AFRL-AFOSR-JP-TR-2018-0047

Investigation of Change Detection and Forecasting Methods for Predictive Maintenance

Manabu Tsunokai
NIHON NO SHORAIWO KANGAERU KAI, N.P.O.

02/21/2018
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ IOA
Arlington, Virginia 22203
Air Force Materiel Command

FORM SF 298

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
24-05-2018	Final	24 Sep 2015 to 23 Sep 2017

4. TITLE AND SUBTITLE
Investigation of Change Detection and Forecasting Methods for Predictive Maintenance

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA2386-15-1-4121

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Manabu Tsunokai

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NIHON NO SHORAIWO KANGAERU KAI, N.P.O.
IMONIKENOHATA BLDG. 10F., 2-7-17, IKENOHATA, TAITO
TOKYO, 110-0008 JP

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR IOA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-JP-TR-2018-0047

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this project, the research team evaluated the applicability of neural networks to the detection of abnormalities in equipment and developed a method using simulation and experimental data. The team evaluated how neural networks classify or estimate data deviating from the training data. Conclusions were drawn for both the classification and regression problems. Neural networks were applied to acoustic data of a pump, and the detectability of abnormalities was evaluated. The research focused on the development of algorithms for forecasting the movement of advanced images such as the human lung and to test the feasibility of an organ motion monitoring algorithm to detect anomalies during treatment in near real time. All work was performed with the available images in our data base.
In the case of forecasting using PCA/MSSA, further calculation time reduction can be achieved by using this method in combination with other methods such as optical flow by which only few components can be tracked for the tracking of the entire region of displacement in the image. The combination is also believed to reduce the noise appearing in longer sequences.

15. SUBJECT TERMS
AOARD, SSA, SST

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			KNOPP, JEREMY
Unclassified	Unclassified	Unclassified	SAR	91	19b. TELEPHONE NUMBER *(Include area code)* 315-227-7006

Investigation of change detection and forecasting methods for predictive maintenance

Final Report 1

NIHON NO SHORAIWO KANGAERU KAI

Dec.18, 2017

Contents

This is the final report 1 on the basic research for AOARD entitled "Investigation of change detection and forecasting methods for predictive maintenance"

1. Introduction

1.1. Goal

It is desirable in many industries to reduce the burden of maintenance of aging infrastructure by transitioning to condition-based maintenance (CBM). In order to make this transition successful, change detection and forecasting methods that can be applied to complex mechanical systems are required.

Although there are various methods that are considered to be effective for predictive maintenance, their practical applications have not progressed much because there is no guidance for choosing, customizing, and developing a method according to targets and based on the evaluation of its capability.

The goal of this research is to establish such a guidance and also to develop actual methods for specific targets.

1.2. Summary of the past research

Multiple change detection methods by machine learning were investigated to clarify their characteristics. Although each method has a different way of modeling, they all have a common structure (see Fig. 1-1).

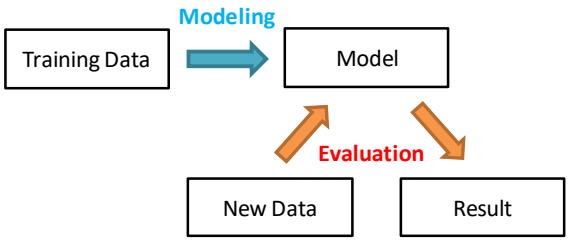


Fig. 1-1 Common structure of change detection methods

The research of last year proposed a common procedure for calculating an abnormality score that is independent of the methods. With this procedure, the five following methods were evaluated using simulation and experimental data.

- Linear Regression (LR)
- Kernel Regression (KR)
- Mahalanobis Distance (MD)
- Kernel Density Estimation (KDE)

- Similarity Based Modeling (SBM)

Furthermore, in order to intuitively understand how each method models data, a common visualization method of the model for two dimensional data was developed and the characteristics of each method were compared by visualizing the model (see Fig. 1-2).

