University of Alberta

Artificial Intelligence in Electrical Machine Condition Monitoring

by

Youliang Yang

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Fall 2009
Edmonton, Alberta

Examining Committee

Dr. Qing Zhao, Electrical and Computer Engineering

Abstract

Electrical machine condition monitoring plays an important role in modern industries. Instead of allowing the machines to run until failure, it is desired to gather more information about the machine condition before the machine is shutdown, so that the machine downtime can be reduce due to repair. Also, it would be very useful to keep track of the machine condition and predict the future machine condition so that maintenance plan can be scheduled in advance. In this thesis, artificial intelligence techniques are utilized for machine condition monitoring. The thesis consists of 3 parts. In the first part, Neural Network and Support Vector Machine models are built to classify different machine conditions. In the second part, time series prediction models are built with Support Vector Regression and Wavelet Packet Decomposition to predict the machine future vibration. Support Vector Regression is applied again in the final part of the thesis to try to keep track of the machine condition and determine if the machine has thermal sensitivity issue or not. In all 3 parts, experimental results are promising and they certainly can be used in practice in order to facilitate the machine condition monitoring process.

Acknowledgements

I would like to express my appreciation to Dr. Qing Zhao for giving me the opportunity to join the Control Systems group and supervising me throughout my graduate studies.

I would like to thank my colleague, Shugen Li, for his assistance on my research.

I would also like to thank Syncrude Canada for its support on my research. Special thanks to Dan Wolfe. Dan has provided me valuable industrial expertise throughout my research.

Finally, I would like to thank my parents for their encouragements and supports on my graduate studies.

Contents

List of Figures

List of Tables

List of Abbreviations

NN	.....	Neural Network
SVM	.....	Support Vector Machine
SVR	.....	Support Vector Regression
BPSTG	.....	Back Pressure Steam Turbine Generator
RMS	.....	Root Mean Square
RMSE	.....	Root Mean Square Error
DWT	.....	Discrete Wavelet Transform
WPT	.....	Wavelet Packet Transform
WPD	.....	Wavelet Packet Decomposition
PCA	.....	Principal Component Analysis
ICA	.....	Independent Component Analysis
GA	.....	Genetic Algorithm
KKT	.....	Karush-Kuhn-Tucker
RBF	.....	Radial Basis Function

Chapter 1

Introduction

1.1 Motivation

Electrical machine condition monitoring plays an important role in modern industries and it is an on going research topic. Traditionally, electrical machines are allowed to run until failure, and then the machines are taken off-line and either repaired or replaced. One of the main disadvantage of this kind of maintenance strategy is that it usually results in unexpected shutdowns of the machines. Also, since the condition of the machine is not monitored before the machine runs into failure, the machine downtime can be very long because after the machine is taken off-line, its condition needs to be checked first in order to determine which part of the machine does not function properly. Thus, this kind of maintenance strategy can cause great productivity lost and hence economic lost.

Later on, another kind of maintenance strategy is introduced, call predictive maintenance. In predictive maintenance, the condition of the machine is monitored continuously. There are two goals in predictive maintenance. One is to determine if the machine is operating in normal condition. If not, what is the fault the machine is experiencing. This is called diagnosis. The other goal is to keep track of the machine state and determine if the machine condition is becoming worse and worse and when the machine is likely needed to be shutdown and repaired, which is called prognosis. With predictive maintenance,

valuable information regarding machine condition can be obtained before the machine is shutdown, allowing maintenance activity to be scheduled in advance and hence reduce the machine downtime.

When predictive maintenance is first introduced, highly trained experts are usually required in order to analysis the data collected from the machine and determine the current condition of the machine. As the development of artificial intelligence, it becomes possible to built statistic models to replace human experts to monitor the machine conditions. In machine diagnosis, a classifier is required to be trained with the machine data beforehand so that it can recognize different machine conditions. If the trained classifier is sufficiently accurate, it can be used in the future to classify different machine conditions and thus provides valuable information about the machine condition before it is shutdown. In machine prognosis, a time series prediction model is needed. The model is trained to predict the machine future condition based on its past and current condition. Thus, it has the ability to tell if the machine condition is becoming worse and worse. It may even able to tell when the machine should be shutdown and repaired. Therefore, prediction model is very useful since it can greatly reduce the number of unexpected machine shutdowns.

Neural Network (NN) and Support Vector Machine (SVM) are two important topics in artificial intelligence and they have been used extensively in the field of machine condition monitoring. For example, in [1] and [2], NN has been utilized to classify different faults for rotating machines. In [3], NN is used in regression instead of classification to predict the vibration of a turbo-generator. SVM is a relatively new method comparing to NN. It was introduced in 1995 by Vapnik [4]. After its introduction, SVM is rapidly employed in the field machine condition monitoring. For instance, classifier is built with SVM in [6] to classify multiple faults for induction motors. While SVM cannot be used directly in regression problems, based on the SVM theory, Support Vector Regression (SVR) is introduced to deal with regression problem. In [7], SVR is

employed to predict the machine future condition.

As can be seen, NN, SVM, and SVR are very important tools in machine condition monitoring. In this thesis, three identical back pressure steam turbine generators (BPSTG) in a local oil-sand company are studied. Those machine learning tools will be utilized to keep track of the machine conditions and provide important information about the machines conditions before the machines are shutdown, and hence reduce the machines downtime.

1.2 Research objectives

The objective of this thesis consists of 3 parts. The first part is to build classifier with machine learning method to classify the machines conditions before the machine are shutdowns and repaired. Currently, all 3 machines are continuously monitored and their vibration data are collected. However, knowledge base has not been built to automatically classify the machines conditions. Each machine is scheduled to shutdown and repaired every certain number of years, and before the shutdown, very limited information about the machines conditions are known. In this thesis, classifiers will be built to try to automatically classify the machines conditions. Based on the classification result, maintenance plan can be made accordingly and therefore reduce the machine downtime. The second part of the thesis is to build time series prediction models to predict the machine future vibration. During the operation of the machines, there is a limit on their vibrations because if the vibration is too high, it can cause damage to the machine equipments and cause serious safety issue. Thus, accurately predicting the future machine vibration is very important. If the predicted vibration is higher than the pre-set limit, actions can be taken in advance to try to decrease the vibration. This last part of this thesis is to apply the machine learning techniques to keep track of the machines conditions. Accordingly to the on-site engineers in the oil-sand company, some of the machines may be suffered from serious thermal sensitivity problem. In this thesis, system models will be built to keep track of the machine vibrations

and determine if the machines have thermal sensitivity issue or not based on the operational data. The models can also provide information on how the machine vibrations are changing as time progresses due to thermal sensitivity.

1.3 Organization of the thesis

The thesis is organized as follows. In Chapter 2, the basic theories of Wavelet Transform will be introduced. Artificial intelligence techniques, including Neural Network, Support Vector Machine, and Support Vector Regression, are reviewed in Chapter 3. In Chapter 4, Neural Network and Support Vector Machine are utilized to classify different machine conditions, while in Chapter 5, Support Vector Regression model is built to predict machine future vibration. In Chapter 6, Support Vector Regression is used again to keep track of the machine condition and determine if the machines have thermal sensitivity problem. Finally, the conclusion of the thesis and possible areas for future research will be presented in Chapter 7.

Chapter 2

Wavelet Transform

In machine diagnosis, when building classification models to classify machine conditions, in order to improve the classification accuracy, raw machine data, such as the machine vibration data, may not be used directly to build classification models. Instead, some techniques may be applied to first extract features from the raw data. Extracting features in the time domain and frequency domain from the machine vibration data are 2 common feature extraction methods in machine condition monitoring. Time domain features usually include mean, variance, root mean square (rms), etc. In [8], the authors used time domain features to built classification model to classify machine faults. On the other hand, machine data will need to be first transformed into the frequency domain and then frequency domain features can be extracted [9].

After Wavelet Transformed is introduced, it becomes more and more popular in extracting features from machine raw data. In [10], discrete Wavelet Transform is used to extract features from vibration signals of a gearbox, while in [11], discrete Harmonic Wavelet Packet Transform is employed to extract features from vibration signals measured from bearings. One of the main advantages the Wavelet Transform has is that it can transform a time domain signal into a time-frequency domain signal [10]. When a signal is represented in the time domain, its frequency domain information is lost. On the other hand, time domain information will not be available when a signal is represented in the frequency domain. The general theory of Wavelet Transform is

reviewed in the the following sections.

2.1 Continuous Wavelet Transform

The continuous Wavelet Transform can be expressed by the following equation ([12]):

$$T(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi^*(\frac{t-b}{a})dt \tag{2.1}$$

where $x(t)$ is a finite energy signal, a is the dilation parameter, b is the translation parameter, and $\psi(t)$ is the mother wavelet. The asterisk in the equation indicates the complex conjugate is used. The factor $1/\sqrt{a}$ is used for energy conservation.

2.2 Discrete Wavelet Transform

In practice, parameter a and b are usually chosen to be some discrete numbers in order to reduce the computation load. A common selection for those 2 parameters are $a = 2^m$ and $b = n2^m$. Thus, Eq (2.1) becomes:

$$T(a,b) = 2^{m/2} \int_{-\infty}^{\infty} x(t)\psi^*(2^m t - n)dt \tag{2.2}$$

which is the discrete Wavelet Transform (DWT). In practice, the discrete wavelet decomposition (WD) can be implemented by using a low-pass filter, $h(n)$, which related to the scaling function $\varphi(t)$, and a high-pass filter, $g(n)$, which related to the wavelet function $\phi(t)$ ([13], [14]):

$$h(n) = \frac{1}{\sqrt{2}} < \varphi(t), \varphi(2t-n) > \tag{2.3}$$

$$g(n) = \frac{1}{\sqrt{2}} < \phi(t), \phi(2t-n) >= (-1)^n h(1-n) \tag{2.4}$$

Figure 2.1 shows a 1 level discrete wavelet decomposition and reconstruction. During the decomposition, the original signal, $x(t)$, is convoluted with a low-pass filter and a high-pass filter. The outputs from both filters will be

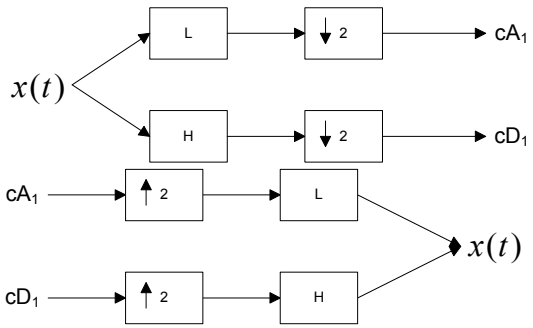


Figure 2.1: One-level of discrete wavelet decomposition and reconstruction

down sampled and resulting in two vectors, cA_1 and cD_1, which are called the approximation coefficients and the detail coefficients, respectively. During the reconstruction, the process is reversed. Sub-signals A_1 and D_1 can be reconstructed from the wavelet coefficients cA_1 and cD_1, respectively. The final reconstructed signal, $\hat{x}(t)$, can be obtained by summing A_1 and D_1:

$$\hat{x}(t) = A_1 + D_1 \tag{2.5}$$

In general, for a multi-level wavelet decomposition case, the approximate coefficients in the ith level, cA_i, will be further decomposed into cA_{i+1} and cD_{i+1}. Figure 2.2 shows a 3-level wavelet decomposition. The signal reconstruction can be described mathematically as:

$$A_{i-1} = A_i + D_i \tag{2.6}$$

$$\hat{x}(t) = A_j + \sum_{i \leq j} D_i \tag{2.7}$$

where i and j are positive integers and j is the level of wavelet decomposition.

2.3 Wavelet Packet Transform

Wavelet Packet Transform (WPT) is an extension to DWT. Unlike in WD, only the approximate coefficients will be further decomposed at each level, both the approximate coefficients and the detail coefficients will be further decomposed at each level in wavelet packet decomposition (WPD). Thus, a wavelet packet tree is generated. Figure 2.3 shows a wavelet packet tree of a 3-level WPD. As

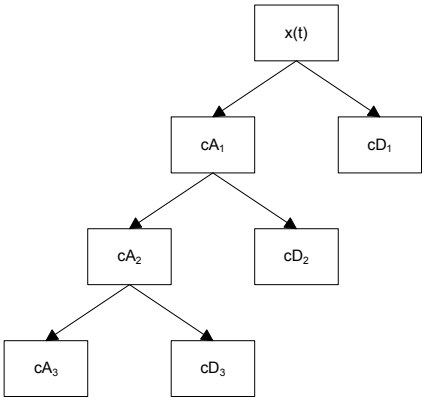


Figure 2.2: Three-level of discrete wavelet decomposition

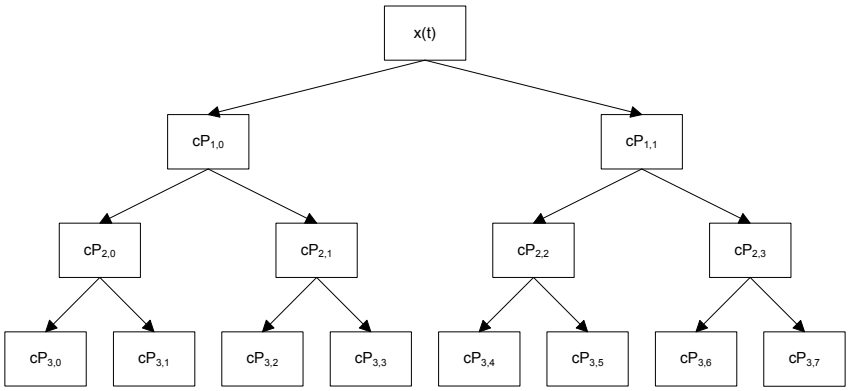


Figure 2.3: Three-level of wavelet packet decomposition

stated in [14], each node in the wavelet packet tree can be labeled by a pair of integers (j,k), where j is the corresponding level of decomposition and k is the order of the node position in a certain decomposition level with $0 \leq k \leq 2^j - 1$. The original signal can be reconstructed by summing all the sub-signals $P_{j,k}$ which are reconstructed from the wavelet packet coefficients $cP_{j,k}$ for each node (j, k) in the jth level decomposition:

$$\hat{x}(t) = \sum_{k=0}^{2^j-1} P_{j,k} \tag{2.8}$$

One important point needs to be emphasized is that, when a signal is decomposed into j-level using WPD, the order of the reconstructed sub-signals, $P_{j,k}$, may not be the same as the frequency order. For example, Table 2.1 shows the frequency band each reconstruction signal $P_{j,k}$ contains for a 3-level WPD

Reconstruction signals	Frequency bands
$P_{3,0}$	0-125Hz
$P_{3,1}$	125-250Hz
$P_{3,2}$	375-500Hz
$P_{3,3}$	250-375Hz
$P_{3,4}$	875-1000Hz
$P_{3,5}$	750-875Hz
$P_{3,6}$	500-625Hz
$P_{3,7}$	625-750Hz

Table 2.1: Reconstruction signals with corresponding frequency bands for a 3-level WPD

when the sampling rate of the original signal $x(t)$ is 2000Hz. It can be seen that the reconstruction signals with higher order may not contain higher frequency band. The reason is that downsampling may cause frequency folding in low pass filters and hence they may contain high frequency contents of the signal ([13] [14]).

Chapter 3

Artificial intelligence

As mentioned earlier, Neural Network, Support Vector Machine, and Support Vector Regression are 3 important topics in artificial intelligence. They are mainly used in classification and regression problems. In this chapter, the basic theories of these 3 methods are briefly introduced with some literature reviews.

3.1 Neural Network

Neural Network is an important part of machine learning. As more and more powerful computers are available, larger and larger dimension Neural Network models may be built [20] for complex systems. One of the main applications of Neural Network is building classifiers. As stated in [21], when dealing with simple regression and classification problems, linear models, such as least squares, are preferred due to its simplicity. However, for problems with higher dimension and non-linearity, the applicability of the linear models are usually limited. On the other hand, Neural Network is theoretically capable of approximating any non-linear function to any arbitrary accuracy when two or more hidden layers of neurons are used [1]. Nowadays, Neural Network is used in many different fields. For example, In [22], in order to help electricity suppliers make better marketing strategy, Neural Network classifier was built to classify electricity consumers into different groups based on their energy consumption. Also, the authors in [23] built Neural Network classifiers to predict if a com-

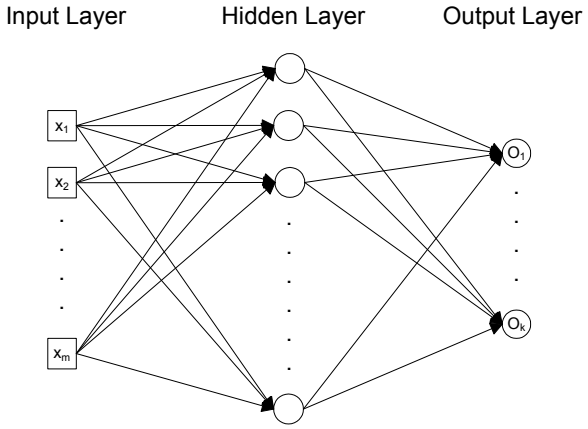


Figure 3.1: Neural Network structure

pany will go bankruptcy based on its financial statement. Moreover, Neural Network has been used extensively in the field of image recognition. In [24], Neural Network classifier is used for face recognition. Also, Neural Network is utilized in [25] to recognize handwritten digits.

Figure 3.1 shows the structure of a Feed-Forward Neural Network with 1 hidden layer. From the figure, it can be seen that the structure consists of 3 layers, input layer, hidden layer, and output layer. The input layer contains the inputs to the network, x_i, while the output layer contains the outputs of the network, O_i. The hidden layer is used to map the inputs to the output and it can have more than 1 layer. The circles in the figure are referred to as neurons. Each neuron is connected to the elements in the previous layer and the layer after by a line, and there is a weight, w_i, associated with each line. Each neuron can have many inputs and produce 1 output. The operation inside a neuron can be described by the following equation:

$$O^i = \sigma(\sum_r w_r^{i-1} O_r^{i-1}) \tag{3.1}$$

where the superscript i indicates the layer number. The function σ is called the activation function [21]. In a regression problem, the activation function is identity. Hence,

$$O^i = \sum_r w_r^{i-1} O_r^{i-1} \tag{3.2}$$

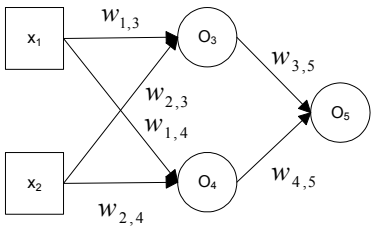


Figure 3.2: Neural Network structure with 2 inputs and 2 hidden neurons

On the other hand, in a classification problem, the activation function can be some threshold functions, such as the sign function and the tanh function. Usually, a sigmoid function is used as an activation function,

$$\sigma(a) = \frac{1}{1 + e^{-a}} \tag{3.3}$$

It can be easily noticed that $0 \leq \sigma(a) \leq 1$. Thus, in a two-class classification problem, when the network output is larger than 0.5, the inputs can be classified to Class 1. On the other hand, when the network output is less than 0.5, the inputs can be classified to Class 0. In a multi-class problem, one option is to increase the number of output, so that for a specific class, only one output can have a value higher than 0.5, and all the other outputs have values less than 0.5.

