6 (2 production + 1 lab). Also, for 4 cases without fault, GP has 0 false positive while PWLRR has 3 false positives. Further, the detection time of GP is longer than PWLRR, GP achieves average percentage of detection points in total points 47.85% (production) and 11.71% (lab) while PWLRR has 19.28% (production) and 3.01% (lab).
- The estimated computational and/manual effort to generate a GP model is far less than the equivalent estimated effort to generate a PWLRR model.

5.3 Recommendations for Future Work

A number of algorithmic improvements can be considered to further enhance the GP model predictive performance:

- Optimize the GP model generation process by considering variants to covariance matrix inversion, etc.

- Decouple the GP training and prediction steps to accelerate the prediction process.
- Further reduce the blind spot in GP models when the prediction region is too far away from the training region; enable some follow-on (or incremental) training while limiting the risks carried by extensive follow-on training.

REFERENCES

[1] L. Wang, “Induction Motor Bearing Fault Detection Using a Sensorless Approach”, PhD Dissertation in Texas A&M University, May 2009.

[2] J. E. McInroy and S. F. Legowski, “Using power measurements to diagnose degradations in motor drivepower systems: A case study of oilfield pump jacks", IEEE Transactions on Industry Applications, Vol. 37, Issue 6, 2001.

[3] M. Messaoudi and L. Sbita, "Multiple Faults Diagnosis in Induction Motor Using the MCSA Method", International Journal of Signal & Image Processing, Vol. 1, Issue 3, 2010.

[4] P. P. Harihara, “Sensorless Fault Diagnosis of Centrifugal Pumps”, PhD Dissertation in Texas A&M University, May 2007.

[5] Md. R. Islam, J. Uddin and J.M. Kim "Acoustic Emission Sensor Network Based Fault Diagnosis of Induction Motors Using a Gabor Filter and Multiclass Support Vector Machines", Adhoc & Sensor Wireless Networks, Vol. 34, Issue 1-4, 2016.

[6] T. Ch. Anil Kumar, G. Singh and V. N. A. Naikan, "Broken Rotor Bar Fault Diagnosis in VFD Driven Induction Motors by an Improved Vibration Monitoring Technique", International Journal of Performability Engineering, Vol. 13, Issue 1, Janurary 2017.

[7] S. Zolfaghari, S. B. Mohd Noor, Md. R. Mehrjou, Md. H. Marhaban and N. Mariun, "Broken Rotor Bar Fault Detection and Classification Using Wavelet Packet Signature Analysis Based on Fourier Transform and Multi-Layer Perceptron Neural Network", Applied Sciences (2076-3417), Vol. 8, Issue 1, Janurary 2018.

[8] A. Stief, J. R. Ottewill, M. Orkisz and J. Baranowski, "Two Stage Data Fusion of Acoustic, Electric and Vibration Signals for Diagnosing Faults in Induction Motors", Elektronika ir Elektrotechnika, ISSN 1392-1215, Vol. 23, Issue 6, 2017.

[9] A. M. Venugopal, “Comparative Analysis of Electrical and Mechanical Fault Signatures in Induction Motors”, Master of Science thesis, Texas A&M University, College Station, December 2003.

[10] IAS Motor Reliability Working Group, “Report of Large Motor Reliability Survey of Industrial and Commercial Installations, Part I”, IEEE Transactions on Industry Applications, Vol. IV-21, Issue 4, Pages 853–864, July 1985.

[11] P. F. Albrecht, J. C. Appiarius, R. M. McCoy, E. L. Owen, and D. K. Sharma, “Assessment of the Reliability of Motors in Utility Applications - Updated”, IEEE Transactions on Energy Conversion, Vol. EC-1, Issue 1, Pages 39–46, March 1986.

[12] IEEE recommended practice for the design of reliable industrial and commercial power systems. IEEE Standard 493-1997 [IEEE Gold Book]

[13] N. D. Hoang, A. D. Pham, Q. L. Nguyen and Q. N. Pham, "Estimating Compressive Strength of High Performance Concrete with Gaussian Process Regression Model", Journal of Machine Learning Research, Vol. 18, Issue 99-123, 2017.

[14] S. Park, J. Lee and Y. Son, "Predicting Market Impact Costs Using Nonparametric Machine Learning Models”, PLoS ONE, Vol. 11, Issue 2, February 2016.

[15] M. P. Deisenroth, D. Fox and C. E. Rasmussen, “Gaussian Processes for Data-Efficient Learning in Robotics and Control", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, Issue 2, Pages 408-423, February 2015.

[16] N. K. Ahmed, A. F. Atiya, N. El Gayar, and H. El-Shishiny, “An Empirical Comparison of Machine Learning Models for Time Series Forecasting”, Econometric Reviews, Vol. 29, Issue 5-6, 2010.

[17] J. Reid, “infpy 0.4.13”, http://pythonhosted.org/infpy/gps.html, 2012

[18] G. Yi, J. Q. Shi and T. Choi, "Penalized Gaussian Process Regression and Classification for High-Dimensional Nonlinear Data", Biometrics, Vol. 67, Issue 4, Dec. 2011.

[19] C. E. Rasmussen and C. K. I. Williams, “Gaussian Processes for Machine Learning”, University Press Group Limited, 2006.

[20] J. W. McKean, "Robust Analysis of Linear Models". Statistical Science. Vol. 19, Issue 4, Pages 562–570, 2004.

[21] R. R. Wilcox, "Linear regression: robust heteroscedastic confidence bands that have some specified simultaneous probability coverage", Journal of Applied Statistics, Vol. 44, Issue 14, Pages 2564-2574, November 2017.

[22] Wikiwand, “Gaussian Process”, http://www.wikiwand.com/en/Gaussian_process, 2016.

[23] S. M. Mukhtar, H. Daud, S. C. Dass, "Squared exponential covariance function for prediction of hydrocarbon in seabed logging application", AIP Conference Proceedings, Vol. 1787, Issue 1, Pages 1-6, 2016.

[24] S. Baran, "K-optimal designs for parameters of shifted Ornstein–Uhlenbeck processes and sheets", Journal of Statistical Planning & Inference, Vol. 186, Pages 28-41, July 2017.

[25] V. Rao, R. P. Adams, D. D. Dunson, “Bayesian inference for Matern repulsive processes", Journal of the royal statistical society series b-statistical methodology, Vol. 79, Issue 3, Pages 877-897, June 2017.

[26] C. Fulton, "Mechanics of linear quadratic Gaussian rational inattention tracking problems", U.S. Federal Reserve Board's Finance & Economic Discussion Series. Pages 1-102, November 2017.

[27] C. E. Rasmussen and H. Nickisch, “GPML Matlab Code version 4.1”, http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html, 2017.

[28] C. B. Barber, D. P. Dobkin and H. Huhdanpaa, "The quickhull algorithm for convex hulls", ACM Transactions on Mathematical Software. Vol. 22, Issue 4, Pages 469–483, Janurary 1995.

[29] E. Mucke, "Computing Prescriptions: Quickhull: Computing Convex Hulls Quickly", Computing in Science and Engineering, Vol. 11, Issue 5, Pages 54-56, September 2009.

[30] C.B. Barber, "Qhull", http://www.qhull.org/, 2016