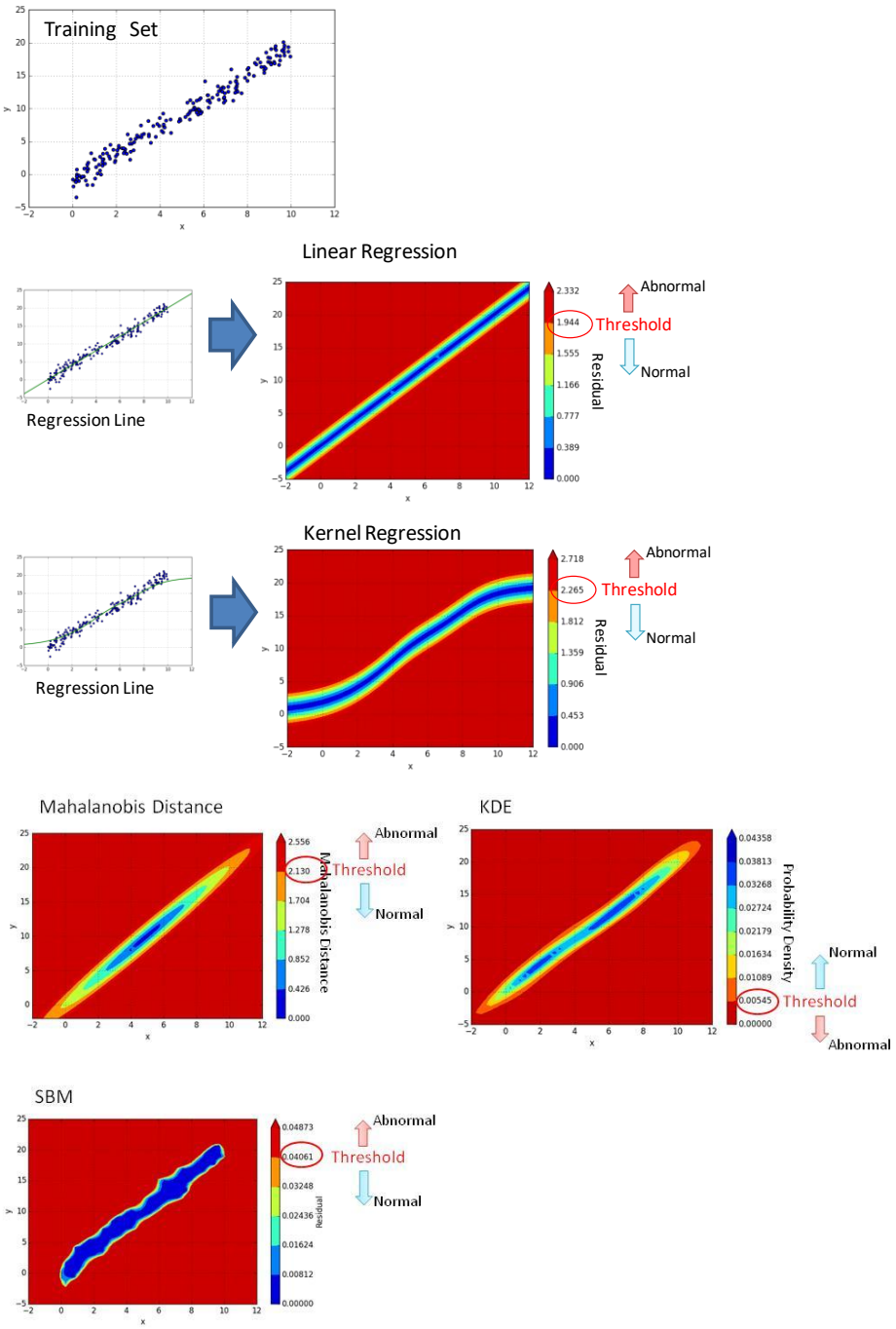


Fig. 1-2 Visualization of models

The features of each method and their optimal use, derived from those investigations, are modified according to the results of this year's research and summarized in section 4.2

1.3. Investigation of Neural Networks

In this year's research, we focused on Neural Networks (NN), and evaluated their applicability to the detection of abnormalities in equipment. With recent improvements in computer performance, NN have become remarkable in fields such as image recognition. However, there are not many studies from the viewpoint of their application to the detection of abnormalities in equipment.

In this research, we not only evaluated basic characteristics of NN from the viewpoint of their application to the detection of abnormalities in equipment, but also developed a new method to determine the state of equipment by applying a technique used in image recognition to acoustic data.

1.4. Outline of Neural Networks

A neural network is a computing system imitating the neural circuits of the human brain. The human brain transmits information by connecting a large number of cells called neurons through multiple layers. The behavior of each neuron is as follows.

1) Receives stimuli from other neurons from input terminals called dendrites,
2) The electric potential inside the neuron gradually increases,
3) When the potential exceeds a threshold, it outputs the neurotransmitter from the axon.

Furthermore, there is a difference in the strength of the connection between neurons. Some neurons receive large stimulations, while others receive small stimulations. Adjusting the strength of this transmission of stimulations is the foundation of learning and memory of the brain.

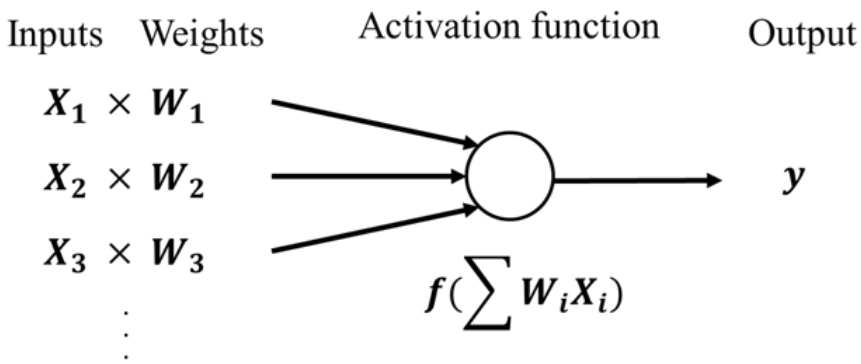
On the other hand, the behavior of each neuron model in (artificial) neural networks is as follows.

1) Receives inputs from other neuron models and apply a weight to each of these inputs,
2) Adds the weighted values together,
3) Passes the summed value through an activation function and outputs it.

The weights (step 1) correspond to the strength of the connections with other neurons in the brain. The added values (step 2) corresponds to the electric potential in the neuron. Step 3 simulates the action of outputting the neurotransmitter (see Fig 1-3).

A neural network connects neuron models through multiple layers similarly to the human brain, and transmits the information to obtain results. Learning or training of neural networks is also similar to the human brain, as it is achieved by adjusting the weights, which model the strength of the connections between neuron models.

During the learning process, the weights are adjusted in such a direction as to minimize the error between the value obtained from the neural network and the correct value.



Model of neuron

1.5. Methods of learning in Neural Networks

1.5.1. Backpropagation

Learning of neural networks is performed by adjusting the weights, which corresponds to the strength of the connection between the neuron models, using a technique called backpropagation. In backpropagation, the error between the output values of the neural network and the desired values propagates from the output layer back to the input layer. Each weight is updated so that the error at each node (local error) obtained by backpropagation decreases. This is performed by a gradient descent algorithm, which updates the weights in the direction opposite to the gradient obtained as the partial derivative of the error function of weights (see Fig. 1-3).

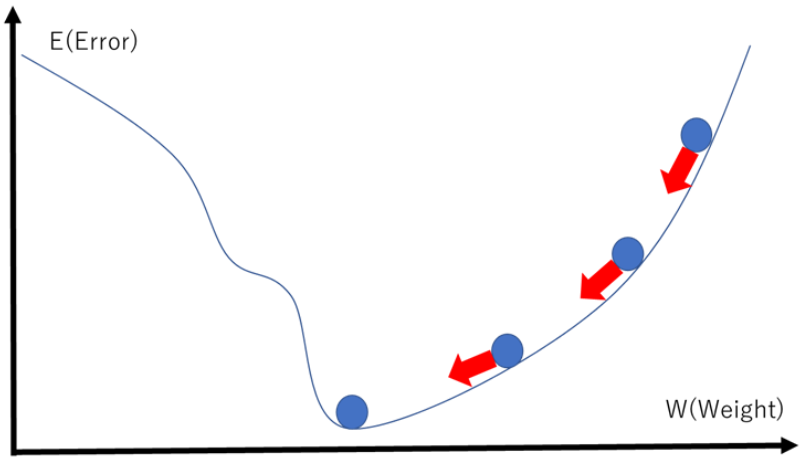


Fig. 1-3 Gradient descent algorithm

Let the n inputs of a node be

$$x_1 \sim x_n$$

and the n weights be

$$w_1 \sim w_n \ .$$

The output is expressed as

$$\mathrm{f}(\sum x_i w_i)$$

where f is a function called activation function, which controls the output of each node. The error function E is

$$\mathrm{E}\left(\mathrm{f}\left(\sum x_i w_i\right)\right) .$$

The partial derivative of the error function with respect to w_i, the weight to be updated, is expressed as

$$\frac{dE}{dw_i} = \frac{dE}{df}\frac{df}{d\sum x_i w_i}\frac{d\sum x_i w_i}{dw_i} = \frac{dE}{df}\frac{df}{d\sum x_i w_i}x_i$$

Since the activation and error functions are uniquely determined, it can be seen that the gradient depends only on the input and output values of the node. The gradient can be computed when both the activation and error functions can be differentiated.