The calculation of a NN output and the NN training algorithm are presented in [26], and they are briefly reviewed in the following sections. For simplicity, the network in Figure 3.2 is considered. The network has 2 inputs, 1 output, and 1 hidden layer. By working backward, the following relationship can be obtained:

$$\begin{aligned} O_5 &= \sigma(\sum_r w_{r,5} O_r) = \sigma(w_{3,5} O_3 + w_{4,5} O_4) \\ &= \sigma(w_{3,5}\sigma(\sum_s w_{s,3} O_s) + w_{4,5}\sigma(\sum_t w_{t,4} O_t) \\ &= \sigma(w_{3,5}\sigma(w_{1,3} O_1 + w_{2,3} O_2) + w_{4,5}\sigma(w_{1,4} O_1 + w_{2,4} O_2)) \end{aligned} \tag{3.4}$$

When training the Neural Network, the objective is to minimize the following error function:

$$E = \frac{1}{2}(O - t)^2 \tag{3.5}$$

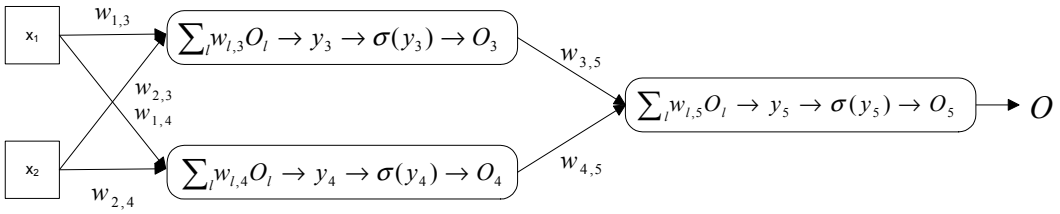


Figure 3.3: Neural Network structure with 2 inputs and 2 hidden neurons

where E is the error, O is the Neural Network output, and t is the real value. Backpropagation algorithm is usually used to train a Neural Network. Its general theory is reviewed below.

Let

$$\delta_i = \frac{\partial E}{\partial y_i} \tag{3.6}$$

$$\frac{\partial E}{\partial w_{3,5}} = \frac{\partial E}{\partial y_5}\frac{\partial y_5}{\partial w_{3,5}} = \delta_5 \frac{\partial y_5}{\partial w_{3,5}} \tag{3.7}$$

$$\frac{\partial y_5}{\partial w_{3,5}} = \frac{\partial (w_{3,5}O_3 + w_{4,5}O_4)}{\partial w_{3,5}} = O_3 \tag{3.8}$$

Hence,

$$\frac{\partial E}{\partial w_{3,5}} = \delta_5 O_3 \tag{3.9}$$

In general,

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial w_{i,j}} = \delta_j O_i \tag{3.10}$$

To compute δ_5,

$$\delta_5 = \frac{\partial E}{\partial y_5} = \frac{\partial E}{\partial O_5}\frac{\partial O_5}{\partial y_5} \tag{3.11}$$

$$\frac{\partial E}{\partial O_5} = \frac{\partial}{\partial O_5}[\frac{1}{2}(O_5 - t)^2] = (O_5 - t)\frac{\partial}{\partial O_5}(O_5 - t) = O_5 - t \tag{3.12}$$

$$\frac{\partial O_5}{\partial y_5} = \frac{\partial \sigma(y_5)}{\partial y_5} = \sigma(y_5)(1 - \sigma(y_5)) = O_5(1 - O_5) \tag{3.13}$$

Hence,

$$\delta_5 = (O_5 - t)O_5(1 - O_5) \tag{3.14}$$

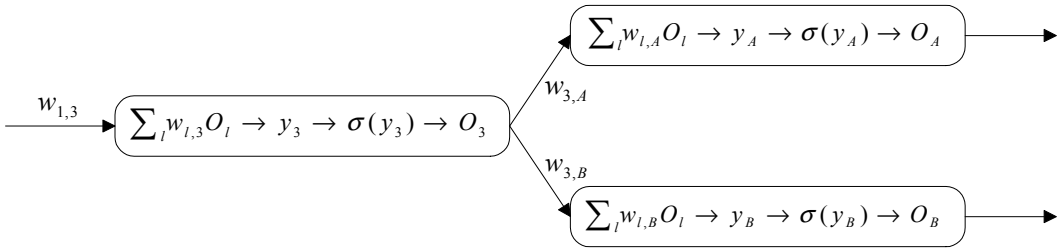


Figure 3.4: A non-terminal node is connected to more than 1 neurons in the next layer

Following the same procedure,

$$\frac{\partial E}{\partial w_{1,3}} = \frac{\partial E}{\partial y_3}\frac{\partial y_3}{\partial w_{1,3}} = \delta_3 O_1 \tag{3.15}$$

$$\delta_3 = \frac{\partial E}{\partial y_3} = \frac{\partial E}{\partial O_3}\frac{\partial O_3}{\partial y_3} \tag{3.16}$$

$$\frac{\partial E}{\partial O_3} = \frac{\partial E}{\partial y_5}\frac{\partial y_5}{\partial O_3} = \delta_5 \frac{\partial(\sum_l w_{l,5} O_l)}{\partial O_3} = \delta_5 w_{3,5} \tag{3.17}$$

$$\frac{\partial O_3}{\partial y_3} = \frac{\partial \sigma(y_3)}{\partial y_3} = \sigma(y_3)(1 - \sigma(y_3)) = O_3(1 - O_3) \tag{3.18}$$

Hence,

$$\delta_3 = (\delta_5 w_{3,5}) O_3(1 - O_3) \tag{3.19}$$

Comparing the equations for δ_3 and δ_5, it can be seen that they are not the same in terms of the equation format. This is because node 5 is a terminal node, while node 3 is not. In this case, node 3 is connected to only 1 node, node 5. The equation for δ_3 will be different again if node 3 is connected to more than 1 node. Consider the case shown on Figure 3.4. Node 3 is connected to 2 nodes, node A and B. As before,

$$\frac{\partial E}{\partial w_{1,3}} = \frac{\partial E}{\partial y_3}\frac{\partial y_3}{\partial w_{1,3}} = \delta_3 O_1 \tag{3.20}$$

$$\delta_3 = \frac{\partial E}{\partial y_3} = \frac{\partial E}{\partial O_3}\frac{\partial O_3}{\partial y_3} = \frac{\partial E}{\partial O_3}[O_3(1 - O_3)] \tag{3.21}$$

$$\frac{\partial E}{\partial O_3} = \frac{\partial E}{\partial y_A}\frac{\partial y_A}{\partial O_3} + \frac{\partial E}{\partial y_B}\frac{\partial y_B}{\partial O_3} = \sum_k \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial O_3}$$
$$= \sum_k \delta_k \frac{\partial(\sum_l w_{l.k} O_l)}{\partial O_3} = \sum_k \delta_k w_{3,k} \tag{3.22}$$

where k is the number of nodes node 3 is connected to. In this case,

$$\delta_3 = O_3(1 - O_3)[\delta_A w_{3,A} + \delta_B w_{3,B}] \tag{3.23}$$

In general, for a non-terminal node,

$$\delta_l = O_l(1 - O_l)\sum_k \delta_k w_{l.k} \tag{3.24}$$

To summarized, the backpropagation algorithm includes the following computations:

1. Calculate the network output from the given input.

2. Update δ based on the following equations:

$$\frac{\partial E}{\partial w_{l,n}} = \delta_n O_l \tag{3.25}$$

$$\delta_n = \frac{\partial E}{\partial y_n} = O_n(1 - O_n)\begin{cases} t - O & \text{if terminal} \\ \sum_k \delta_k w_{n,k} & \text{otherwise} \end{cases} \tag{3.26}$$

3. Update the weights of the network,

$$w_{i,j} = w_{i,j} + \eta \delta_j O_i \tag{3.27}$$

where η is the learning rate.

3.2 Support Vector Machine

Support Vector Machine is a relatively new technique which is developed in the last decade. It is first introduced by Vapnik in 1995 [4]. In SVM, by using kernel functions, the inputs belonging to different classes are mapped

into feature spaces, and the goal is to find an optimal hyperplane to separate the inputs based on their classes. Since it is introduced, SVM has been widely used in many different fields. For example, in [27], the authors built classifiers with SVM to predict in advance if a student will be admitted to a physical education school. With a candidate's performance on the physical ability test and National Selection and Placement Examination, and his/her grade point average at high school as the classifier inputs, SVM classifier is trained and used to predict if a student will be admitted or not. The authors showed that the classification accuracy is more than 90%. In [28], SVM classifier is built to classifier a credit card applicant into 2 types, 'good credit' and 'bad credit'. An applicant classified to 'good credit' type has higher probability to repay the financial obligation and his/her application will be approved. On the other hand, for an applicant classified to 'bad credit' type, his/her application will be denied. Further more, SVM is also widely used in the medical field. Automatic classifier is particular useful in this field since there are many cases that the doctors need to determine if a person is healthy, or what kinds of disease a person is having. With the classifiers, a person's health condition may be determined automatically. As an example, in [29], the authors built SVM classifier to predict what type of cancers a patient may have. The general theory of SVM is reviewed below [30].

3.2.1 Hard-margin SVM

Considering a 2-class classification problem, assuming there are N data points x_i $(i = 1, 2, ..., N)$, they are either belonged to Class 1 ($y_i = 1$) or belonged to Class 2. If the data are linearly separable, there would be a decision function:

$$D(x_i) = wx + b \geq 1 \quad \text{for} \quad y_i = 1 \tag{3.28}$$

$$D(x_i) = wx + b \leq 1 \quad \text{for} \quad y_i = -1 \tag{3.29}$$

where w is the weighted vector and b is a constant. Equation 3.29 can be rewritten as:

$$y_i(wx_i + b) \geq 1 \tag{3.30}$$

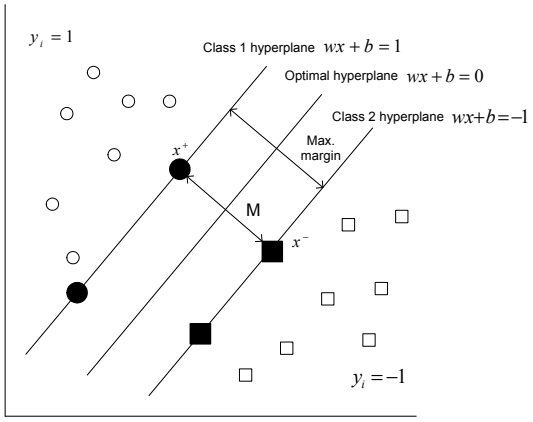


Figure 3.5: Linear separable data in two-dimension space

The equation

$$D(x) = wx + b = c \quad \text{for} \quad -1 < c < 1 \tag{3.31}$$

can be considered as a separating hyperplane that separates the data x_i. As defined in [30], the distance between the separating hyperplane to the nearest data is called the margin, as illustrated in Figure 3.5. In a classification problem with SVM, the goal is to find a separating hyperplane which maximizes the margin, and it is called the optimal separating plane. in this case, when $c = 0$, the margin is maximized.

Considering 2 points on the Class 1 plane and Class 2 plane, as shown in Figure 3.5, where $|x^+ - x^-|$ is perpendicular to the separating hyperplane. Thus,

$$|x^+ - x^-| = M \tag{3.32}$$

where M is the margin width. Since w is perpendicular to all 3 planes, the relationship between x^+, x^-, and w can be expressed as:

$$x^+ - x^- = \lambda w \tag{3.33}$$

where λ is a positive number. Also, since the points x^+ and x^- are on the Class 1 plane and Class 2 plane, respectively, they satisfy the following equations:

$$wx^+ + b = 1 \tag{3.34}$$

$$wx^- + b = -1 \tag{3.35}$$

Combining the above 4 equations, λ can be solved.

$$\begin{aligned} w(x^- + \lambda w) + b &= 1 \\ (wx^- + b) + \lambda w * w &= 1 \\ -1 + \lambda w * w &= 1 \end{aligned}$$

Hence,

$$\lambda = \frac{2}{w * w} \tag{3.36}$$

Also,

$$\begin{aligned} M &= |x^+ - x^-| = |\lambda w| \\ &= \lambda |w| = \lambda \sqrt{w * w} \\ &= \frac{2\sqrt{w * w}}{w * w} = \frac{2}{\sqrt{w * w}} \end{aligned} \tag{3.37}$$

Therefore, the goal is to maximize Eq. (3.37). In other words, the objective is to minimize the following equation:

$$Q(w) = \frac{1}{2}||w||^2 \tag{3.38}$$

subjecting to the following condition:

$$y_i(wx_i + b) \geq 1 \quad \text{for} \quad i = 1, 2, ...N \tag{3.39}$$

The data which lay on the Class 1 plane or the Class 2 plane will satisfy the equality in Eq (3.39), and these data are called support vectors. For the data which are not on the Class 1 nor Class 2 plane, they will satisfy the inequality in Eq (3.39). These data can be deleted while the optimal separating plane can still be determined.

Before trying to solve Eq. (3.38) and (3.39), the calculation can be transformed into a Lagrangian dual problem,

$$Q(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{M} \alpha_i y_i (w^T x_i + b) - 1 \tag{3.40}$$

where $\alpha = (\alpha_1, ..., \alpha_N)^T$ and they are the Lagrange multipliers. Eq. (3.40) satisfies the following Karush-Kuhn-Tucker (KKT) constrains:

$$\frac{\partial Q(w, b, \alpha)}{\partial w} = 0 \tag{3.41}$$

$$\frac{\partial Q(w, b, \alpha)}{\partial b} = 0 \tag{3.42}$$

$$\alpha_i y_i (w x_i + b - 1) = 0 \quad \text{for} \quad i = 1, 2, ..., N \tag{3.43}$$

$$\alpha_i \geq 0 \quad \text{for} \quad i = 1, 2, ..., N \tag{3.44}$$

For support vectors, they satisfy the equation $y_i(wx_i + b) = 1$. Thus, based on Eq. (3.44), $\alpha_i > 0$ for all support vectors. On the other hand, $\alpha_i = 0$ for all non-support vectors.

Using Eq. (3.40), Eq. (3.42) and (3.43) can be simplified as

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{3.45}$$

and

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.46}$$

With Eq. (3.45) and (3.46), Eq. (3.40) can be simplified as

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{3.47}$$

subject to

$$\sum_{i=1}^{N} y_i \alpha_i = 0, \quad \alpha_i \geq 0 \quad \text{for} \quad i = 1, 2, ..., N \tag{3.48}$$

At this point, quadratic programming can be used to solve the optimization problem and calculate α_i. Finally, the classification decision function becomes

$$D(x) = \sum_{i,j=1}^{N} \alpha_i y_i (x_i x_j) + b \tag{3.49}$$

3.2.2 Soft-margin SVM

In practice, there are many cases that the data are linearly inseparable. In those cases, hard-margin SVM cannot be used to classify the data and soft-margin SVM needs to be used. In soft-margin SVM, a slack variables ξ is

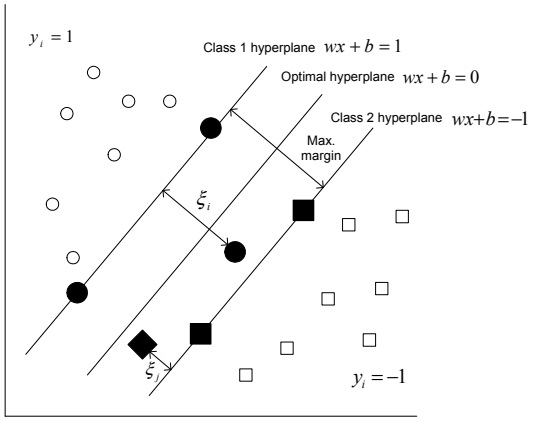


Figure 3.6: Linear inseparable data in two-dimension space

introduced and Eq. (3.30) becomes:

$$y_i(wx_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{and} \quad i = 1, 2, ..., N \tag{3.50}$$

From Figure 3.6, it can be seen that when $0 \leq \xi_i \leq 1$, even thought the distance between the data x_i and the optimal hyperplane is less than the maximum margin, the data can still be correctly classified. On the other hand, if $\xi_i \geq 1$, the data x_i will be misclassified. It is easy to understand that in soft-margin SVM, the goal is to minimize

$$Q(w, b, \xi) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i \tag{3.51}$$

subject to

$$y_i(wx_i + b) \geq 1 - \xi_i \tag{3.52}$$

where $\xi = (\xi_1, \xi_2, ..., \xi_N)$ and C is a constant parameter which determined the trade-off between the classification margin and the classification error.

Again, the above optimization problem can be transform into a Lagrange dual problem

$$\begin{aligned} Q(w, b, \xi, \alpha, \beta) =& \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i \\ & - \sum_{i=1}^{N} \alpha_i(y_i(wx_i + b) - 1 + \xi_i) - \sum_{i=1}^{N} \beta_i \xi_i \end{aligned} \tag{3.53}$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_N)$ and $\beta = (\beta_1, \beta_2, ..., \beta_N)$. The optimal solution satisfies the following KKT conditions:

$$\frac{\partial Q(w, b, \xi, \alpha, \beta)}{\partial w} = 0 \tag{3.54}$$

$$\frac{\partial Q(w, b, \xi, \alpha, \beta)}{\partial b} = 0 \tag{3.55}$$

$$\frac{\partial Q(w, b, \xi, \alpha, \beta)}{\partial \xi} = 0 \tag{3.56}$$

$$\alpha_i(y_i(wx_i + b) - 1 + \xi_i) = 0 \tag{3.57}$$

$$\beta_i \xi_i = 0 \tag{3.58}$$

$$\alpha \geq 0, \quad \beta_i \geq 0, \quad \xi_i \geq 0 \tag{3.59}$$

Similar to the hard-margin SVM case, using Eq. (3.53), Eq. (3.55) to (3.57) can be simplified as

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{3.60}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.61}$$

$$\alpha_i + \beta_i = C \tag{3.62}$$

Substituting Eq. (3.61) and (3.62) into Eq. (3.53), the optimization problem becomes to maximize

$$Q(\alpha) = \sum_{i=1}^{N} -\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{3.63}$$

subject to

$$\sum_{i=1}^{N} y_i \alpha_i = 0, \quad \text{and} \quad 0 \leq \alpha_i \leq C \tag{3.64}$$

Same as in the hard-margin SVM case, quadratic programming can be utilized to solved for α_i, and the classification decision function is

$$D(x) = \sum_{i=1}^{N} \alpha_i y_i x_i^T x + b \tag{3.65}$$

which is identical to the decision function in the hard-margin SVM case.