In standard gradient descent method, a constant α $(0 < \alpha \leq 1)$ is used as a learning rate in order to update the weights:

$$w^{(t+1)} = w^{(t)} - \alpha\nabla E(w^{(t)})$$

However, since the rate of convergence of the calculation of the optimal solution is generally low when the learning rate α is a constant, the following methods are used.

- Momentum

Momentum accelerates convergence by giving momentum in the direction of updating. Let the update vector at past step $t-1$ be v_{t-1}. The current update vector at current step t is expressed as

$$v_t = \gamma v_{t-1} + \alpha\nabla E(w^{(t)})$$

where γ is a constant lower than 1. The update of weights is performed as follows.

$$w^{(t+1)} = w^{(t)} - v_t .$$

- AdaGrad

AdaGrad is an algorithm proposed by John Duchi et. al in 2011 [1], which decreases the learning rate as the number of learning steps increases. The initial learning rate α_0 and the parameter ε must be set manually.

$$h_0 = \varepsilon$$
$$h_t = h_{t-1} + \nabla E(w^{(t)})^2$$
$$\alpha_t = \frac{\alpha_0}{\sqrt{h_t}}$$
$$w^{(t+1)} = w^{(t)} - \alpha_t \nabla E(w^{(t)})$$

- RMSprop

RMSprop proposed by Tijmen Tieleman et. al in 2012 [2] is an improvement of AdaGrad, which suppresses the influence of the past gradient through a parameter β that is set manually.

$$h_t = \beta h_{t-1} + (1 - \beta)\nabla E(w^{(t)})^2$$
$$\alpha_t = \frac{\alpha_0}{\sqrt{h_t} + \varepsilon}$$
$$w^{(t+1)} = w^{(t)} - \alpha_t \nabla E(w^{(t)})$$

- AdaDelta

AdaDelta proposed by Matthew D. Zeiler in 2012 [3], is an improvement of AdaGrad or RMSprop, which removes the initial learning rate α_0 and strengthens the influence of recent gradients by introducing the parameter s_t. Parameter ρ is set manually.

$$s_0 = 0, \quad h_0 = 0$$
$$h_t = \rho h_{t-1} + (1 - \rho)\nabla E(w^{(t)})^2$$
$$v_t = \frac{\sqrt{s_t + \varepsilon}}{\sqrt{h_t + \varepsilon}} \nabla \mathrm{E}(w^{(t)})$$
$$s_{t+1} = \rho s_t + (1 - \rho) v_t^2$$

$$w^{(t+1)} = w^{(t)} - v_t$$

- Adam

Adam proposed by Diederik P. Kingma et. al in 2015 [4], is an improvement of AdaGrad, RMSProp or AdaDelta. The update of the weights is performed as follows.

$$m_0 = 0, \quad v_0 = 0$$

$$m_{t+1} = \gamma_1 m_t + (1 - \gamma_1)\nabla E(w^{(t)})$$

$$v_{t+1} = \gamma_2 v_t + (1 - \gamma_2)\nabla E(w^{(t)})^2$$

$$\hat{m} = \frac{m_{t+1}}{1 - \gamma_1^t}$$

$$\hat{v} = \frac{v_{t+1}}{1 - \gamma_2^t}$$

$$w^{(t+1)} = w^{(t)} - \alpha \frac{\hat{m}}{\sqrt{\hat{v}} + \varepsilon}$$

Currently Adam is regarded as the most effective algorithm, and is used in this study.

2. Evaluation of the characteristics of neural networks with simulation data

First, the basic characteristics of neural networks that are considered to be important for the purpose of detecting abnormalities in equipment are evaluated using simulation data.

2.1. Evaluation of the ability of extrapolation in classification problems

The most common application of Neural Networks (NN) is classification problems where they learn the relationship between input data and an attached "label". If learning was successful, NN is able to assign a label to new input data.

When performing classification by NN in order to detect abnormalities in equipment, it is necessary to at least learn what data to label "normal" and what data to label "abnormal". If learning was successful, the model assigns the label “normal” to input data similar to training data labeled "normal", and the label “abnormal” to input data similar to training data labeled "abnormal". However, it is not self-evident whether data deviating from any training data are "normal" or "abnormal".

Considering the application of detecting abnormalities in equipment, it is desirable that a model that was trained to label a certain level of defects as "abnormal" can not only label similar defects as "abnormal", but also larger defects as "abnormal" too. This characteristic is called the ability of extrapolation and we now investigate, using simple simulation data, whether NN has this characteristic in the case of classification problems.

2.1.1. Simulation conditions

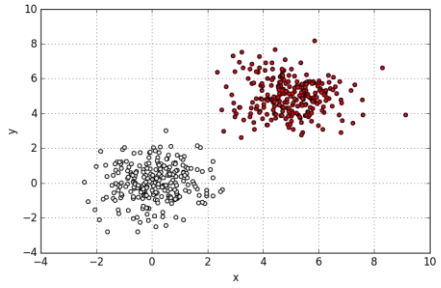
First, two-dimensional data labeled as normal or abnormal are generated, and used to train the neural network. Then the points of a two-dimensional plane are then evaluated using the trained model, and their degree of belonging to the abnormal class is represented as a contour graph.

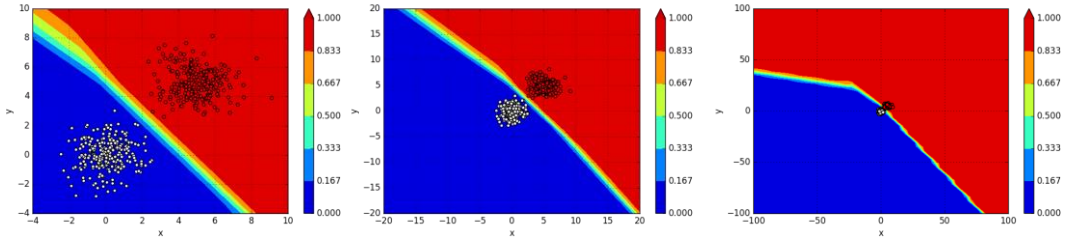
Evaluation was performed on four patterns of data. For each pattern, the respective location of the normal and abnormal data points on the two-dimensional plane were changed. In order to evaluate the influence of the layer structure of the neural network, three layer structures were used for each input data set.

For each input data set, the layer structure that was used is indicated between brackets (see Fig. 2-1 to 2-3) as the number of nodes of each layer from the input to the output layer. For example, [2-5-2] indicates that the number of nodes of the input layer is 2, the number of nodes of the hidden layer is 5, and the number of nodes of the output layer is 2.

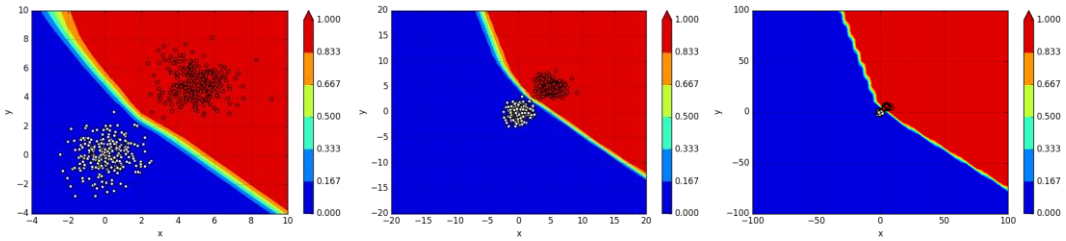
For all neural networks, the activation function of each layer is a Relu function, the dropout ratio is 0.1, and the soft max function is used for the output layer. ReLU function outputs the value as it is if the input is a positive value and 0 if it is a negative value. Dropout is a technique proposed by Hinton et al.[5] to prevent over fitting by randomly erasing a certain ratio of neurons during learning.

2.1.2. Simulation results

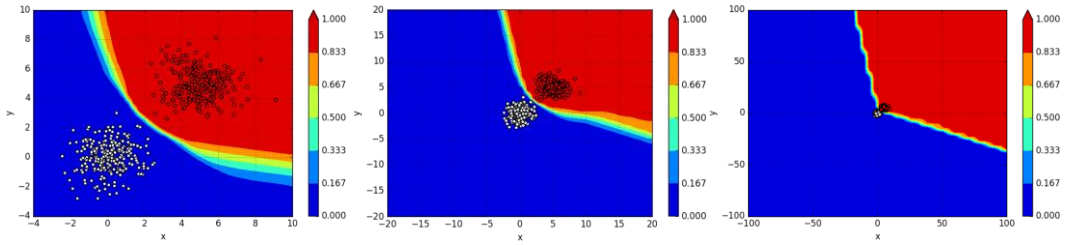




(a) Layer structure [2-5-2]



(b) Layer structure [2-10-2]

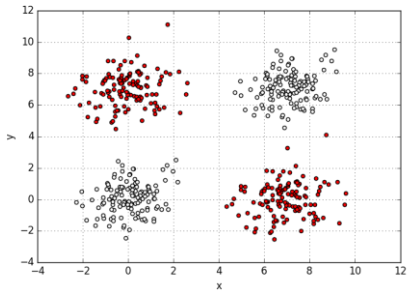


(c) Layer structure [2-10-10-2]

Fig. 2-1 Classification of two-dimensional data 1

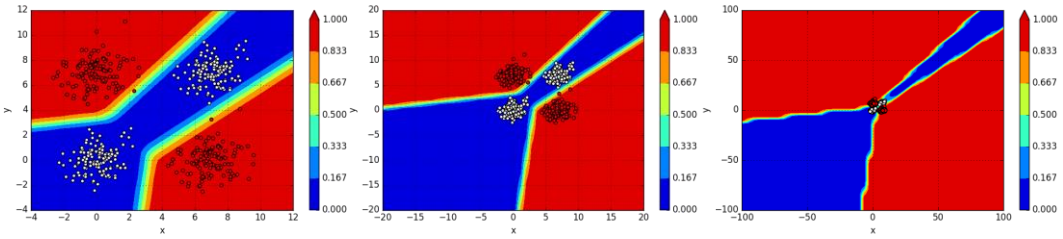
The three graphs in a single row (a, b or c in Fig. 2-1 to 2-3) are obtained by using the same model but with different ranges of evaluation. This is intended to evaluate how data points far from the learning data points are labelled by the model.

The data points in the blue regions are labelled as "normal" and the ones in the red regions as "abnormal".

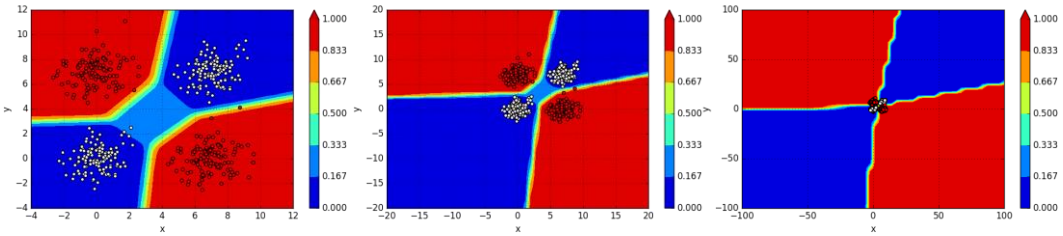


Training Data

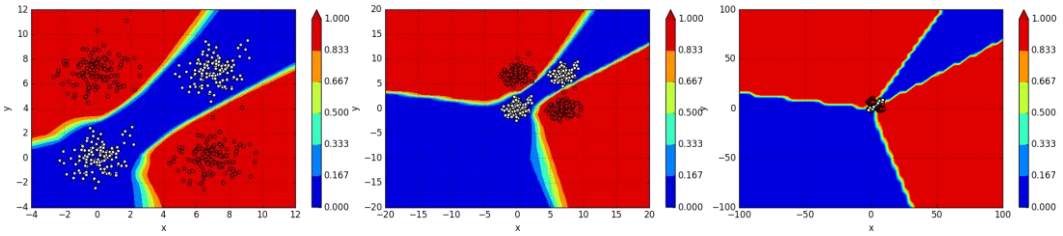
White: Normal
Red : Abnormal



(a) Layer structure [2-5-2]