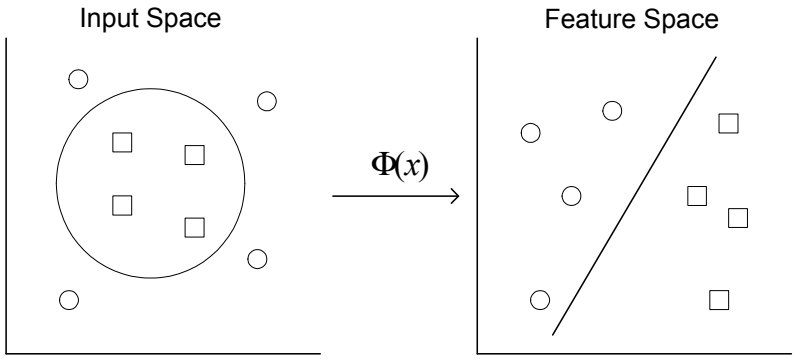


Figure 3.7: Mapping input data to feature space

3.2.3 Kernel functions

Soft-margin SVM certainly provides a useful way to classify linearly inseparable data; however, there are cases that even thought Soft-margin SVM is used, the optimal classifier still does not have high generalization capability. In order to enhance the linearly separability of the data, a non-linear vector function, $\Phi(x)$, can be used to transform the original m-dimension data x into q-dimension feature space, as shown in Figure 3.7, and the classification decision function becomes:

$$D(x) = \sum_{i,j=1}^{N} \alpha_i y_i \Phi(x_i)\Phi(x_j) + b^* \tag{3.66}$$

As stated in [4] and [31], if the dimension of the feature space is high, the transformation could be computational expensive. Since only the dot product, $\Phi(x_i)\Phi(x_j)$ is considered, it is possible to define a function:

$$K(x_i, x_j) = \Phi(x_i)\Phi(x_j) \tag{3.67}$$

to calculate the dot products, and it is called the kernel function. With the kernel function, the optimization problem becomes to maximize

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3.68}$$

subject to

$$\sum_{i=1}^{N} y_i \alpha_i = 0, \quad \text{and} \quad 0 \leq \alpha_i \leq C \tag{3.69}$$

Thus, the decision function is

$$D(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b^* \tag{3.70}$$

where

$$b = y_j - \sum_{i=1}^{N} \alpha_i y_i K(x_i, x_j) \tag{3.71}$$

Kernel functions cannot be defined arbitrary and they need to satisfy the Mercers theorem [4]. Some common kernel functions include linear, polynomial, and radial basis function (RBF). Readers can refer to [30] for more detail descriptions on those kernel functions and the other kernel functions.

3.2.4 SVM multi-class classification

SVM was first used as a dual classes classifier. However, in practice, there are many cases that more than 2 classes are involved and needs to be classified. Fortunately, many classification strategies using SVM have been introduced for multi-class classification. The one-against-all and one-against-one are two of them which are widely used when dealing with multi-class problems. They are briefly reviewed in the following sections. Readers can refer to [32] for more details.

In the one-against-all method, N SVM classifiers will be trained for an N-class classification problem. When training the ith classifier, $i \leq N$, the data samples from the ith class are labeled as $+1$, and the data samples from the other classes are labeled as -1. Thus, during testing, when data samples from the ith class are presented to the classifiers, only the ith classifier will classify the samples as $+1$, and the other classifiers will classify the samples as -1.

On the other hand, in the one-against-one method, $N(N-1)/2$ classifiers will be trained for an N-class classification problem. Each classifier is trained to classify 2 cases. For example, classifier A can be trained to classify the ith and jth classes. In testing, all test samples are presented to all classifiers. If a sample is classified to the ith class by classifier A, the score for the ith class

is increased by 1, otherwise the score for the jth class is increased by 1. After the sample is classified by all the classifiers, the class of the data sample can be determined based on which class has the highest score.

3.3 Support Vector Regression

SVM was original limited to solve classification problems. However, as the introduction of loss functions, SVM has been extended to solve regression problems, and the new technique is called Support Vector Regression [33]. As stated in [34], in SVR, the goal is to find a function $f(x)$, which maps the input to the output, while minimizing the difference between the predicted value $\hat{y}_i$ and the actual value y_i based on the loss function. Supposed there are training data $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$, where x_i is the input and y_i is the output. In a linear case, $f(x)$ can be expressed as

$$\hat{y} = f(x) = wx + b \tag{3.72}$$

where w is the weighted vector and b is a constant. While trying to minimize the difference between the predicted value and the actual value, in SVR, it is also desirable to keep the function $f(x)$ as flat as possible [34], which means w should be as small as possible. One way to find a small w is to minimize the norm, i.e. $||w||^2 =< w, w >$, where $< \cdot, \cdot >$ denotes the dot product. Thus, the regression problem becomes to

$$\text{min.} \quad \frac{1}{2}\sum_{n=1}^{N}(\hat{y}_i - y_i)^2 + \frac{1}{2}||w||^2 \tag{3.73}$$

Quadratic error function is used in Eq. (3.73) to calculate the error between the predicted value and the actual value. In practice, the ϵ-insensitive error function is often used, which is shown on Figure 3.8 and can be mathematically expressed as

$$E_\epsilon(\hat{y}_i - y_i) = \begin{cases} 0, & \text{if } |\hat{y}_i - y_i| < \epsilon \\ |\hat{y}_i - y_i| - \epsilon, & \text{otherwise} \end{cases} \tag{3.74}$$

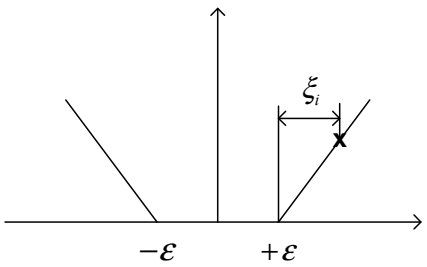


Figure 3.8: ϵ-insensitive error function

Reader can refer to [35] for more details on error function. With the ϵ-insensitive error function, the regression problem becomes to minimize

$$C\sum_{n=1}^{N} E_\epsilon(\hat{y}_i - y_i) + \frac{1}{2}||w||^2 \tag{3.75}$$

Where C is the trade-off between the flatness of $f(x)$ and the prediction error. Similar to SVM, in SVR, there are cases that with the optimal $f(x)$, some actual values may not lie within the region $[\hat{y} - \epsilon, \hat{y} + \epsilon]$, and slack variables ξ and $\hat{\xi}$ needs to be introduced to deal with those cases so that for any given actual value y_i, it lies within the region $[\hat{y}_i - \epsilon - \hat{\xi}_i, \hat{y}_i + \epsilon + \xi_i]$ (Please refer to Figure 3.9). When y_i lies above $\hat{y}_i + \epsilon$, $\xi_i > 0$ and $\hat{\xi}_i = 0$. On the other hand, when y_i lies below $\hat{y}_i - \epsilon$, $\xi_i = 0$ and $\hat{\xi}_i > 0$. Thus, the objective function of the SVR problem can be rewritten as

$$\text{min.} \quad C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i) + \frac{1}{2}||w||^2 \tag{3.76}$$

subject to

$$\xi_i \geq 0 \tag{3.77}$$

$$\hat{\xi}_i \geq 0 \tag{3.78}$$

$$y_i \leq \hat{y}_i + \epsilon + \xi_i \tag{3.79}$$

$$y_i \geq \hat{y}_i + \epsilon + \hat{\xi}_i \tag{3.80}$$

Again, the above optimization problem can be transformed into a La-

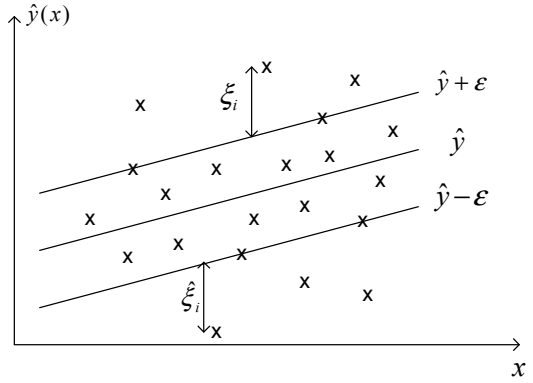


Figure 3.9: Linear SVR with slack variables

grangian dual problem [21]:

$$\begin{aligned} L =& C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i) + \frac{1}{2}||w||^2 - \sum_{i=1}^{N}(\mu_i\xi_i + \hat{\mu}_i\hat{\xi}_i) \\ & - \sum_{i=1}^{N}\alpha_i(\epsilon + \xi_i + \hat{y}_i - y_i) - \sum_{i=1}^{N}\hat{\alpha}_i(\epsilon + \hat{\xi}_i - \hat{y}_i + y_i) \end{aligned} \tag{3.81}$$

where $\alpha_i, \hat{\alpha}_i, \mu_i, \hat{\mu}_i \geq 0$ and they are the Lagrange multipliers. Substituting Eq. (3.72) into Eq. (3.81) and set the derivatives of L with respect to w, b, ξ_i, and $\hat{\xi}_i$ to zero, the following equations can be obtained

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N}(\alpha_i - \hat{\alpha}_i)x_i \tag{3.82}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N}(\alpha_i - \hat{\alpha}_i) = 0 \tag{3.83}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i + \mu_i = C \tag{3.84}$$

$$\frac{\partial L}{\partial \hat{\xi}_i} = 0 \Rightarrow \hat{\alpha}_i + \hat{\mu}_i = C \tag{3.85}$$

Substituting Eq. (3.82) to (3.85) into Eq. (3.81), the optimization problem is equivalent to maximize

$$\begin{aligned} \hat{L} =& -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j)x_i x_j \\ & - \epsilon\sum_{i=1}^{N}(\alpha_i + \hat{\alpha}_i) + \sum_{i=1}^{N}(\alpha_i + \hat{\alpha}_i)\hat{y}_i \end{aligned} \tag{3.86}$$

if kernel function is used, Eq. (3.86) can be rewritten as to maximize

$$\begin{aligned}\hat{L} = &-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j)k(x_i, x_j) \\ &- \epsilon\sum_{i=1}^{N}(\alpha_i + \hat{\alpha}_i) + \sum_{i=1}^{N}(\alpha_i + \hat{\alpha}_i)\hat{y}_i \end{aligned} \tag{3.87}$$

Eq. (3.87) needs to satisfy the condition in Eq. (3.83). From Eq. (3.84) and Eq. (3.85), it is easy to see that Eq. (3.87) also needs to satisfy the following conditions

$$0 \leq \alpha_i \leq C \tag{3.88}$$

$$0 \leq \hat{\alpha}_i \leq C \tag{3.89}$$

and the predicted value is given by

$$\hat{y}_i = \sum_{i=1}^{N}(\alpha_i - \hat{\alpha}_i)k(x, x_i) + b \tag{3.90}$$

The solution to Eq. (3.87) satisfies the following KKT conditions

$$\alpha_i(\epsilon + \xi_i + \hat{y}_i - y_i) = 0 \tag{3.91}$$

$$\hat{\alpha}_i(\epsilon + \hat{\xi} + \hat{y}_i - y_i) = 0 \tag{3.92}$$

$$(C - \alpha_i)\xi_i = 0 \tag{3.93}$$

$$(C - \hat{\alpha}_i)\hat{\xi}_i = 0 \tag{3.94}$$

It can be noticed that $\alpha_i \neq 0$ when $\epsilon + \xi_i + \hat{y}_i - y_i = 0$, which implies y_i lies on the upper boundary or above the upper boundary (see Figure 3.9). Similarly, $\hat{\alpha}_i \neq 0$ when $\epsilon + \hat{\xi} + \hat{y}_i - y_i = 0$, which means y_i lies on the lower boundary or below the lower boundary. Since $\epsilon + \xi_i + \hat{y}_i - y_i = 0$ and $\epsilon + \hat{\xi} + \hat{y}_i - y_i = 0$ cannot be satisfied simultaneously, at least one of α_i and $\hat{\alpha}_i$ must be zero.

For those data points which either $\alpha_i = 0$ or $\hat{\alpha}_i = 0$, since they determine the boundaries, they are the support vectors. For those data points which lie within $[\hat{y}_i - \epsilon, \hat{y}_i + \epsilon]$, they have $\alpha_i = \hat{\alpha}_i = 0$.

The parameter b can be found by

$$\begin{aligned} b =& y_i - \epsilon - \hat{y}_i \\ =& y_i - \epsilon - \sum_{j=1}^{N} (\alpha_j - \hat{\alpha}_j) k(x_i, x_j) \end{aligned} \tag{3.95}$$

In practice, it is often to find b by taking the average of all such estimates of b.

Chapter 4

Machine conditions classification

As shown in the previous chapters, NN and SVM are 2 important tools to solve classification problems in many different fields, and they have also been used extensively in the field of electrical machine condition monitoring to classify different machine conditions. For example, NN model is built in [36] to classify different gears and bearings conditions. In [4], SVM is used to classify different gear faults. In this chapter, with the machine vibration data, both NN and SVM classification models are built to classify different conditions of 3 back pressure steam turbine generators (BPSTG), so that the classification models can be used in the future to classify different machines conditions. As indicated in [5], NN may have limitations on generalization and can overfit the training data, which means NN may achieve excellent classification rate on the training data while giving poor results on the test data. This will also be validated in this chapter. The experimental setup is first introduced in the following section.

4.1 Experimental setup

In this thesis, 3 identical BPSTGs which are used in a local oil-sand company are studied, and they are labeled as G1, G2, and G4. All 3 generators are rated at 50MW and Figure 4.1 shows the typical layout of a BPSTG. It consists of a

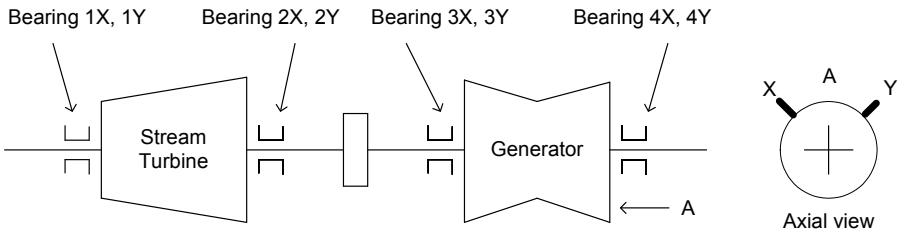


Figure 4.1: Typical layout of a BPSTG

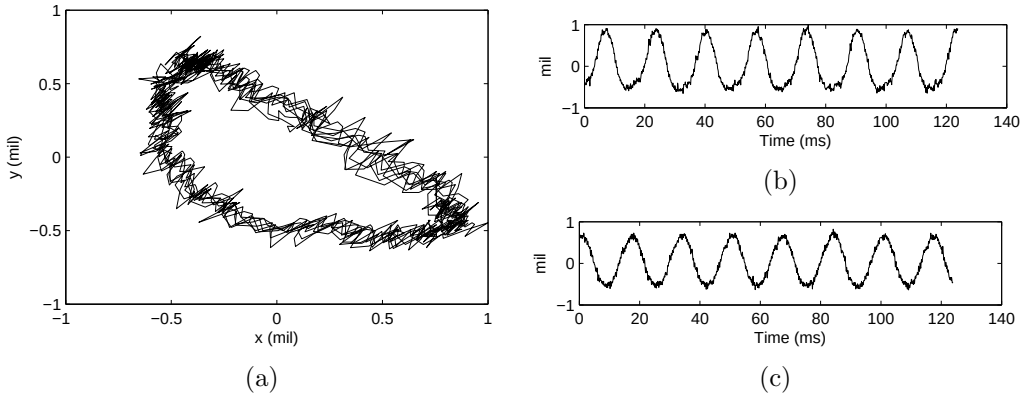


Figure 4.2: (a) typical machine vibration orbit and vibration waveforms in the (b) x and (c) y directions

steam turbine and a generator. They are 4 bearings in total for each generator, bearing 1 to bearing 4. Vibration sensors are installed on each bearing. There are 2 vibration sensors on each bearing, X and Y, and they are 90 degrees apart. During normal operation, the machines are running at 3600 RPM and the vibration sensors measure the vibration on each bearing every 2 hours. Every time when the sensors measure the machine vibration, they capture the vibration in the X and Y direction for about 8 rotating cycles. Figure 4.1 is a typical plot of the machine vibration waveforms and orbit from the vibration data measured by the sensors. Based on the vibration waveforms, many useful parameters can be extracted, such as the vibration peak-to-peak value and the amplitude of the first harmonic of the vibration waveform (1X), etc.

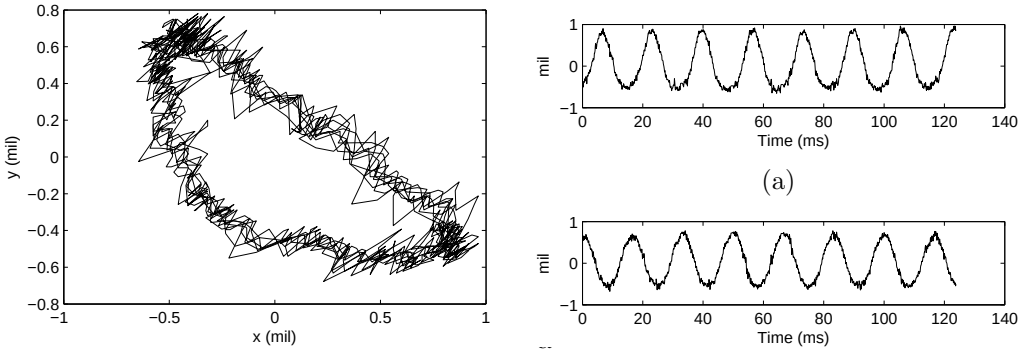
Since all 3 machines are identical, the conditions collected from 3 machines can be considered as collected from 1 machine. Four different machine conditions are considered in this thesis. Based on the on-site engineers' experience,

those 4 conditions are normal condition, unbalance, looseness, and bend shaft. Their vibration orbits and waveforms are shown in Figure 4.3 to 4.6. For each condition, 100 vibration waveforms are collected on each sensor on bearing 3 of each machine. With the collected vibration waveforms, wavelet packet decomposition can be utilized to extract features from the vibration waveforms, and then the feature dimension can be reduced by GA. Finally, NN and SVM classification models can be built with the selected features. The complete classification process is illustrated in Figure 4.7.

4.2 Feature extraction

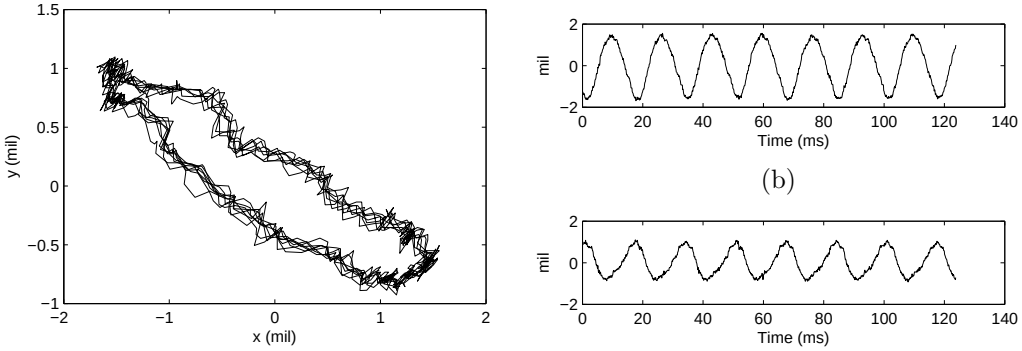
During the digitalization, the vibration waveforms are sampled at 9600 Hz, hence the analytical frequency of the Hilbert envelope spectrum is 4800 Hz [37]. Each vibration waveform is decomposed into 6 levels using WPD with wavelet function DB8. The reason why a 6-level WPD is chosen is that, after the decomposition, there are $2^6 = 64$ segments. For each segment, the frequency range is 75 Hz. Thus, each segment will contain 1 frequency which is the integer multiple of the machine rotating frequency (60 Hz), such as 60 Hz, 120 Hz, 180 Hz, etc. Also, the wavelet function DB8 is selected based on the vanishing moment. For more detail on the vanishing moment, please refer to [10]. Since there may not be any valuable information contained in the higher frequencies, segments containing higher frequencies may be discarded. Therefore, in this particular case, only the first 16 segments are retained and the rest of the segments are discarded. The first 16 segments contain the frequency ranging from 0 Hz to 1200 Hz.

Sub-signals can be reconstructed from those 16 segments and several features can be extracted from the sub-signals. In this thesis, 5 features, which are independent of loads and speeds of rotating machinery [38], are extracted from the signals. The details of those 5 features are shown below.



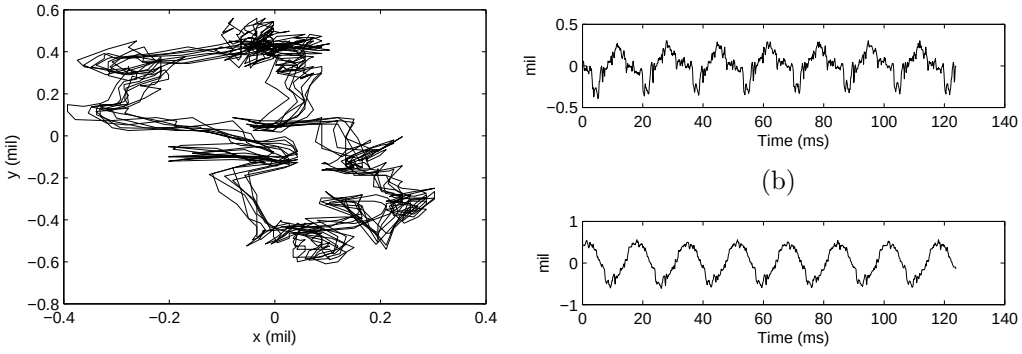
(a) (b)

Figure 4.3: (a) machine vibration orbit and waveforms in the (b) x and (c) y directions, normal



(a) (b) (c)

Figure 4.4: (a) machine vibration orbit and waveforms in the (b) x and (c) y directions, unbalance



(a) (b) (c)

Figure 4.5: (a) machine vibration orbit and waveforms in the (b) x and (c) y directions, looseness

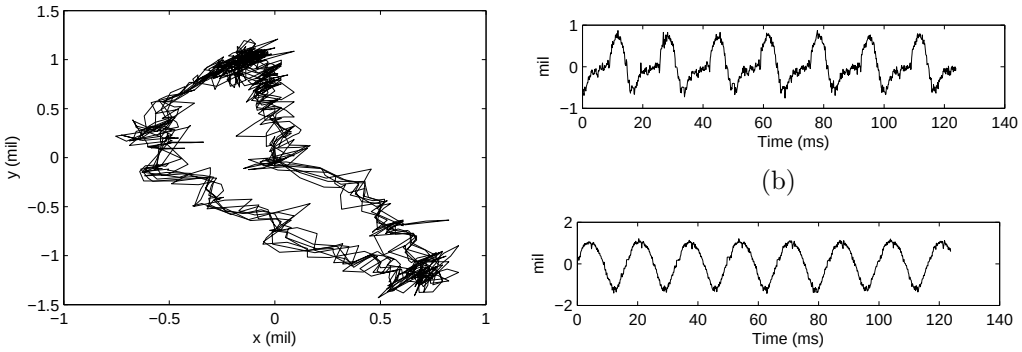


Figure 4.6: (a) machine vibration orbit and waveforms in the (b) x and (c) y directions, bend shaft

1. Skewness (SK)

$$SK = \frac{\sum_{t=1}^{T}(x_t - \mu)^3}{T\sigma^3}$$

where $x_t(t = 1, 2, ..., T)$ is the tth sample of the signal x. T is the total number of sampling points. μ is the mean value of the signal x defined as:

$$\mu = \frac{1}{T}\sum_{t=1}^{T} x_t$$

and σ is the standard deviation of x,

$$\sigma = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t - \mu)^2}$$

2. Kurtosis (KU)

$$KU = \frac{\sum_{t=1}^{T}(x_t - \mu)}{T\sigma^4}$$

3. Crest indicator (CI)

$$CI = \frac{max|x_t|}{\sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t)^2}}$$

4. Clearance indicator (CLI)

$$CLI = \frac{max|x_t|}{(\frac{1}{T}\sum_{t=1}^{T}\sqrt{|x_t|})^2}$$

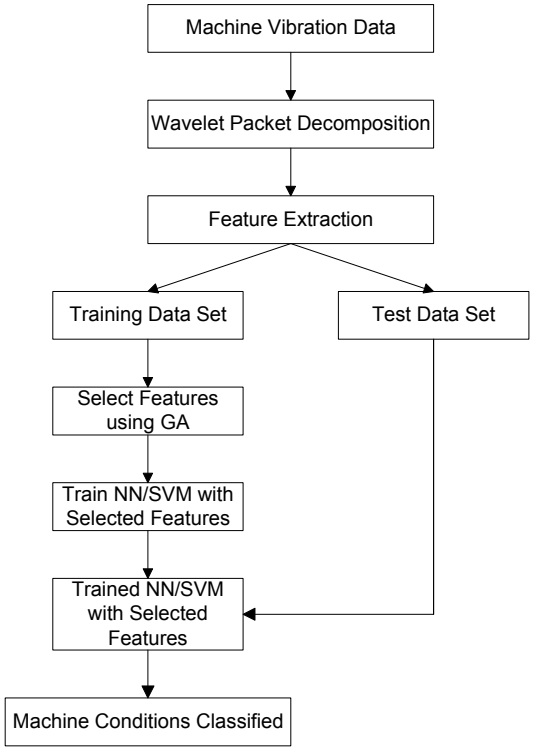


Figure 4.7: Flow chart of machine conditions classification

5. Shape indicator (SI)

$$SI = \frac{\sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t)^2}}{\frac{1}{T}\sum_{t=1}^{T}|x_t|}$$

All of the 5 features are extracted from each sub-signals and the original vibration signal. Thus, for each vibration signal, 85 features will be generated. Since the machine vibration orbit is considered in this thesis rather than focusing on the vibration signal in a single direction, and each vibration orbit consists of 2 vibration signals corresponding to 2 different directions, there will be 170 features in total for each vibration orbit.

4.3 Feature selection

In many cases, including this one, after the features are extracted from the raw data, since the feature dimension is relatively large, it is desired to reduce the feature dimension in order to reduce the training time of a classification

model while ensuring the classification accuracy is at least as good as without feature dimension reduction. There are many methods can be used to reduce the feature dimension, including principal component analysis (PCA) [15], independent component analysis (ICA) [16], and Genetic Algorithm (GA). As stated in [17], GA is one of the most efficient methods to reduce feature dimension when the feature size is large.

4.3.1 Genetic Algorithm theory

Genetic Algorithms is first introduced by John Holland in the 1970s [17]. It is an optimization process to find the best solution for a given problem. It mainly contain 3 operations: selection, genetic operation, and replacement [18]. In GA, initially, a population is generated. It consists of a subset of all possible solutions to a problem. Each member in the population is referred to as a chromosome. As stated in [18], in order to improve the performance of the algorithm, a chromosome is usually encoded in a string of variables, and each element in the string is called a gene. The variable can be represented in many forms, and due to simplicity, bit string encoding is often used [18] [19]. In bit string encoding, a gene is represented by either 0 or 1.

Each chromosome in the population will then be evaluated by a fitness function. The fitness function takes a chromosome as the input and output a number to indicate the performance of the chromosome. This is a very important step since it tells GA which chromosomes are better solution to the problem and should be used for further operations. Thus, as can be expected, designing a suitable fitness function is a crucial step in GA, and it is usually different for different applications. For example, in a curve-fitting problem, the fitness function could be a function to measure how close is the predicted value comparing to the real value. On the other hand, in a classification problem, the fitness function could be a one to produce the classification accuracy for each chromosome. With the fitness values, GA will select some chromosomes in the population for reproduction. There are many ways to select the

chromosomes, such as ranking, tournament, and proportionate scheme [18]. The rule is that the higher the fitness value of a chromosome, the higher the chance it will be selected for reproduction. In crossover, with 2 selected chromosomes, also know as the parent chromosomes, the operator will cross over those 2 chromosomes at a randomly chosen point and produces 2 offsprings. For example, with the parent chromosomes 10011000 and 11100101, if the crossover happens at the third bit, 2 offsprings will be produced, 10000101 and 11111000. Hence, each offspring has part of the genetic information of both parents. There is a probability, P_c, associated with this operation. If no crossover occurs, the offsprings will the same as the parent chromosomes. In mutation, some of the bits in the offspring may be flipped. Each bit with the probability P_m. For example, if the fourth bit is flipped in the offspring 10000101, the resulting chromosome is 10010101.

Up until this point, 2 new chromosomes are generated. If the initial population has N chromosomes, and depending on the setting, the first M chromosomes which have the highest fitness value may be directly copied to the new population to make sure the best chromosomes can survive, the above operator process will continue to run until $N - M$ new chromosomes are generated, hence a new population with size N is formed, and the performance of the chromosomes in the new population will be evaluated again by the fitness function. To summarize, a GA works as follows [19]:

1. An initial population is randomly generated with N chromosomes (possible solutions to a given problem).

2. Calculate the fitness value of the chromosomes in the population.

3. Two chromosomes are selected from the current population based on the selection strategy. The selected chromosomes will be crossed over in a randomly chosen point with the probability P_c to produce 2 new offsprings. For each new offspring, some of its bits may be flipped with the probability P_m in mutation, and 2 two new chromosomes may be

generated. If M chromosomes which have the highest fitness value in the current population are copied directly to the new population, this step is run until $N - M$ new chromosomes are generated.

4. A new population is generated. Go back to step 2 until a termination criterion is met, such as the maximum number of generations is reached, or the fitness values reaches the pre-defined value.

At the end of the process, usually some highly fit chromosomes are in the population, especially in the case when some chromosomes which have the highest fitness value are directly copied to the new population in step 3, which guarantee the survival of the best chromosome. Better results may be obtained by adjusting the GA parameters, such as increasing the number of generation and/or increasing the the size of the population.

4.3.2 Feature and model parameter selection with Genetic Algorithm

In order to apply GA, the final size of the feature dimension N, needs to be determined first. In this thesis, $N = 5$ and 20 have been tried. Other than selecting the optimum features, in this thesis, GA is also used to select the optimum parameters related to the classifiers, which are the number of neurons in the hidden layer for the NN and the RBF kernel parameter σ for SVM. The RBF kernel is defined as follow:

$$K(x, y) = e^{(\frac{-|x-y|^2}{2\sigma^2})} \tag{4.1}$$

Thus, for each chromosome in GA, there will be $N + 1$ elements. The first N elements contain the selected features from the total 170 features, and the last element is the parameter for the classifiers. The last element has to be within a certain range which is defined in advance. In this thesis, the range of the number of neurons in NN which will be selected by GA is from 15 to 35, and the range of σ in SVM is from 1 to 20, which a step size of 1. The range are

selected based on trail runs as to ensure the classifiers built with the selected parameters can achieve reasonable classification results.

Other than the final size of the feature dimension and the classifier parameter range, there are some variables needs to be set before running GA, such as the population size, crossover rate, etc. In this case, the population size has been chosen to be 10. The size has to large enough so that there will be relatively high interchange among different chromosomes [31]. The crossover rate and the mutation rate is set to be 0.5 and 0.3, respectively. The maximum number of generation and the fitness value are used as the termination criterion for the GA process. The process will stop if the maximum number of generation is reached, which is 50 in this case, or it will stop if the fitness value reaches 0.

After all the variables are defined, the final step is to design the fitness function. The features generated from the first 50 vibration orbits of each machine condition are used in GA to select the optimum features. Since there are 4 machine conditions, there will be 200 data samples, and each sample contains 170 features. In the fitness function, for each chromosome, first of all, the number of features in each data sample will be reduced based on the features selected in the chromosome. Thus, the number of features in each data sample can decrease from 170 to 20. After that, a 3-fold cross validation is used. The data samples will be separated into 2 parts. The first part contains 2/3 of the total data samples, and the other part contains 1/3 of the total data samples. The first part is used as training data and the second part is used as test data. At this point, classifiers can be built. In the Neural Network case, NN classifier is built with the training data and the number of neurons in the hidden layer selected in the chromosome, and then it is used to classify the test data. The number of misclassification will be added up until the 3-fold cross validation is over, and then the total number of misclassification will be divided by the total number of data samples to produce the fitness value of each chromosome. The procedure is similar in the SVM case, except that 4

SVM classifiers are required in order to classify all 4 machine conditions with the one-against-all classification strategy.

4.4 Performance evaluation

With the features selected from GA, classifiers can be built to classify those 4 machine conditions. Again, the selected features from the first 50 vibration orbits of each machine condition are used to train the classifiers, and the rest of the 50 samples of each machine condition will be used for validation purpose. In NN, the classifier is built with the selected number of neurons in the hidden layer. The number of input nodes is the same as the number of the selected features, and there are 4 output nodes, corresponding to 4 machine conditions. Each output node is set to be 1 if the input is classified to the condition the node is assigned to, otherwise the output node is set to be 0. Thus, during the training, the network output is a 4*200 matrix which contains either 1 or 0. In the SVM case, the classifiers are built with the RBF kernel using the selected σ. As same as building SVM classifiers in the GA fitness function, 4 SVM classifiers are required to classify 4 machine conditions, and hence there are 4 output matrices, each with size 200*1. In each matrix, a value is set to 1 if the classifier is trying to classify the corresponding input against the other 3 machine conditions. Otherwise, the value is set to 0. The classification performance of NN and SVM with and without features selection, and with different number of features, are compared below.

1. *Classification without feature selection*

   The classification results using NN and SVM without GA are shown on Table 4.1. In NN, there are 20 neurons in the hidden layer, while σ has been set to be 10 in SVM. The classification rates on the test data are 95.5 % and 94.5 % for NN and SVM, respectively. It can be seen that when all the features are used, NN performs better than SVM.

Classifier	Parameter (N/σ)	Class. rate (training) (%)	Class. Rate (testing) (%)
NN	20	100	95.5
SVM	16	100	94.5

Table 4.1: Classification results without feature selection

Classifier	# of features	Feature indices	N	Class. rate (training) (%)	Class. Rate (testing) (%)
NN without GA	170	1-170	20	9	95.5
NN with GA	20	22,23,32,36,42,43,68,81,85, 93,95,102,105,109,111, 117,140,141,149,158	19	100	95.5
NN with GA	5	81,91,94,103,166	25	100	87.5

Table 4.2: NN model classification results with and without feature selection

2. *Classification with NN*

   Table 4.2 shows the classification results using NN with and without GA. The classification result on the test data remain the same when the feature size is reduced to 20 by GA while the hidden neuron number is chosen to be 19. The classification rate decreases to 87.5% when the feature size is further reduced to 5. Also, in this case, it is noticed that although the classification rate on the test data is relatively poor, the model can classify the training data with 100 % accuracy. This validates the statement in the beginning of this chapter that when using NN, it is possible that the model is overfit to the training data and gives poor results on the test data.

3. *Classification with SVM*

   The classification results using SVM with and without GA are shown on Table 4.3. The classification rate is increased from 94.5% to 96.5% when the feature size is reduced to 20 by GA and setting $\sigma = 4$. The classification rate decreases to 95.5% when the feature size is further reduced to 5. However, the classification rate is still better than the one with SVM and without GA.

Classifier	# of features	Feature indices	σ	Class. rate (training) (%)	Class. Rate (testing) (%)
SVM without GA	170	1-170	10	100	94.5
SVM with GA	20	10,22,39,40,43,63,68,81,82, 88,97,100,101,108,120, 146,152,165,166,168	4	100	96.5
SVM with GA	5	1,16,70,105,166	3	98.5	95.5

Table 4.3: SVM model classification results with and without feature selection

4.5 Conclusion

In this chapter, artificial intelligence techniques, Neural Network and Support Vector Machine, are used to classify machine conditions. Features are first extracted from machine raw vibration data using wavelet packet decomposition, and then the dimension of the features is reduced by Genetic Algorithm. With the selected features, Neural Network and Support Vector Machine classification models are built to classify different machine conditions. From the classification results, it is shown that when using Neural Network with Genetic Algorithm, The feature dimension can be reduced from 170 to 20 while the classification result remains the same. When Support Vector Machine and Genetic Algorithm are used, the feature dimension can be reduced from 170 to 20 or even to 5, and the classification is better than using Support Vector Machine alone. Also, under different conditions, one classification model may outperform the other one. In this case, Neural Network performs better when all features are used, while Support Vector Machine produce better classification results when less features are used.