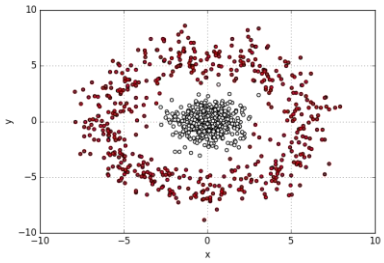


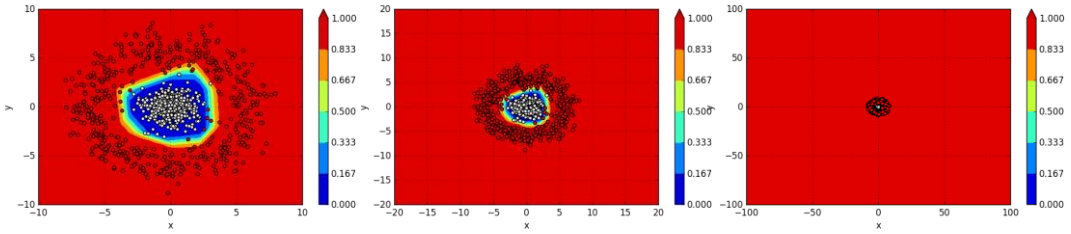
(b) Layer structure [2-10-2]



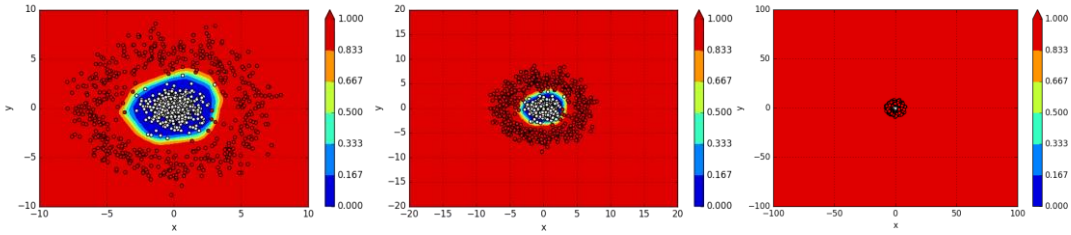
(c) Layer structure [2-10-10-2]

Fig. 2-2 Classification of two-dimensional data 2

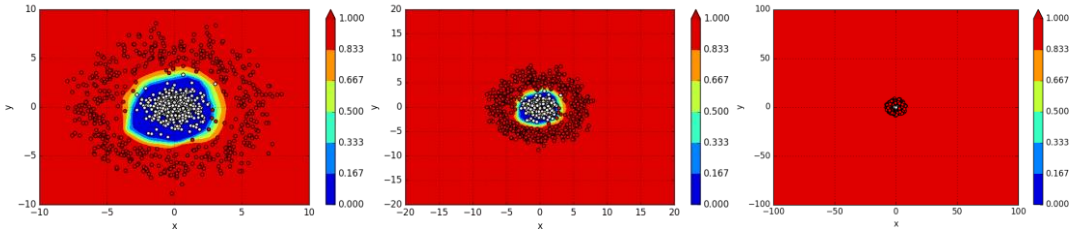




(a) Layer structure [2-5-2]

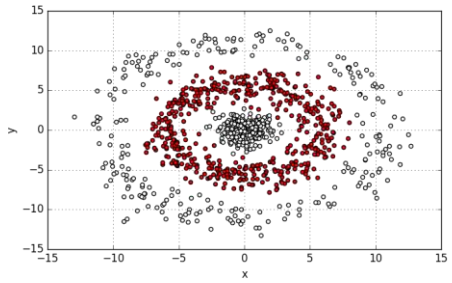


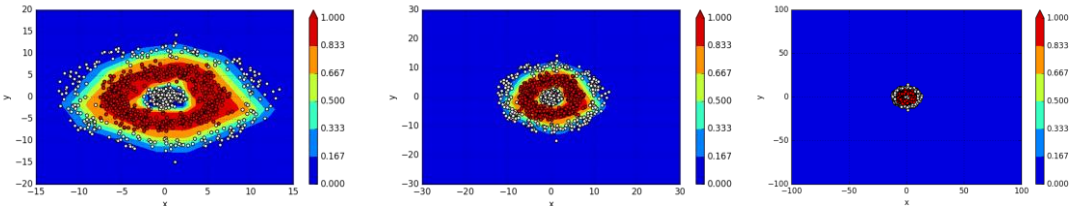
(b) Layer structure [2-10-2]



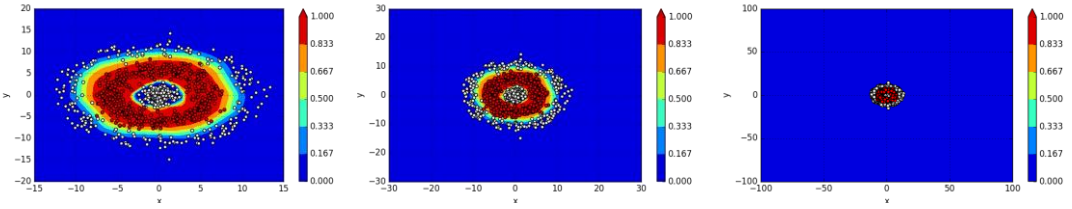
(c) Layer structure [2-10-10-2]

Fig. 2-3 Classification of two-dimensional data 3

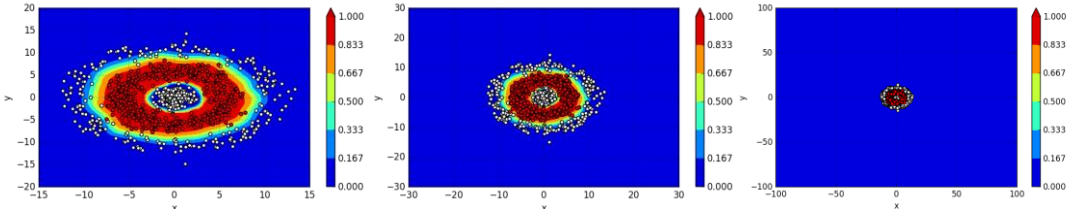




(a) Layer structure [2-5-5-2]



(b) Layer structure [2-10-10-2]



(c) Layer structure [2-30-30-2]

Fig. 2-4 Classification of two-dimensional data 4

2.1.3. Discussion

Each classification region ("normal" or "abnormal") extends from each region defined by the respective training data, without the presence of outlying regions. This is a desirable characteristic for the application of NN to the detection of abnormalities in equipment.

The larger the number of nodes in the hidden layers, the more the boundaries between normal and abnormal regions follow the training data points. This indicates that if a complicated layer structure is used, NN can learn a complicated model that better fits the training data. However, there is also the risk of overfitting. In order to prevent overfitting, it is common to divide the data into training and verification data, and to evaluate whether the model trained with the training data can also be applied to verification data. This method is used in section 3 "Evaluation by experimental data".