Chapter 5

Machine prognostic

Beside machine fault classification, machine prognostic is another important subject in machine condition monitoring. It is useful to predict the machine future condition based on its past and current condition so that if the predicted machine condition is unacceptable, maintenance plan can be schedule in advance and hence reduce the number of unexpected shutdown.

In order to predict the machine future condition, a time series model is required. Many techniques have been developed to build time series model. Classical approaches include autoregressive (AR) modeling and autoregressive moving average (ARMA) modeling. AR and ARMA modeling are relatively easy to use and they are suitable for building models for simple systems. However, for complex systems, AR and ARMA modeling may be found to be difficult to build accurate models [39]. Later on, as artificial intelligence developed, Neural Network (NN) has been widely used in machine prognostic. For example, In [3], Feed-Forward Neural Network (FNN) was built to predict the vibration of a rotor. In [40], Recurrent Neural Network (RNN) is built to predict machine deterioration. In 1993, adaptive-neuro-fuzzy inference system (ANFIS) was introduced by Jang [41], and the author showed that ANFIS outperforms the classical approaches and NN. In [7] and [42], ANFIS is utilized

Part of the materials in this chapter are included in 'Machine Vibration Prediction Using ANFIS and Wavelet Packet Decomposition' and submitted for publication in *International Journal of Modelling, Identification and Control*, special issue on: "Neural Networks and Fuzzy Logic for Modelling and Control of Mechatronic Systems", July 2009.

to build time series models to predict the condition of a gear system. Other than ANFIS, Support Vector Regression (SVR) is another technique which is developed recently and has been widely use in the field of machine prognostic. In [33], a hybrid model is built with SVR to predict the future state of a turbo-generator. Also, in [43], Least-Square Support Vector Machine (LS-SVM) combining with wavelet decomposition is utilized to predict the future vibration of a hydroturbine generating unit.

5.1 Time series prediction model

For a given signal $x(n)$, a time series prediction model can be expressed by the following equation:

$$x_{n+r} = f(x_{n-(m-1)k}, x_{n-(m-2)k}, ..., x_{n-2k}, x_{n-k}, x_n) \tag{5.1}$$

where x_{n+r} is the value at r time steps ahead, m is the embedding dimension, k is the time delay step, which is set to be 1 in this thesis, and f is the time series prediction model. Thus, x_{n+r} is predicted based on its previous and current values. When $r = 1$, the model is called a single step ahead prediction model. On the other hand, when $r > 1$, the model is called a multi-step ahead prediction model. In this thesis, both single step and multi-step prediction will be considered, and r is chosen to be 1, 3, and 6.

5.2 Input pre-processing

Similar to building a classification model, before building a time series model, in order to improve the model performance, it is usually preferred to pre-process the input first. The are many methods to pre-process the inputs. For example, in [44], after the model minimum embedding dimension has been decided, boosting tree algorithm are used to weight the importance of the inputs. As expected, the more recent the value in the time sequence, the more importance it is. Thus, it has higher weigh and has more effect on the model output. Another method is wavelet decomposition. In [43], wavelet

decomposition was used to decompose the machine vibration waveform first before building the time series prediction model to predict the future machine vibration. In this thesis, wavelet technique is also used to pre-process the input. However, instead of using WD to decompose the machine vibration waveform, wavelet packet decomposition is used. As shown in chapter 2, when WD is used, on each decomposition level, only the approximate coefficients will be further decomposed, while the detail coefficients will be untouched. Thus, no matter how many levels the original vibration waveform is decomposed into, the prediction model built based on the sub-signal reconstructed from the first level detail coefficients will be the same, and hence the prediction accuracy cannot be improved. On the other hand, when WPD is used to decompose the vibration waveform, both the approximate coefficients and the detail coefficients will be further decomposed, and the overall model prediction may be improved as the number of decomposition level increases.

5.3 Embedding dimension

Other than pre-processing the input, the model minimum embedding dimension is also needed to be determined before building the prediction model in order to improve the model prediction accuracy. There are 2 popular methods to determine the model minimum embedding dimension. The first one is that the embedding dimension is arbitrarily chosen at the beginning, denoted as m'. Thus, taking $m' = 6$ as an example, the original model is

$$x_{n+r} = f(x_{n-5}, x_{n-4}, x_{n-3}, x_{n-2}, x_{n-1}, x_n) \tag{5.2}$$

after that, some methods, such as the k-nearest neighbors approximation method (k-NN) and the mutual information (MI) method can be used to determine which inputs in the original model can be omitted while the model prediction accuracy stays the same or even improves. If the result shows that x_{n-4} and x_{n-3} can be omitted, the final prediction model becomes

$$x_{n+r} = f(x_{n-5}, x_{n-2}, x_{n-1}, x_n) \tag{5.3}$$

and the final minimum embedding dimension is 4. Readers can refer to [45] and [46] for more details on those input selection methods. Unlike the first method, the second method is used to determine the model minimum embedding dimension directly. This method includes the false neighbors method [47] and the Cao's method [48]. In this thesis, Cao's method is used to determine the minimum embedding dimension of the time series prediction model and its basic theory is reviewed in the following section.

5.3.1 Determining the embedding dimension

Consider a time series $x(n)$, where $n = 1, 2, ..., N$. The time delay vector is defined as

$$\begin{aligned} y_i(m) &= (x_i, x_{i+k}, ..., x_{i+(m-1)k}), \\ i &= 1, 2, ..., N - (m-1)k, \end{aligned} \tag{5.4}$$

where $y_i(m)$ is the ith reconstructed vector with embedding dimension m. Also, another parameter is defined as

$$a(i, m) = \frac{||y_i(m+1) - y_{n(i,m)}(m+1)||}{||y_i(m) - y_{n(i,m)}(m)||} \tag{5.5}$$

where $||\cdot||$ is the Euclidian distance, $y_{n(i,m)(m)}$ is the nearest neighbor of $y_i(m)$ in terms of the Euclidian distance, and $n(i, m)$ is an integer which $1 \leq n(i, m) \leq N - m\tau$. If $y_{n(i,m)(m)} = y_i(m)$, the second nearest neighbor will be used so that the denominator in Eq. (5.5) will not be 0.

When any two points stay close in the m and $m + 1$ dimensional reconstructed space, m is qualified as an embedding dimension. However, as can be expected, it is difficult to choose a threshold value T so that when two points have a value $a(i, m)$ which $a(i, m) < T$, those two points can be said to be close to each other. Also, different time series may have different threshold

values. Thus, instead of trying to find an appropriate threshold value for a given time series, the author in [48] defined a quantity as

$$E(m) = \frac{1}{N - mk} \sum_{i=1}^{N-mk} a(i, m) \tag{5.6}$$

and

$$E1(m) = E(m + 1)/E(m) \tag{5.7}$$

It is found out that $E1(m)$ will keep changing as m increases until m reaches a certain value m_0, and $m_0 + 1$ will the minimum embedding dimension.

5.4 Time series prediction model with SVR and WPD

In this thesis, for a given time series $x(n)$, in order to build a prediction model to predict its value at x_{n+r}, $x(n)$ is first decomposed into N levels using WPD. After the decomposition, for each coefficient vector in the Nth level, sub-signal can be reconstructed. Cao's method is then utilized to determine the minimum embedding dimension for each sub-signal. With the embedding dimension, time series model can be built for each sub-signal using SVR, and the final prediction output is the sum of the outputs of all the models. The complete process is illustrated in Figure 5.1. The overall performance of the model prediction is evaluated based on the root mean square error(RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (x(k) - \hat{x}(k))^2} \tag{5.8}$$

where n is the total number of data point predicted and $\hat{x}(k)$ is the predicted value at time k.

5.5 Case study

In this section, single-step and multi-step time series prediction models are built to predict the future vibration of the machine G2. Figure 5.2 shows the

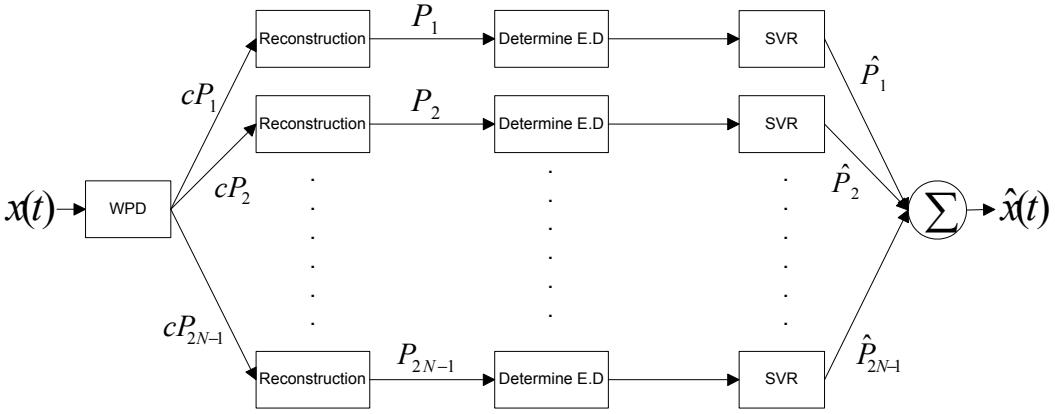


Figure 5.1: Process of building time series prediction model with SVR and WPD

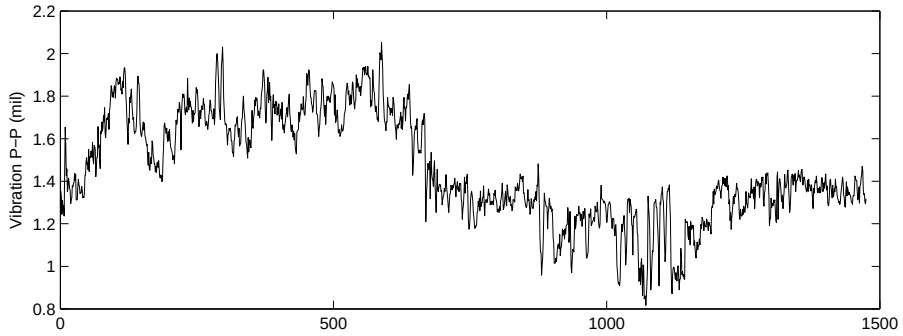


Figure 5.2: Machine vibration peak-to-peak values

vibration peak-to-peak values on bearing 4 in the x direction of machine G2. The vibration data were measured during the period from May to Aug. 2003 and there are 1475 data points in total.

As mentioned earlier, the vibration data are measured every 2 hours. With the raw vibration data, time series prediction model can be built. The first step is to decompose the vibration signal into different levels using WPD. In this thesis, the decomposition level ranges from 2 to 5 and the DB8 wavelet function is used during the decomposition. Figure 5.3 shows the reconstructed sub-signals for a 2-level decomposition.

With the reconstructed sub-signals, minimum embedding dimension can be determined by Cao's method. When applying Cao's method, the algorithm

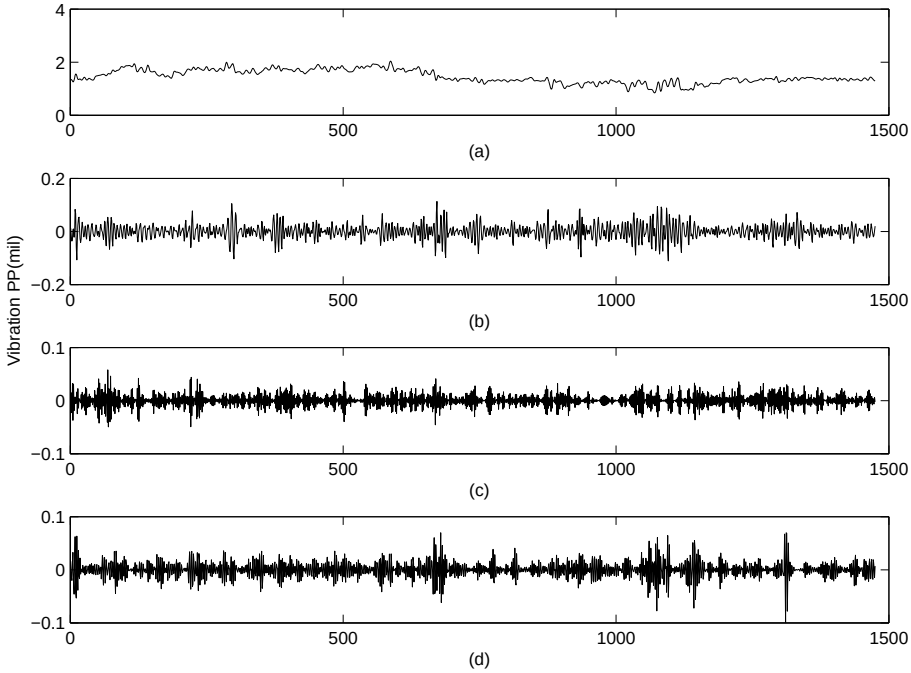


Figure 5.3: Sub-signals reconstructed from a 2-level WPD of the original vibration signal, (a): $P_{2,0}$, (b): $P_{2,1}$, (c): $P_{2,2}$, (d): $P_{2,3}$

will try to calculate the parameter $E1(m)$ for $m = 1$ to 19 using the first 500 points from the vibration data. Based on the calculated $E1(m)$, the minimum embedding dimension for each sub-signal can be determined. The embedding dimensions for all sub-signals reconstructed from 2 to 5-level WPD are shown on Table 5.1. Figure 5.4 shows the plot of $E1(m)$ for all 4 sub-signals reconstructed from the wavelet packet coefficients after a 2-level WPD is applied to the original vibration signal. Based on this figure, the embedding dimensions for those 4 sub-signals can be selected as 9, 7, 6, and 6, respectively.

After the minimum embedding dimensions are determined, time series prediction models are ready to be built with SVR. Since the first 500 points of the vibration data are used to determine the minimum embedding dimension, those 500 data points are also used to train the SVR models. Assuming the embedding dimension is 6, when training a single-step prediction model, the training input matrix x and output matrix y will be

Sub-sig.	M	Sub-sig.	M	Sub-sig.	M	Sub-sig.	M	Sub-sig.	M
$P_{2,0}$	9	$P_{4,0}$	6	$P_{4,12}$	7	$P_{5,8}$	5	$P_{5,20}$	6
$P_{2,1}$	7	$P_{4,1}$	10	$P_{4,13}$	7	$P_{5,9}$	4	$P_{5,21}$	7
$P_{2,2}$	6	$P_{4,2}$	8	$P_{4,14}$	7	$P_{5,10}$	7	$P_{5,22}$	7
$P_{2,3}$	6	$P_{4,3}$	7	$P_{4,15}$	7	$P_{5,11}$	4	$P_{5,23}$	7
$P_{3,0}$	6	$P_{4,4}$	5	$P_{5,0}$	6	$P_{5,12}$	7	$P_{5,24}$	5
$P_{3,1}$	10	$P_{4,5}$	5	$P_{5,1}$	5	$P_{5,13}$	7	$P_{5,25}$	7
$P_{3,2}$	5	$P_{4,6}$	7	$P_{5,2}$	10	$P_{5,14}$	9	$P_{5,26}$	7
$P_{3,3}$	7	$P_{4,7}$	7	$P_{5,3}$	7	$P_{5,15}$	7	$P_{5,27}$	7
$P_{3,4}$	10	$P_{4,8}$	4	$P_{5,4}$	8	$P_{5,16}$	6	$P_{5,28}$	7
$P_{3,5}$	8	$P_{4,9}$	5	$P_{5,5}$	8	$P_{5,17}$	8	$P_{5,29}$	7
$P_{3,6}$	7	$P_{4,10}$	9	$P_{5,6}$	6	$P_{5,18}$	6	$P_{5,30}$	7
$P_{3,7}$	7	$P_{4,11}$	9	$P_{5,7}$	8	$P_{5,19}$	6	$P_{5,31}$	7

Table 5.1: Embedding dimensions for sub-signals reconstructed from 2 to 5-level WPD

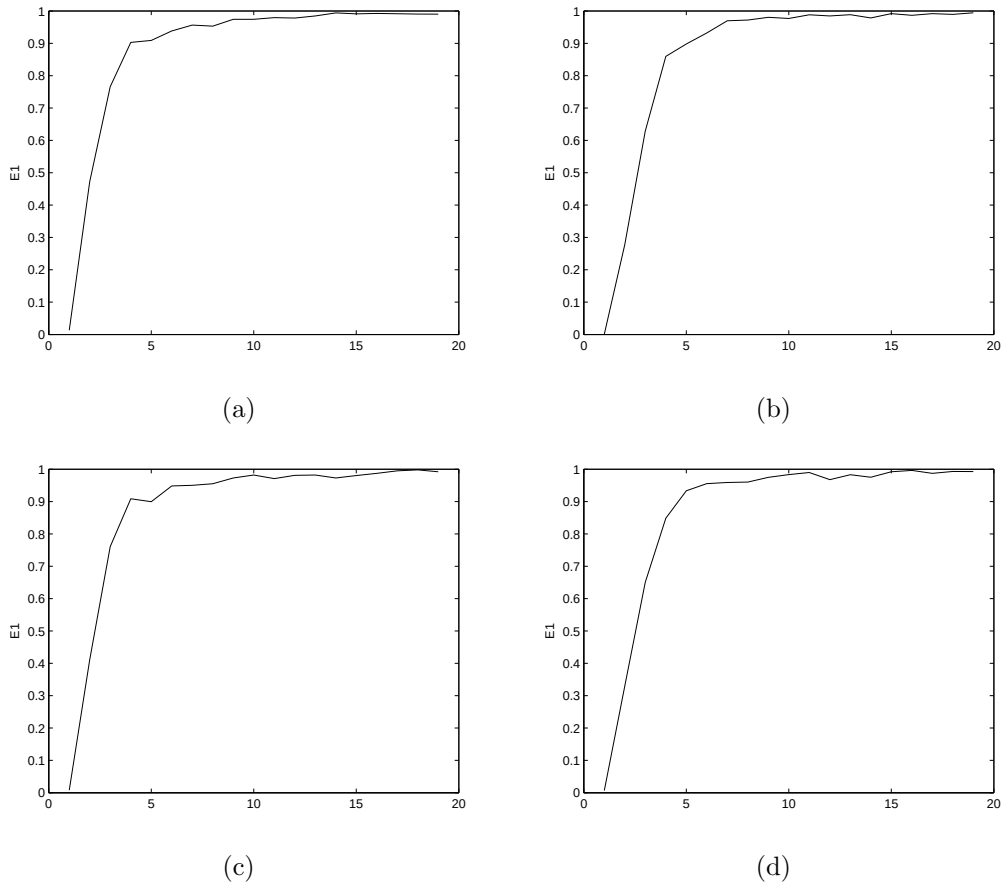


Figure 5.4: Embedding dimensions for sub-signals reconstructed from (a): $cP_{2,0}$, (b): $cP_{2,1}$, (c): $cP_{2,2}$, (d): $cP_{2,3}$

# step ahead prediction	Levels of WPD	RMSE
1	2	0.0094
	3	0.0062
	4	0.0055
	5	0.0053
3	2	0.0411
	3	0.0217
	4	0.0186
	5	0.0147
6	2	0.1061
	3	0.0509
	4	0.0301
	5	0.0231

Table 5.2: Prediction results for different number step ahead predictions with SVR and different levels of WPD

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{493} & x_{494} & x_{495} & x_{496} & x_{497} & x_{498} \\ x_{494} & x_{495} & x_{496} & x_{497} & x_{498} & x_{499} \end{bmatrix} \quad y = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ \vdots \\ x_{499} \\ x_{500} \end{bmatrix}$$

The rest of the 975 data points are used to test the SVR models. By trial and error, all the SVR prediction models are built with the polynomial kernel function, which can be expressed by the following equation,

$$K(x, x') = (x^T x' + 1)^d \tag{5.9}$$

where d is the kernel parameter, degree, and it is set to be 1 in this case. The trade-off parameter C is set to be 100, and ϵ is set to be 0.001. The model prediction errors for different numbers of step ahead prediction with different levels of WPD are shown on Table 5.2, and the prediction results with 5-level WPD are plotted in Figure 5.5 against the real vibration values. From Table 5.2, it can be seen that the RMSE increases as the number of step ahead prediction increases with the same level of WPD, which is expected. Also, for the same number of steps ahead prediction, as the level of WPD increases, the RMSE decreases.

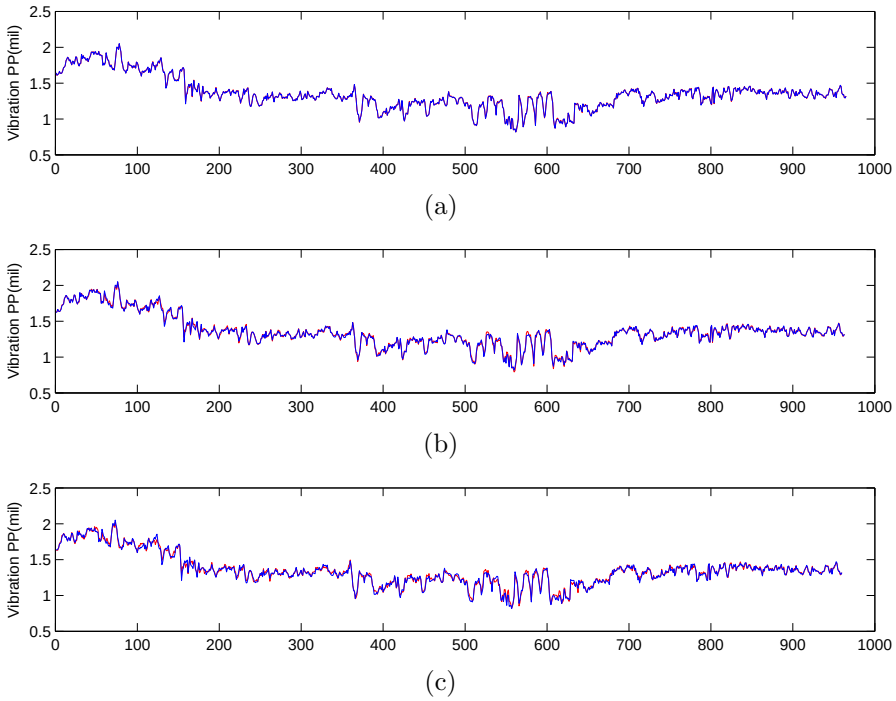


Figure 5.5: Vibration prediction results using SVR and 5-level WPD: predicted values (red), actual values (blue): (a) 1-step ahead prediction, (b) 3-step ahead prediction, (c) 6-step ahead prediction

For comparison, two other methods, building time series model using SVR alone and building time series model using SVR with WD, are also used to build models to prediction the future machine vibration. Again, the first 500 data points are used to determine the minimum embedding dimensions. The minimum embedding dimensions for all sub-signals reconstructed from 2 to 5-level WD are shown on Table 5.3, and the minimum embedding dimension is 10 when the model is built with SVR only. Table 5.4 and Figure 5.6 shows the results using SVR alone, and Table 5.5 and Figure 5.7 shows the results using SVR with WD. Comparing Table 5.2 to Table 5.5 and 5.4, it is clear that time series model built with SVR and WPD gives the best prediction results, while the model built with SVR alone gives the worse prediction results. This highlights the importance of pre-processing the raw data before building the prediction models. Also, from Table 5.5, it can be notice that for any of the 3 cases, when the level of WD increases from 4 to 5, The RMSE

Sub.sig. 2-level WPD	M	Sub.sig. 3-level WPD	M	Sub.sig. 4-level WPD	M	Sub.sig. 5-level WPD	M
A_2	9	A_3	6	A_4	6	A_5	6
D_1	6	D_1	6	D_1	6	D_1	6
D_2	7	D_2	7	D_2	7	D_2	7
		D_3	10	D_3	10	D_3	10
				D_4	10	D_4	10
						D_5	5

Table 5.3: Embedding dimensions for sub-signals reconstructed from 2 to 5-level WD

# step ahead prediction	RMSE
1	0.0522
3	0.0955
6	0.1233

Table 5.4: Prediction results for different number step ahead prediction with SVR alone

does not change too much. There is actually a little increase in RMSE in the single step and 3-step ahead prediction cases, while the RMSE is decreased by 1.65% in the 6-step ahead prediction case. Applying the same analysis to Table 5.2, it is found out that, except for the single step ahead prediction case, the RMSE is decreased by at least 21% for the other 2 cases when the decomposition level increases from 4 to 5. This clearly shows the advantage WPD has over WD. As stated before, in WD, only the approximate coefficients will be further decomposed. When the decomposition level is large enough that a nearly perfect prediction model may be built for the sub-signal reconstructed from the approximate coefficients, further decompose the original signal may not have any significant impact on the overall model prediction. At this stage, while a sufficient accurate model may be able to build for the sub-signal reconstructed from the approximate coefficients, accurate models probably cannot be built for the sub-signals reconstructed from the detail coefficients. Also, it is usually more difficult to built accurate prediction model for sub-signals reconstructed from the detail coefficients since those signals contain higher frequencies and higher non-linearity. On the other hand, this is not the case in WPD. Since detail coefficients will also be further decomposed, it is possible

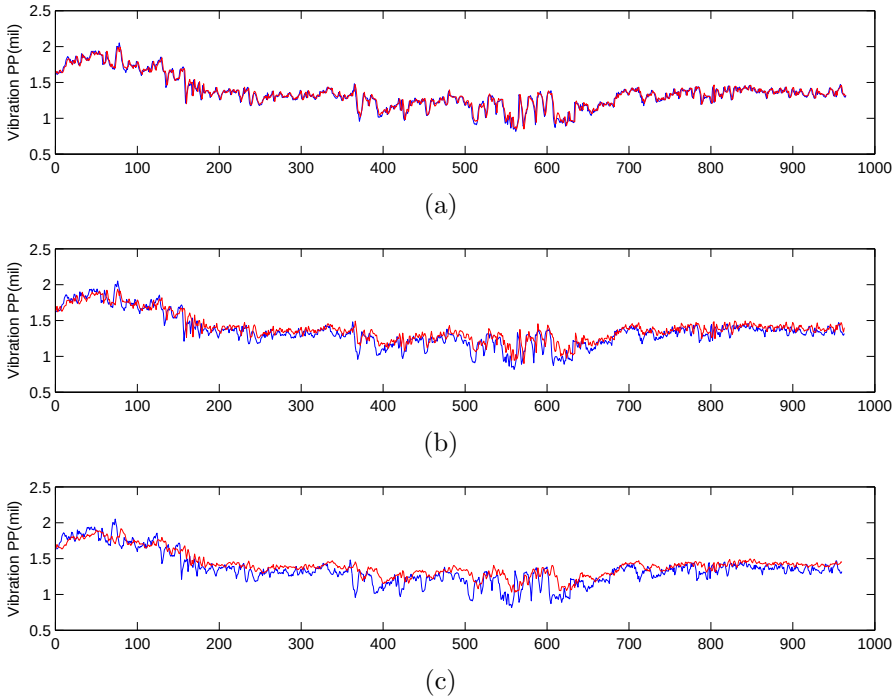


Figure 5.6: Vibration prediction results using SVR alone: predicted values (red), actual values (blue): (a) 1-step ahead prediction, (b) 3-step ahead prediction, (c) 6-step ahead prediction

# step ahead prediction	Levels of WD	RMSE
1	2	0.0113
	3	0.0107
	4	0.0106
	5	0.0108
3	2	0.044
	3	0.0289
	4	0.0275
	5	0.0276
6	2	0.1058
	3	0.0565
	4	0.0428
	5	0.0417

Table 5.5: Prediction results for different number step ahead prediction with SVR and different levels of WD

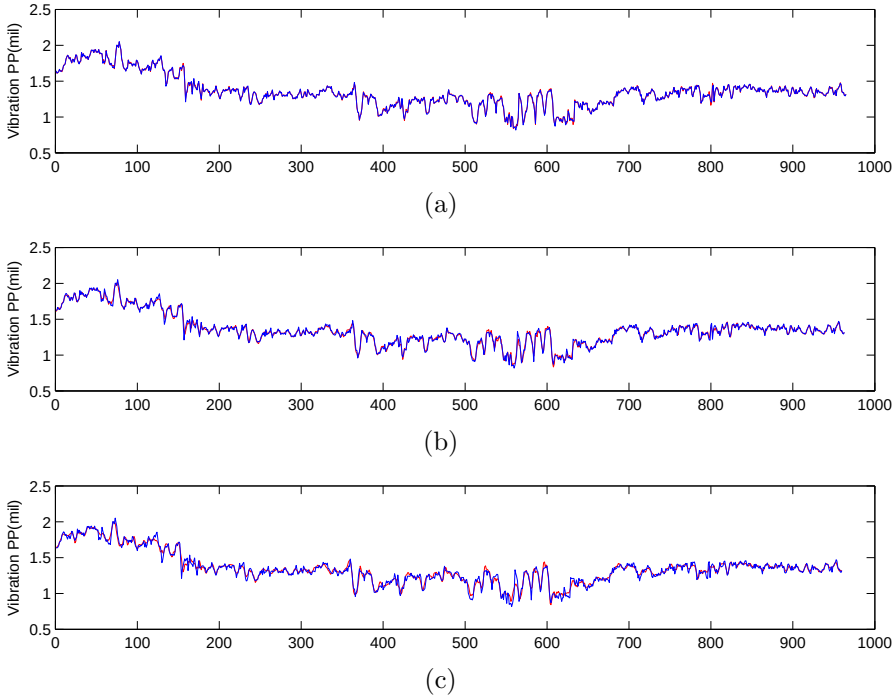


Figure 5.7: Vibration prediction results using SVR and 5-level WD: predicted values (red), actual values (blue): (a) 1-step ahead prediction, (b) 3-step ahead prediction, (c) 6-step ahead prediction

to build accurate prediction model for the sub-signals reconstructed from the detail coefficients. Thus, in a situation which a sufficient accurate model can be built for the sub-signal reconstructed from the approximate coefficients, further decompose the original signal can still improve the overall prediction result because better models can be built for the sub-signals reconstructed from the detail coefficients.

5.6 Conclusion

In this chapter, single and multi-step ahead time series prediction models have been built to predict the future machine vibration. With the collected vibration signal, wavelet packet decomposition is first utilized to decompose the signal. Sub-signals can be reconstructed from the approximate and detail coefficients and the minimum embedding dimension can be determined for

each sub-signal using Cao's method. Time series model is then built for each sub-signal using Support Vector Regression. The overall prediction result is the sum of the outputs from all SVR models. Comparison has been make to the other 2 methods, building prediction model using SVR alone and using SVR with wavelet decomposition. The result shows that the method using SVR with WPD outperforms the other 2 methods.

Chapter 6

Machine thermal sensitivity analysis

In the previous chapter, SVR combining with WPD is utilized to built time series models to predict the machine future vibration based on the machine past and current vibrations, and the results are promising. In this chapter, SVR is used again to build models for the machine system. However, instead of using it to predict the machine vibrations in the future, it is used to predict the current machine vibrations with the machine output power as the model inputs. By calculating the difference between the predicted values and the real values, it is possible to keep track of the machine condition and see how the condition is changing as time progresses. More specifically, SVR is used to keep track of the generator rotor condition due to thermal sensitivity. The general concepts of generator rotor thermal sensitivity is reviewed in the next section followed by the current practice in industry regarding generator rotor thermal sensitivity and how SVR can be applied to keep track of the generator rotor condition regarding this issue.

6.1 Review on machine thermal sensitivity

In this section, the basic theory and some of the common causes of generator rotor thermal sensitivity are introduced. Readers are encouraged to consult with [49] and [50] for for details. Generator rotor thermal sensitivity is a

phenomenon which the rotor vibration is changed when the generator field current is increased. As stated in [49], even for a rotor which has thermal sensitivity issue, it is not affected when the generator is operating with the power factor higher than 0.85 lagging or with a leading power factor. On the other hand, when the generator is operating with a power factor lower than 0.85 lagging, a thermal sensitivity rotor will be affected and its vibration will change. The rotor vibration may increase, decrease, or its phase angle may change. Therefore, even with a thermal sensitivity rotor, a generator may not have any issues when operating with low field current; however, its operation may be limited at high field currents or VAR loads as the rotor vibration excesses the acceptable limit.

6.1.1 Types of thermal sensitivity

Generator rotor thermal sensitivity can be classified into 2 types: reversible and irreversible. When the thermal sensitivity is reversible, rotor vibration changes as field current varies. That is, when the field current increases, the rotor vibration increases. Later on, when the field current decreases, the rotor vibration will decrease as well. This type of thermal sensitivity usually does not cause major problems in practice and the rotor can be balanced so that its maximum vibration will not excess the limit. If the rotor vibration does not decrease after the field current is reduced, this type of thermal sensitivity is called irreversible. This type of thermal sensitivity is troublesome since the rotor vibration will keep increasing, and the rotor may have to be taken off-line and repaired in order to reduce the vibration.

6.1.2 Causes of thermal sensitivity

As indicated in [49], one of the main reasons why a rotor is thermally sensitive is that the copper conductors in the winding and the steel field forging have different coefficients of expansion. When field current is applied, although both the copper and the steel forging will try to expand, due to different coefficients

of expansion, the copper will try to expand more. Thus, as the field current increase, the difference in expansion between those two can become quite large and the generated forces can be quite large as well. Eventually, these forces can cause the rotor to bow if they are not distributed uniformly, and the bowing will cause the vibration to change. If other factors are excluded, the bowing should vary as the field current changes and it should be reversible. However, due to the complexity of a generator, there are many factors can affect the thermal vibration, including shorted turns, blocked ventilation or unsymmetrical cooling, insulation variation, wedge fit, distance block fitting, etc. Some of these factors will be briefly explained further in the following section. Readers can refer to [49] for more information on the causes of thermal sensitivity.

- Shorted Turns: Shorted turns is the most common cause of thermal sensitivity. For a field which has shorted turns, when field current is applied, the pole which has higher number of shorts will have lower temperature comparing to the other pole. This is because the pole which has higher number of shorts has lower electrical resistance. As a result, the pole with higher temperature will trend to expand more than the other pole, and hence causes the rotor to bow. As can be expected, the amount of bow is directly related to the field current, and the thermal sensitivity caused by shorted turns is reversible.

- Blocked Ventilation or Unsymmetrical Cooling: Blocked ventilation and unsymmetrical cooling are quite similar to shorted turns. In block ventilation, a foreign object may be involved to disrupt the normal ventilation and cooling of the field. Unsymmetrical cooling is caused by shifting of the insulation or plugging of cooling passages. Both blocked ventilation and unsymmetrical cooling are result in uneven temperature distribution in the field and cause the rotor to bow. When the temperature drops, the rotor may restore to its original form. Therefore, thermal sensitivity caused by blocked ventilation and unsymmetrical cooling are reversible.

- Insulation Variation: When the insulation thickness and buildup are not even in a field, it can cause binding and uneven friction forces in the coil slots and under the retaining rings. If this happens, the field coils may not be able to expand uniformly and hence the field forging may be loaded unevenly and cause the rotor to bow. The bow will increase when increasing the field current. In some cases, the rotor may not be able to restore to its original form when the field current decreases. This is due to the fact that the binding of the coils may persist. Thus, thermal sensitivity caused by insulation variation may be irreversible.

6.1.3 Thermal sensitivity test

In order to test if a rotor has thermal sensitivity problem, a thermal sensitivity test can be performed. The purpose of the test is to isolate the machine vibration which is caused by MW loading from the vibration caused by VAR loading. As stated clearly in [49], vibration changing with MW loading does not indicate the rotor has thermal sensitivity problem. Also, from the previous sections, it can be seen that thermal sensitivity has a very important relationship with the field current. The thermal sensitivity test consists of 3 parts:

1. The thermal sensitivity test is started by loading the generator with small MW and MVAR, 10MW and 0MVAR for example, and then MW will be increased to about 60% of its rated value and MVAR will be reduced. During the test, before going from 1 stage to another stage, it is very important to ensure the generator has reached a steady state, usually it takes about 15 to 30 minutes for each stage, and all the important readings, such as the machine vibration, voltage, current, temperature, etc, should be carefully recorded.