2.2. Evaluation of the ability of extrapolation in regression problems

2.2.1. Simulation conditions

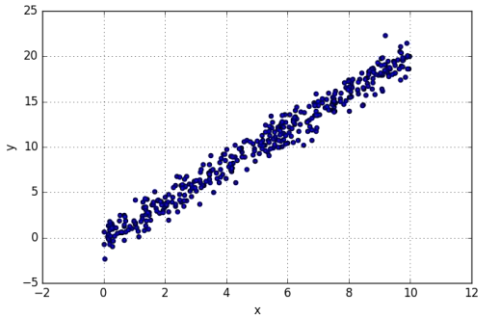
In order to clarify the fundamental characteristics of NN, the ability of extrapolation when it is applied to a regression problem is evaluated. Although in terms of learning the relation between input and output, there is no difference between the classification and the regression problems, the final output of the classification is a label, which is a value in a discrete space, while the final output of the regression is a value in a continuous space.

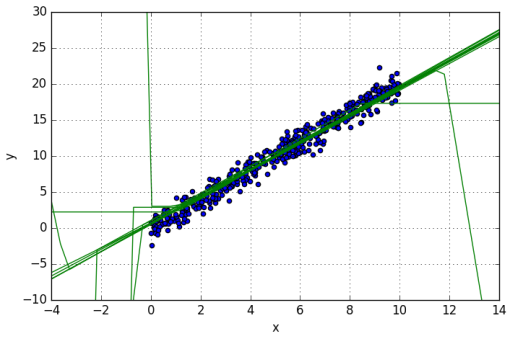
Two dimensional data are generated that consist of one explanatory variable x and one objective variable y, and used to train a neural network. Learning is performed so that the error between y and the output value $\hat{y}$ that is obtained by inputting x to the neural network is minimized. Using the obtained model, the output value $\hat{y}$ is calculated for each input x that takes its value in ranges where training data are present and where they are not.

As in the previous section, each data set is evaluated using neural networks with different layer structures. Other parameters related to a neural network such as dropout ratio and activation function, are the same as in section 2.1.1.

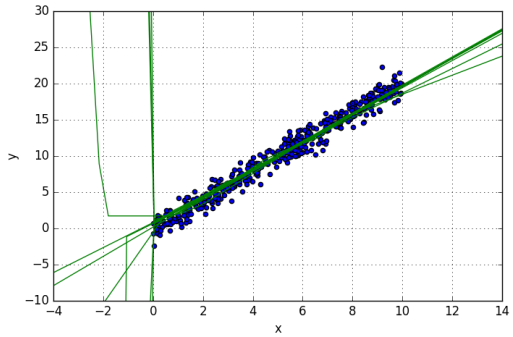
2.2.2. Simulation results

The blue dots in Fig. 2-5 to 2-7 represent the training data and the green lines are the output values when x is the input of the trained model. In order to evaluate the variation of the trained model, learning is performed 10 times and 10 models are obtained for each layer structure. All 10 results are displayed on a single graph.

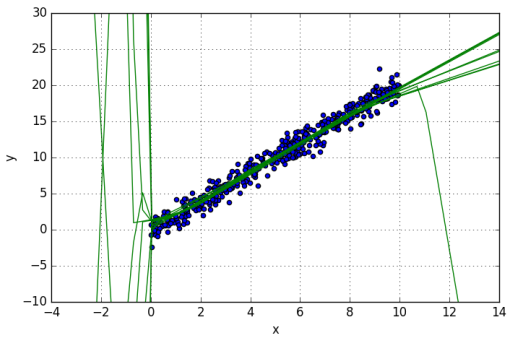




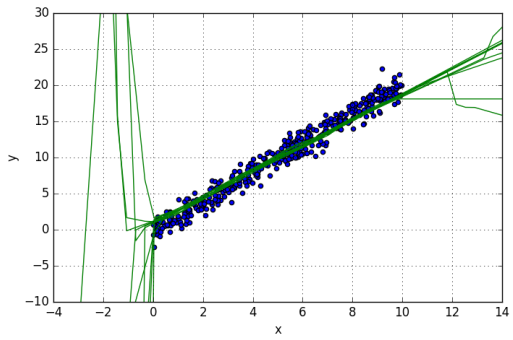
(a) Layer structure [1-3-1]



(b) Layer structure [1-6-1]

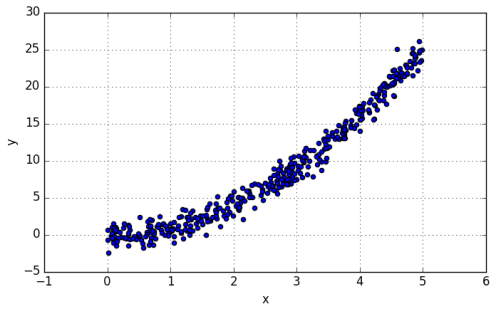


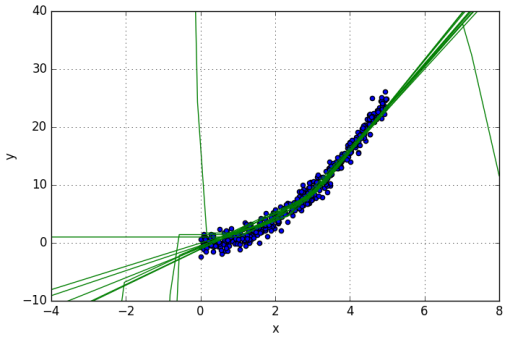
(c) Layer structure [1-10-1]



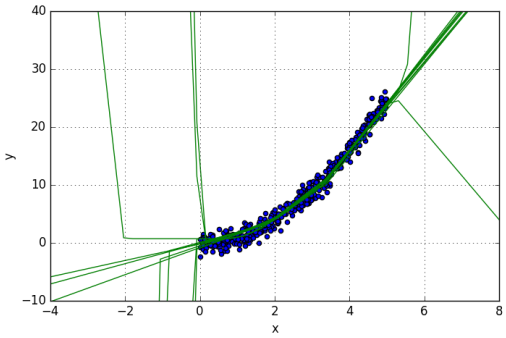
(d) Layer structure [1-10-10-1]