2. In the second part of the test, the generator MW will be kept constant while the field current will be continuously increased until it reaches its rated value. Thus, MAVR will be increased in this part. As mentioned

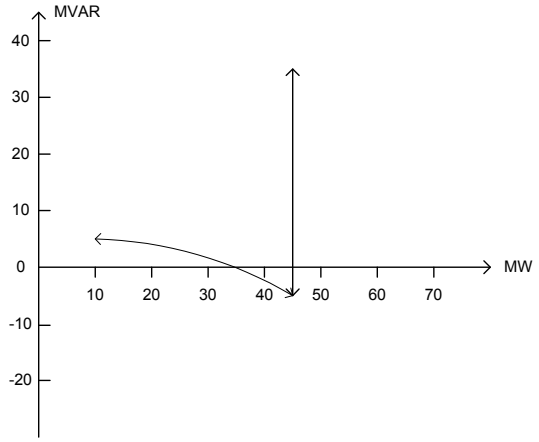


Figure 6.1: Typical plot of machine output power during a thermal sensitivity test

earlier, for a thermal sensitivity rotor, its vibration starts to change when the generator is operating at power factor lower than 0.85 lagging. Hence, it is very important in this part to increase the MVAR high enough so that the generator will be operating with a power factor lower than 0.85 lagging. If the field current cannot reach its rated value without having the machine vibration excess its acceptable limit, this part of the test should be repeated with the maximum allowed field current.

3. The last part of the thermal sensitivity test is the reverse of the first 2 parts. The generator MVAR will be decreased while keeping the MW constant, and then the MW will be reduced and MVAR will be increased, so that the final generator MW and MVAR will be the same as they were when the test is started. The complete process of the thermal sensitivity test are illustrated in Figure 6.1. If the final machine vibration is similar to the vibration when the test is started, it can be concluded that the thermal sensitivity is reversible. On the other hand, if the final machine vibration does not reduce to its started level and remains high, the thermal sensitivity is irreversible and further maintenance actions may need to be taken.

6.1.4 Industry practice with thermal sensitivity

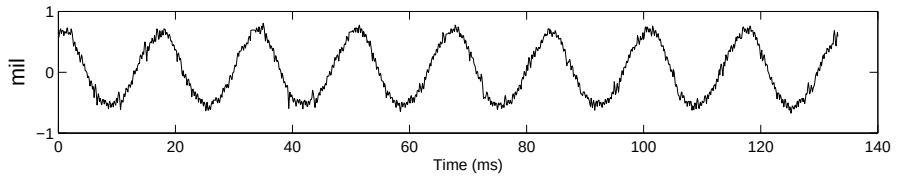
In the local oil-sand company, thermal sensitivity test is performed on all 3 BPSTGs on a yearly basis. The thermal sensitivity test serves 2 purposes. The first one is, as mentioned previously, to compare the machine vibration at the beginning of the test to the machine vibration at the end of the test and thus determine if the thermal sensitivity is reversible or not. The other purpose is to determine how large is the difference between the vibration at the beginning of the test and the vibration when the generator is operating with the highest MW and MVAR during the test. The difference has to be within a certain limit otherwise the generator will not be able to run on its full capacity. The method used to calculate the difference between those 2 vibrations is explained in detail below.

1. During the thermal sensitivity test, at each stage, the machine vibration peak-to-peak value and its phase can be recorded. However, it is believed that the vibration due to thermal bow is mainly shown on 1X, which is 60 Hz in this case; therefore, in order to eliminate the other effects, the 1X vibration peak-to-peak value is used. Figure 6.2 shows a typical machine vibration waveform along with its 1X component only. Thus, every cycle in the 1X vibration waveform in the x and y direction can be expressed by an cosine equation:

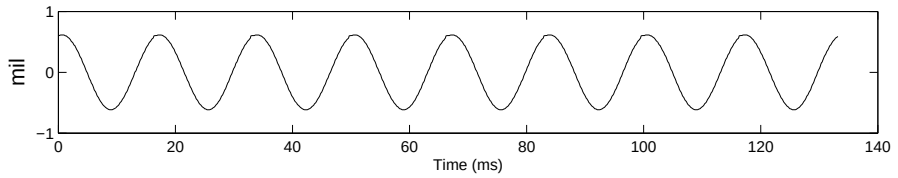
$$\begin{aligned} V_x &= \frac{1}{2} A_x cos(\theta - \theta_x) \\ V_y &= \frac{1}{2} A_y cos(\theta - \theta_y) \\ 0 &\leq \theta < 2\pi \end{aligned} \tag{6.1}$$

   where A_x, A_y, θ_x, and θ_y are the 1X vibration peak-to-peak value and phase angle in the x and y direction, respectively, and they can all be recorded during the thermal sensitivity test.

2. θ_y will be subtracted by $\pi/2$ (or added by $3\pi/2$ if $\theta_y - \pi/2 < 0$) since



(a)



(b)

Figure 6.2: Machine vibration waveform, (a) unfiltered, (b) 1X only

the vibration sensor x and y are 90^o degree apart. Thus,

$$\begin{aligned} \bar{\theta_y} &= \theta_y - \pi/2 \\ V_y &= \frac{1}{2} A_y cos(\theta - \bar{\theta_y}) \end{aligned} \tag{6.2}$$

3. In order to ensure V_x and V_y are larger than 0, constant terms, $\frac{1}{2}A_x$ and $\frac{1}{2}A_y$ will be added to V_x and V_y, respectively. Hence,

$$\begin{aligned} \bar{V_x} &= \frac{1}{2} A_x + \frac{1}{2} A_x cos(\theta - \theta_x) \\ \bar{V_y} &= \frac{1}{2} A_y + \frac{1}{2} A_y cos(\theta - \bar{\theta_y}) \end{aligned} \tag{6.3}$$

4. Finally, by iteration, a θ can be found which maximizes the following equation,

$$V_T = \sqrt{\bar{V_x}^2 + \bar{V_y}^2} \tag{6.4}$$

   The corresponding phase angle can be denoted as θ_T. At this point, the overall maximum machine vibration can be expressed by a vibration vector with magnitude V_T and phase angle θ_T.

By following the procedures outlined above, the maximum vibration vector can be calculated for the machine vibration at the start of the thermal sensitivity test and at the point when the machine is operating at the highest MW

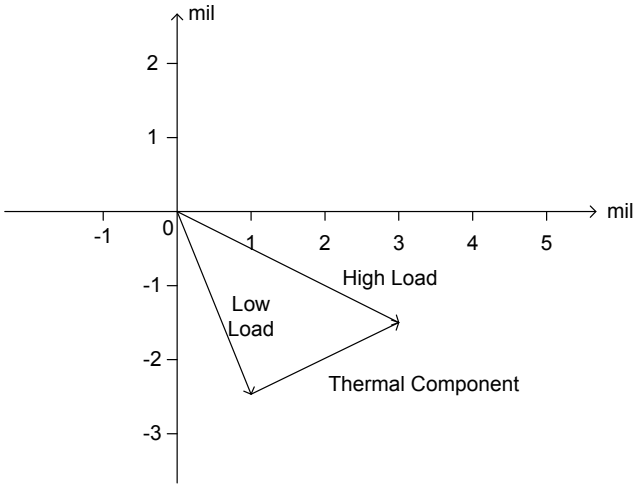


Figure 6.3: Typical plot of the machine 1X vibration vector during a thermal sensitivity test

and MVR during the test, and then the vibration difference between those 2 conditions can be calculated. Figure 6.3 is a typical plot of the vibration vectors during a thermal sensitivity test.

6.1.5 Limitation of current practice on thermal sensitivity

As indicated earlier, currently, the thermal sensitivity test serves 2 purposes, one is to determine if the machine thermal sensitivity is reversible or not. The other purpose is to see how large is the vibration difference between the machine is operating at low load (beginning of the test) and at high load (the middle point of the test). However, how the machine vibration changes due to thermal sensitivity in a long term has not been taken into consideration. As mentioned earlier, if the machine thermal sensitivity is irreversible, after the thermal sensitivity test, the machine vibration will be higher than the vibration at the beginning of the test, and the vibration will probably keep increasing as time progresses. Thus, thermal sensitivity test is destructive and the machine condition may become worse after a thermal sensitivity test. Also, when a machine is undergo a thermal sensitivity test, it has to be removed from the production line, and hence reduce the productivity. It would be

better to find a way to determine if a machine has thermal sensitivity issue or not from the machine regular operational data. In the next section, one of the artificial intelligence techniques, SVR, will be utilized to keep track of the machine condition and provide some preliminary information on whether the machines have thermal sensitivity problem.

6.2 Machine vibration tracking with SVR

In order to keep track of the machine vibration, a system model is required. The inputs of the model will be the generator output real power and reactive power, and the output of the model will be the machine 1X vibration. Other than the machine output power, many other factors, such as the temperature of the machine operating environment, may also have impacts on the machine vibration. However, machine output power can be directly controlled by the on-site engineers, and this is why they are chosen as the inputs of the model. The model will try to predict the machine vibration based on the generator output power, which can be mathematically expressed as

$$y = f(P, Q) \tag{6.5}$$

where p and Q are the machine real and reactive power, respectively, and y is the machine 1X vibration. If the model is properly trained and the machine thermal sensitivity is irreversible, the difference between the predicted vibration and the real vibration may become larger and larger as time progresses. Instead of using the magnitude or phase angle of the vibration vector as the model output, the vibration vector can be separated into 2 parts by the following 2 simple equations:

$$\begin{aligned} V_{TX} &= V_T cos\theta_T \\ V_{TY} &= V_T sin\theta_T \end{aligned} \tag{6.6}$$

Thus, any changes in the magnitude and phase angle of the machine 1X vibration will be noticed in V_{TX} and V_{TY}.

6.2.1 Case studies

In this section, the generators G1 and G2 will be investigated. Based on the previous thermal sensitivity test results, G1 does not seem to be suffered from serious thermal sensitivity problem and the thermal sensitivity is reversible. On the other hand, G2 may have serious thermal sensitivity issue and it is irreversible. Both generators will be considered separately in the following sections.

6.2.2 G1

Figure 6.4 shows the plots of V_{TX} and V_{TY} of G1. The vibration data is obtained during the period from Jan. to Aug. 2003 on bearing 4 and there are 2761 data points in total. From Figure 6.4, no obvious trend can be noticed. V_{TX} and V_{TY} do not seem to increase or decrease as time progresses. In order to confirm that the machine condition did not change during that period, SVR models can be built to predict the machine vibration based on the machine output power. If the machine condition indeed did not change during that period, the model predicted vibration should be very close to the real vibration as long as the model is properly trained.

When building the SVR models in this section, the polynomial kernel function is selected and the kernel parameter, degree, is set to be 2 in this case. The kernel function and its parameter are chosen based on trial and error. The first 700 data points are used to train the SVR models and the prediction error is simply the difference between the predicted vibration value and the real vibration value:

$$error = Vibration_{real} - Vibration_{predicted} \tag{6.7}$$

Figure 6.5 and 6.6 shows the SVR model prediction results along with the real machine vibration and the prediction error. It can be seen that, for both V_{TX} and V_{TY}, the prediction results are very close to the real values and the prediction error stays at around 0 all the time. Also, from the error plots,

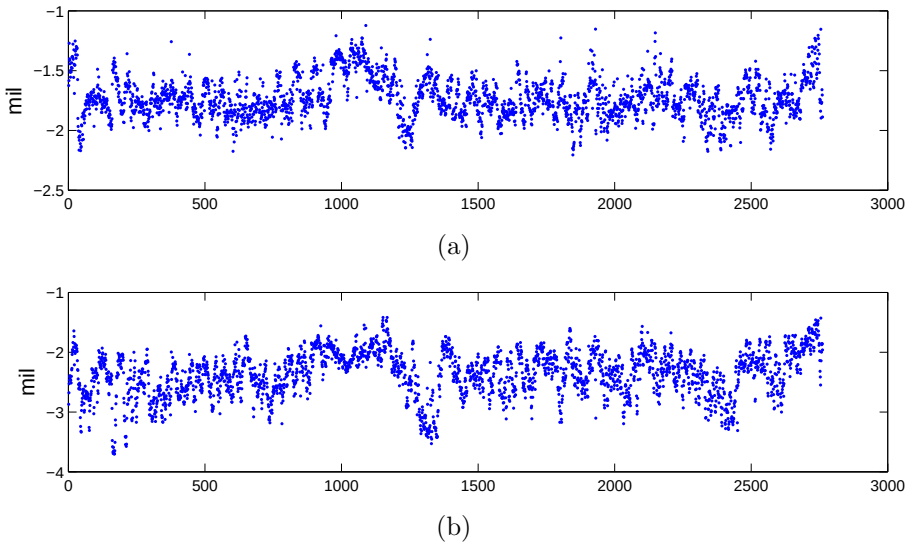


Figure 6.4: Plots of (a) V_{TX} and (b) V_{TY}, G1

there is no clear trend that the prediction error increases or decreases as time progresses. Therefore, it can be concluded that the condition of G1 did not change during the period from Jan. to Aug. 2003, and a thermal sensitivity test around this period may not be necessary for G1. From the plots, it can also be concluded that there is a direct relationship between the machine output power and the machine 1X vibration. It is possible to build an accurate model with machine real and reactive powers as the model inputs to predict the machine 1X vibration.

6.2.3 G2

Similar analysis can be applied to generator G2. Figure 6.8 shows the plots of V_{TX} and V_{TY} of G2. The vibration data is obtained from Jan. to Sep. 2003 on bearing 3 and there are 3123 data points in total. SVR models are built with the same kernel function and parameter for V_{TX} and V_{TY}, and again the first 700 data points are used to trained the models. The prediction results and prediction errors are shown on Figure 6.9 and 6.10. From Figure 6.9, the prediction results are close to the actual values and the prediction errors are around 0 for all data points. On the other hand, on Figure 6.10, starting

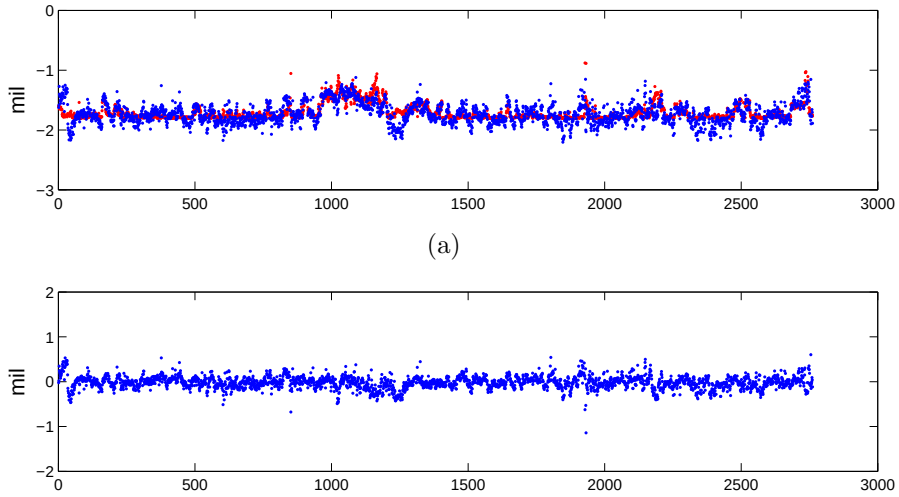


Figure 6.5: (a) SVR model prediction results for V_{TX}, predicted values (red), actual values (blue), and (b) prediction error, G1

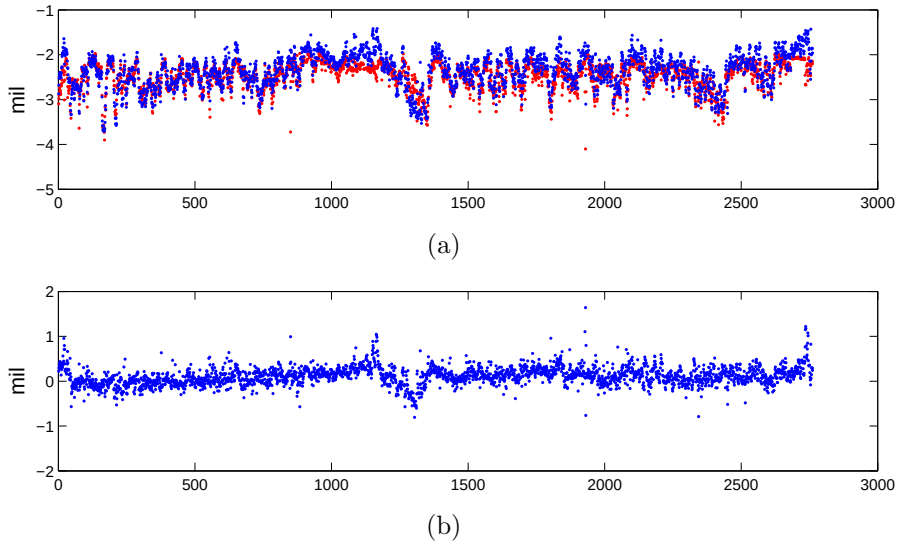


Figure 6.6: (a) SVR model prediction results for V_{TY}, predicted values (red), actual values (blue), and (b) prediction error, G1

from data points around 1920, which corresponding to June 23, 2003 in actual date, the actual vibration starts to increase, which causes the prediction error between the predicted V_{TY} and the actual V_{TY} starts to increase and finally settles down at data points around 2200, which corresponding to July 16, 2003 in actual date. If the vibration vectors are plotted during the period from June 23 to July 16, the result would be similar to Figure 6.7. V_{TX} did not change too much during that period and it remained at about 2.5 mil, while V_{TY} increased approximately from -1 to 1 mil. Thus, the vibration vector went from the forth quadrant to the first quadrant, and the magnitude of the vibration vector actually decreased first and then increased. This is the reason why it is preferred to separate the vibration vector into V_{TX} and V_{TY} instead of considering the magnitude and phase angle of the vibration vector. During a short period of time, if there is a noticeable trend in V_{TX} or V_{TY}, it would be either increasing or decreasing. There are 2 possible reasons that may explain why the actual vibration increases. The first one is that the actual vibration increases after data point 1920 is due to the increase of generator output powers, which are the inputs of the SVR model, and the SVR model cannot produce close results after the increase of the inputs. However, since the model prediction values are very close to the actual values for the first 1800 data points, it can be confirmed that the SVR model has been trained properly and it should produce outputs accordingly if the inputs are increased. Also, by checking the generator output during the period from Jan. to Sep. 2003, the output powers are always fluctuating between 10 to 50 MW and 10 to 20 MVAR. Thus, this may not be the true reason why the actual vibration increases. The second reason is that the machine condition has changed. One way to change the machine condition is that the machine has been taken off-line and some maintenance activities has been preformed on the machine. However, G2 was continuously running during the period from Jan. to Sep. 2003 without any shutdown or maintenance. The other way which may cause the change of machine condition is that the machine

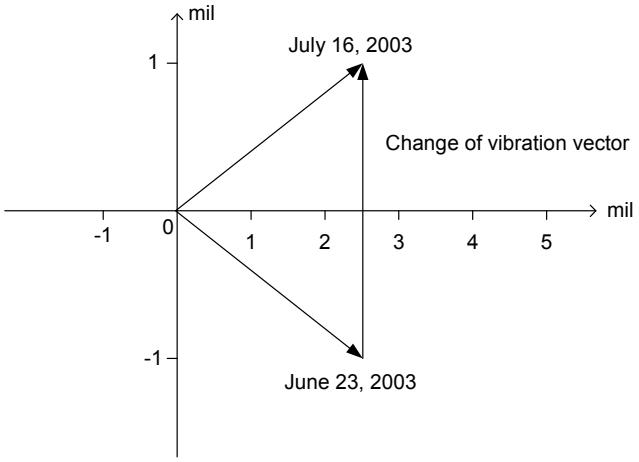


Figure 6.7: Change of vibration vector, G2

has thermal sensitivity problem and it is irreversible. When the machine is operating at high MVAR/field current, its vibration will change and start to increase. By checking the generator output powers, it is found out that, on June 23, 2003, the generator was operating with very high MVAR, such as 25MW and 30MVAR, 45MW and 30MVAR, etc. Thus, although further investigation may be required, at this point, it is reasonable to assume that the vibration change is due to thermal sensitivity and a thermal sensitivity test can be scheduled to confirm this.

6.3 Conclusion

In this chapter, the general ideas of machine thermal sensitivity and the current industry practices regarding machine thermal sensitivity are reviewed. Support Vector Regression is utilized again in this chapter to build system model to predict the machine vibration based on the machine output power. The system model is used to keep track of the machine condition and provide some preliminary information on whether the machine has irreversible thermal sensitivity issue or not. Experimental results show that generator G1 may not have any thermal sensitivity problem, while G2 may have. These results agree with the previous thermal sensitivity test results.

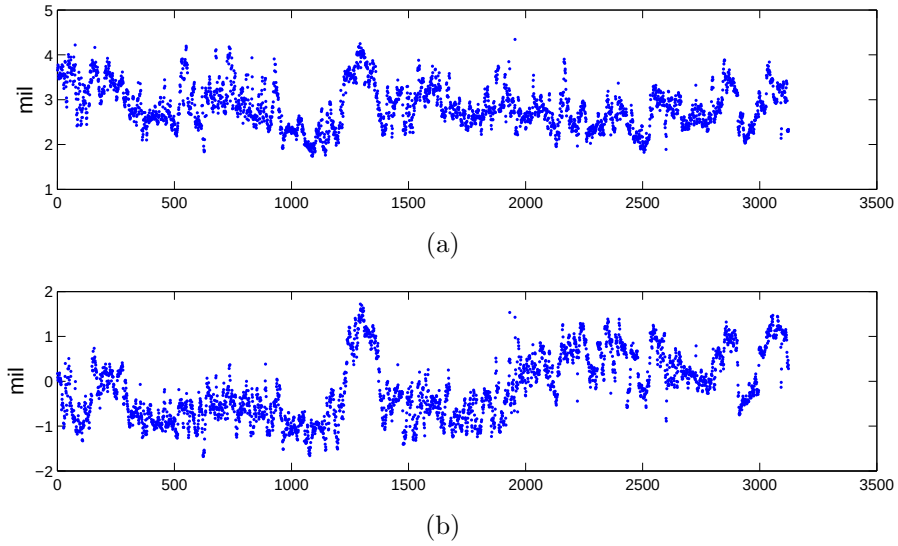


Figure 6.8: Plots of (a) V_{TX} and (b) V_{TY}, G2

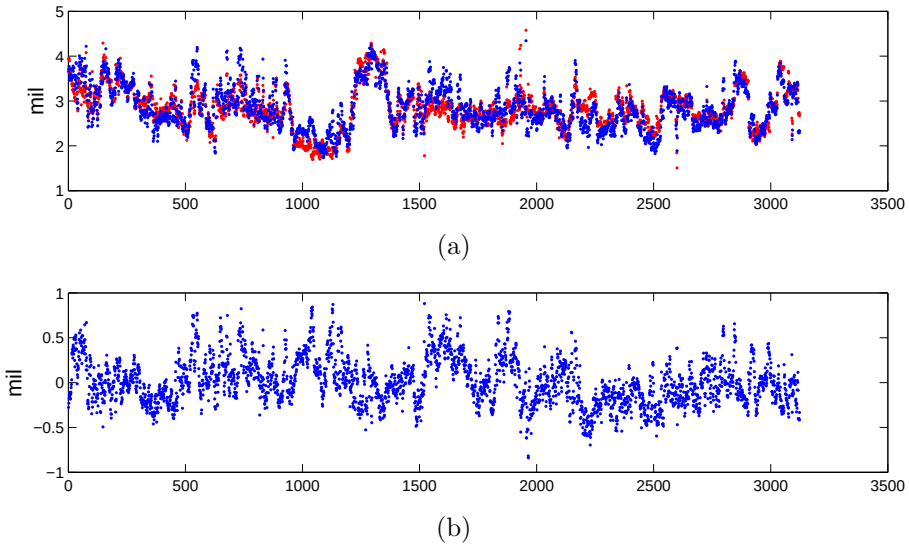


Figure 6.9: (a) SVR model prediction results for V_{TX}, predicted values (red), actual values (blue), and (b) prediction error, G2

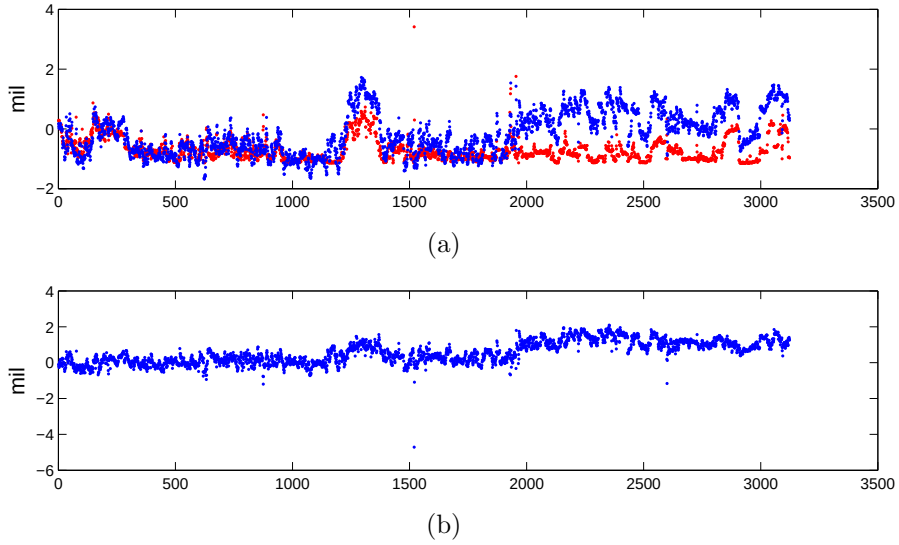


Figure 6.10: (a) SVR model prediction results for V_{TY}, predicted values (red), actual values (blue), and (b) prediction error, G2

Chapter 7

Conclusion and future work

7.1 Conclusion

In this thesis, artificial intelligence techniques are applied to the field of electrical machine condition monitoring. The thesis consists of 3 main parts. In the first part of the thesis, Neural Network and Support Vector Machine, combining with wavelet packet decomposition and Genetic Algorithm, are utilized to build classification models to classify different machine conditions. Experimental result obtained with either method is excellent, although under different conditions, one method may perform better than the other one. In the second part, single step and multi-step ahead time series prediction models are built with Support Vector Regression and wavelet packet decomposition to predict the future machine vibration based on the past and current machine vibration. Prediction results are compared to the results obtained with the other 2 methods, building time series models with SVR alone and with SVR and discrete wavelet decomposition. The comparison shows that the method using SVR and WPD outperforms the other 2 methods. In the last part of the thesis, system model is built with SVR. The model inputs are the machine output power while the model output is the machine 1X vibration. The model tries to map the input to the output and hence keeps track of the machine vibration and provides some useful information to determine if the machine has thermal sensitivity problem or not.

7.2 Future work

Electrical machine condition monitoring is an on going research subject and the future research directions in this project may include the followings. For the machine condition classification part, data representing more machine conditions may be collected so that the classifier can classify more machine conditions. Different feature extraction/selection methods may be used in order to further improve the classification result. Regarding building time series model to predict machine future vibration, more accurate models are required in order to make more steps ahead prediction. The more steps ahead prediction can be make, the more time the on-site experts will have in advance to schedule a maintenance plan. For machine thermal sensitivity, future research may include further investigation on the relationship among machine output power, machine thermal sensitivity, and machine vibration, so that when there is a significant change in machine vibration, it can be determined if it is due to machine thermal sensitivity problem.

Bibliography

[1] A.C. McCormick and A.K. Nandi, 'Classification of the rotating machine condition using artificial neural networks', *Proceedings of the Institution of Mechanical Engineers*, Vol. 211, Part C, pp.439–450, 1997.

[2] C.T. Kowalski and T. Orlowska-Kowalska. 'Neural networks application for induction motor faults diagnosis', *Mathematics and Computers in Simulation* Vol. 63, pp.435–448, 2003.

[3] X. Li, S.Y. Pei, Z. Han, and L. Qu. 'Fault prognosis for large rotating machinery using neural network', *Transactions on Information and Communications Technologies*, Vol. 6, pp.99–105, 1994.

[4] Z. Jiang, H. Fu, and L. Li, 'Support Vector Machine for mechanical faults classification', *Journal of Zhejiang University Science*, Vol. 6A(5), pp.433–439, 2005.

[5] B. Samanta, "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms", *Mechanical Systems and Signal Processing*, Vol. 18, pp.625–644, 2004.

[6] A. Widodo and B. Yang. 'Application of nonlinear feature extraction and support vector machines for fault diagnosis of induction motors', *Expert Systems with Applications*, Vol. 33, pp.241–250, 2007.

[7] C. Nataraj and B. Samanta, 'Prognostics of machine condition using soft computing', *Robotics and Computer-Integrated Manufacturing*, Vol. 24, pp.816–823, 2008.

[8] B. Samanta and K.R. Al-Balushi. 'Artificial Neural Network Based fault Diagnostics of Rollong Elemental Bearings using Time-Domain Features', *Mechanical Systems and Signal Processing*, Vol. 17(2), pp.317–328, 2003.

[9] G. Betta, C. Liguori, A.Paolillo, and A. Pietrosanto. 'A DSP-Based FFT-Analyzer for the Fault Diagnosis of Rotating Machine Based on Vibration Analysis', *IEEE Transactions on Instrumentation and Measurement*, Vol. 51(6), pp.1316–1322, 2002.

[10] J. Sanza, R. Pererab, and C. Huerta. 'Fault diagnosis of rotating machinery based on auto-associative neural networks and wavelet transforms', *Journal of Sound and Vibration*, Vol. 302, pp.981–999, 2007.

[11] R. Yan and X. Gao. 'An efficient approach to machine health diagnosis based on harmonic wavelet packet transform', *Robotics and Computer-Integrated Manufacturing*, Vol. 21, pp.291–301, 2005.

[12] P.S. Addision. *The Illustrated Wavelet Transform Handbook*, London: The Institute of Physics, 2002.

[13] J. Poshtan and J. Zarei 'Bearing fault detection using wavelet packet transform of induction motor stator current', *Tribology International*, Vol. 40, pp.763–769, 2007.

[14] I.A. Antoniadis and N.G. Nikolaou 'Rolling element bearing fault diagnosis using wavelet packets', it NDT& E International, Vol. 35, pp.197-205, 2002

[15] Q. He, R. Yan, F. Kong, and R. Du. 'Machine condition monitoring using principal component representations', *Mechanical Systems and Signal Processing*, Vol. 23(2), pp.446–466, 2009.

[16] A. Widodo, B. Yang, and T. Han. 'Combination of independent component analysis and support vector machines for intelligent faults diagnosis

of induction motors', *Expert Systems with Applications*, Vol. 32, pp.299–312, 2007.

[17] L. Zhang and A.K. Nandi. 'Fault classification using genetic programming', *Mechanical Systems and Signal Processing*, Vol.21, pp.1273–1284, 2007.

[18] K.S. Tang, K.F. Man, S. Kwong, and Q. He, 'Genetic Althorithms and Their Applications', *IEEE Signal Processing Magzine*, Vol. 13(6), pp.22–37, 1996.

[19] M. Mitchell, *Introduction to Genetic Algorithms*, The MIT Press, London, 1999.

[20] A. Bernieri, M. DApuzzo, L. Sansone, and M. Savastano. 'A Neural Network Approach for Identification and Fault Diagnosis on Dynamic Systems', *IEEE Transaction On Instrumentation And Measurement*, Vol. 43(6), 1994.

[21] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.

[22] K.L. Lo and Z. Zakaria. 'Electricity consumer classification using artificial intelligence', *39th International Universities Power Engineering Conference*, Vol. 1, pp.443–447, 2004.

[23] A.S.Vieira, B. Ribeiro, S. Mukkarnala, J.C. Neves, and A.H. Sung. 'On the Performance of Learning Machines for Bankruptcy Detection', *2nd IEEE International Conference on Computational Cybernetics*, pp.323–327, 2004.

[24] M. Grassi1 and M. Faundez-Zanuy. 'Face Recognition with Facial Mask Application and Neural Networks', *Lecture Notes in Computer Science*, Vol. 4507, pp.709–716, 2007.

[25] G. Ou, Y. Murphey, and L. Feldkamp. 'Multiclass Pattern Classification Using Neural Networks', *Proceedings of the 17th International Conference on Pattern Recognition* Vol. 4, pp.585–588, 2004.

[26] S.J. Russell and P.Norvig, *Artificial Intelligence, A Modern Approach*, Pearson Education, Inc., New Jersey, 2003.

[27] M. Acikkar and M.F.Akay. 'Support vector machines for predicting the admission decision of a candidate to the School of Physical Education and Sports at Cukurova University', *Expert Systems with Applications*, Vol. 36, pp.7228–7233, 2009.

[28] W.Chen, C.Ma, and L.Ma. 'Mining the customer credit using hybrid support vector machine technique', *Expert Systems with Applications*, Vol. 36, pp.7611–7616, 2009.

[29] V. Saravanan and R. Mallika. 'An effective classification model for cancer diagnosis using micro array Gene expression data', *International Conference on Computer Engineering and Technology*, Vol. 1, pp.137–141, 2009.

[30] S. Abe, *Support Vector Machine for Pattern Classification*, Springer, London, 2005.

[31] B. Samanta, K.R. Al-Balushi, and S.A. Al-Araimi, 'Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection', *Engineering Applications of Artificial Intelligence*, Vol. 16, pp.657–665, 2003.

[32] A. Widodo, B. Yang. 'Support vector machine in machine condition monitoring and fault diagnosis', *Mechanical Systems and Signal Processing*, Vol. 21, pp.2560–2574, 2007.

[33] L. Xiang, G.J. Tang, and C. Zhang. 'Simulation of time series prediction based on hybrid support vector regression', *Proceedings - 4th International Conference on Natural Computation*, Vol, 2, pp.167–171, 2008.

[34] Alex J. Smola and B. Scholkopf, 'A tutorial on support vector regression', *Statistics and Computing*, Vol. 14, pp.199–222, 2004

[35] S.R. Gunn, 'Support Vector Machines for Classification and Regression', Technical Report, Faculty of Engineering, Science and Mathematics, School of Electronics and Computer Science, University of Southampton, 1998.

[36] M.H. Sadeghi, J. Raflee, F. Arvani, and A. Harifi, ' A Fault Detection and Identification System for Gearboxes using Neural Networks', *International Conference on Neural Networks and Brain*, Vol. 2, pp.964–969, 2005.

[37] Q. Hua, Z. He, Z. Zhang, and Y. Zi. 'Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble', *Mechanical Systems and Signal Processing*, Vol. 21, pp.688–705, 2007.

[38] Y. Lei, Z. He, and Y. Zi. 'Application of an intelligent classification method to mechanical fault diagnosis', *Expert Systems with Applications*, Vol. 36(6), pp.9941–9948, 2009.

[39] J. Liu, W. Wang, and F. Golnaraghi, 'A multi-step predictor with a variable input pattern for system state forecasting', *Mechanical Systems and Signal Processing* Vol. 23, No. 5, pp.1586–1599, 2009.

[40] P.W. Tse and D.P. Atherton. 'Prediction of Machine Deterioration Using Vibration Based Fault Trends and Recurrent Neural Networks', *Journal of Vibration and Acoustics*, July, Vol. 121, pp.355–362, 1999.

[41] J.R. Jang. 'ANFIS: adaptive-network-based fuzzy inference system', *IEEE Transactions on Systems, Man, and Cybernetics*, May/June, Vol. 23, No. 6, pp.665–685, 1993.

[42] W. Wang, and F. Golnaraghi, F. Ismail. 'Prognosis of machine health condition using neuro-fuzzy systems', *Mechanical Systems and Signal Processi*, Vol. 18, pp.813–831, 2004.

[43] M. Zou, J. Zhou, Z. Liu, and L. Zhan, L. 'A Hybrid Model for Hydroturbine Generating Unit Trend Analysis', *Proceedings - Third International Conference on Natural Computation*, Vol. 2, pp.570–574, 2007.

[44] A. Kusiak, H. Zheng, and Z. Song. 'Short-Term Prediction of Wind Farm Power: A Data Mining Approach', *IEEE Transactions On Energy Conversion*, Vol. 24, No. 1, pp.125–136, March 2009.

[45] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse. 'Methodology for long-term prediction of time series', *Neurocomputing*, Vol. 70, pp.2861–2869, 2007.

[46] Y. Ji, J. Hao, N. Reyhani, and A. Lendasse. 'Direct and Recursive Prediction of Time Series Using Mutual Information Selection', *Computational Intelligence and Bioinspired Systems. 8th International Work Conference on Artificial Neural Networks*, pp.1010–1017, 2005.

[47] M.B. Kennel, R. Brown, and H.D.I. Abarbanel. 'Determining embedding dimension for phase-space reconstruction using a gemetrical construction', *Physical Review A*, Volume 45, No. 6, pp.3403–3411, 1992.

[48] L. Cao. 'Practical method for determining the minimum embedding dimension of a scalar time series', *Physica D*, Vol. 110, pp.43–50, 1997.

[49] R.J. Zawoysky and W.M. Genovese. 'Generator Rotor Thermal Sensitivity-Theory and Experience', GE Power Systems, New York, 2001.

[50] D.J. Petty. ' Analysis of bearing vibration monitoring', *IEE Colloquium on Understanding your Condition Monitoring*, pp.4/1-4/11, 1999.