Fig. 2-5 Regression by Neural Network 1

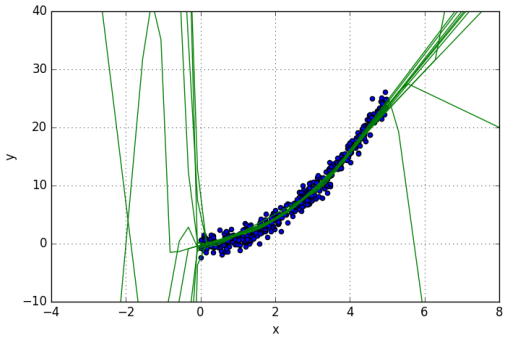




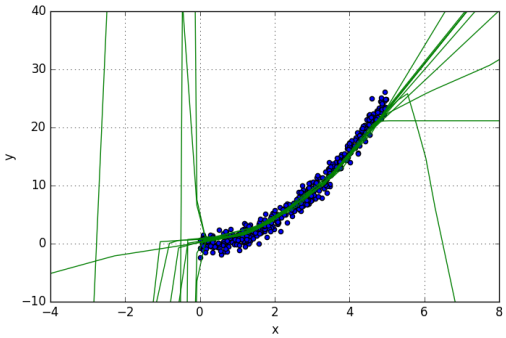
(a) Layer structure [1-3-1]



(b) Layer structure [1-6-1]

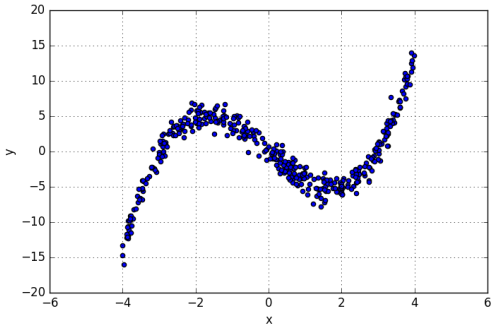


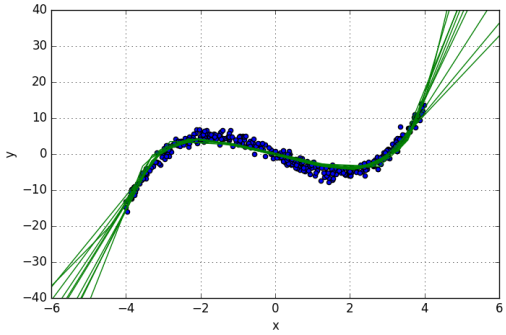
(c) Layer structure [1-10-1]



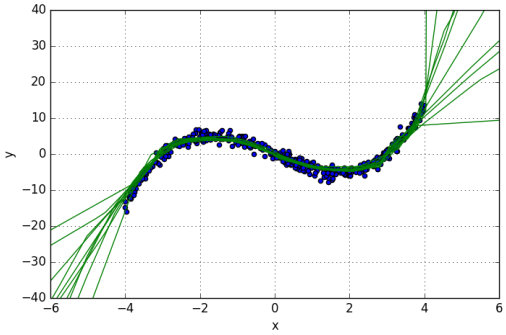
(d) Layer structure [1-10-10-1]

Fig. 2-6 Regression by Neural Network 2





(a) Layer structure [1-10-1]



(b) Layer structure [1-10-10-1]

Fig. 2-7 Regression by Neural Network 3

2.2.3. Discussion

The regression curves determined by NN fit the training data well in the range where these data exist, but it tends to be random outside this range. This is a drawback when NN is applied to regression problems.

For classification problems, the ability of extrapolation was shown to be good. In the case of regression problems, it is shown to be poor.

The reason for this is that in the case of a classification problem, only the output node with the largest value is considered, whereas in the case of a regression problem, the output value is the value of the single output node.

3. Evaluation with experimental data

In order to investigate the applicability of NN to the detection of abnormalities in equipment, acoustic data of an actual pump in operation in various conditions were analyzed through NN.

3.1. Experimental conditions

Fig. 3-1 shows a schematic illustration of the experimental setup. Pump 1 is the monitoring target. Data are acquired with the pump in normal and abnormal conditions (a damaged bearing and a misalignment of the shaft). In addition, data are acquired by changing the following conditions.

- Distance of the acoustic sensor (microphone) to Pump 1 (locations P1~P4).
- Operation or not of another pump (Pump 2) in the same room.

The operation of Pump 2 is intended for generating additional noise from another source than the monitoring target.

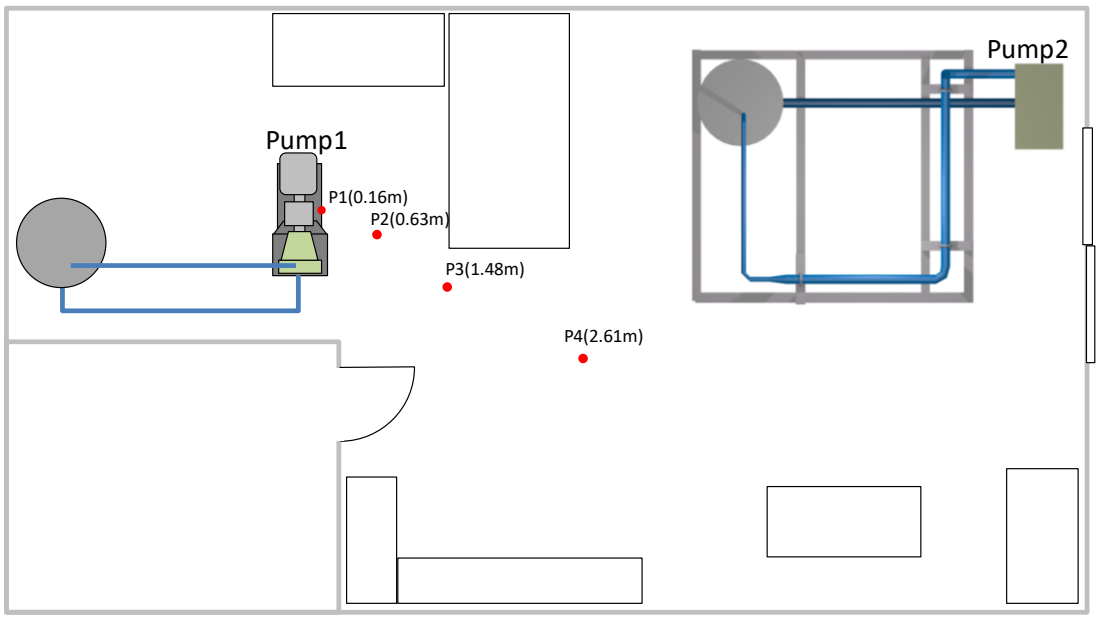


Fig. 3-1 Experimental setup

Table 3-1 Experimental conditions

Pump 1	Position of Sensor	Pump 2 (Noise source)
Normal 1	P1 (0.16m)	Stop
		Run
	P2 (0.63m)	Stop
		Run
	P3 (1.48m)	Stop
		Run
	P4 (2.61m)	Stop
		Run

Pump 1	Position of Sensor	Pump 2 (Noise source)
Bearing Damage 1 (severe)	P1 (0.16m)	Stop
		Run
	P2 (0.63m)	Stop
		Run
	P3 (1.48m)	Stop
		Run
	P4 (2.61m)	Stop
		Run

Pump 1	Position of Sensor	Pump 2 (Noise source)
Misalign ment 1 (severe)	P1 (0.16m)	Stop
		Run
	P2 (0.63m)	Stop
		Run
	P3 (1.48m)	Stop
		Run
	P4 (2.61m)	Stop
		Run

Pump 1	Position of Sensor	Pump 2 (Noise source)
Normal 2	P1 (0.16m)	Stop
		Run
	P2 (0.63m)	Stop
		Run
	P3 (1.48m)	Stop
		Run
	P4 (2.61m)	Stop
		Run

Pump 1	Position of Sensor	Pump 2 (Noise source)
Bearing Damage 2	P1 (0.16m)	Stop
		Run
	P2 (0.63m)	Stop
		Run
	P3 (1.48m)	Stop
		Run
	P4 (2.61m)	Stop
		Run

Pump 1	Position of Sensor	Pump 2 (Noise source)
Misalignm ent 2	P1 (0.16m)	Stop
		Run
	P2 (0.63m)	Stop
		Run
	P3 (1.48m)	Stop
		Run
	P4 (2.61m)	Stop
		Run

Although both "Normal 1" and "Normal 2" are normal conditions, the pump is disassembled and assembled between the measurements in these two conditions. For the condition of a damaged bearing, the inner ring of the bearing was artificially damaged with a drill. A bigger defect was introduced in the case of "Bearing Damage 1" than in the case of "Bearing Damage 2". Misalignment is introduced by inserting a thin plate between the motor and the foundation. A larger misalignment is introduced in the case of "Misalignment 1" than in the case of "Misalignment 2".

The appearance and specifications of Pump 1 are shown in Fig. 3 2 and Table 3 2.

Fig. 3-2 Pump 1

Table 3-2 Specifications of Pump 1

Power	1.5kW
Rotation Speed	2880rpm
Capacity	75L/min
Total Head	23.2m

3.2. Application of Neural Networks

When applying NN to the detection of abnormalities in equipment, mainly two methods can be considered.

(1) Learning is done from labeled data such as "normal" or "abnormal", then new data are classified by assigning a label from the trained labels.

(2) Learning is performed only from normal data, then new data are evaluated depending on the output of the trained model.

The first method is often chosen and has achieved great results especially in the field of image recognition. However, in the case of equipment monitoring, labeling the data is not necessarily a good feature. Therefore, both methods are evaluated, and the results are given in sections 3.3 and 3.4 respectively.

3.3. Classification by Neural Networks

Part of the experimental data described in section 3.1 are used as training data with assigned labels such as "normal", "bearing damage" and "misalignment", and the remaining data are classified by the trained model. Then the accuracy of the classification is evaluated. The two following kinds of neural networks are applied.

- Fully Connected Neural Network (FCNN)
- Convolutional Neural Network (CNN)

For each label, or state, "Normal", "Bearing damage" and "Misalignment", the ideal outputs are defined as shown in Fig. 3-3. Learning is performed so that the difference between the output values and the ideal output values of the corresponding state is minimized. For the classification, the result is the state that corresponds to the node with the highest value. In addition, the accuracy of the result is evaluated. In the case of Fig. 3-3 for example, the classification result is "Misalignment".

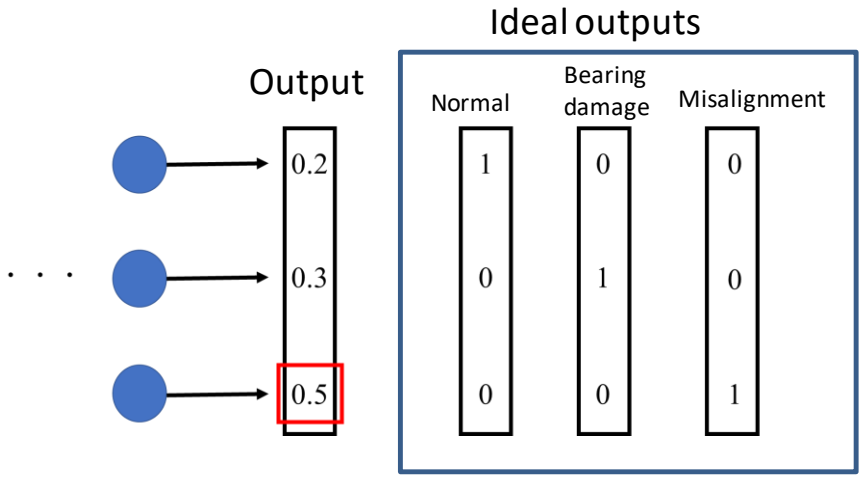


Fig. 3-3 Ideal outputs of each state

3.3.1. Fully Connected Neural Network

The most common type of neural networks is the "Fully Connected Neural Network" where each node of a given layer is connected to all nodes of the next layer.

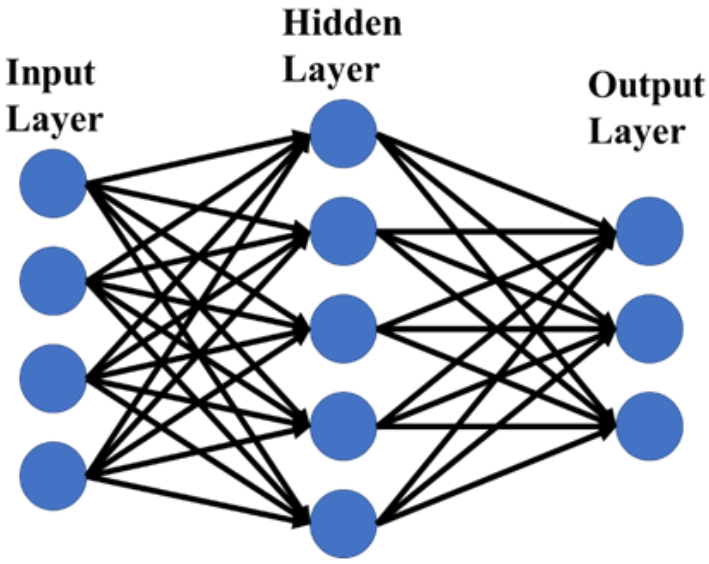


Fig. 3-4 Fully Connected Neural Network

3.3.1.1. Input data

Data are acquired for 10 seconds (200000 samples) for all conditions. The training data are constituted of 10000 contiguous samples that are randomly selected (see Fig. 3-5) from the acquired data.

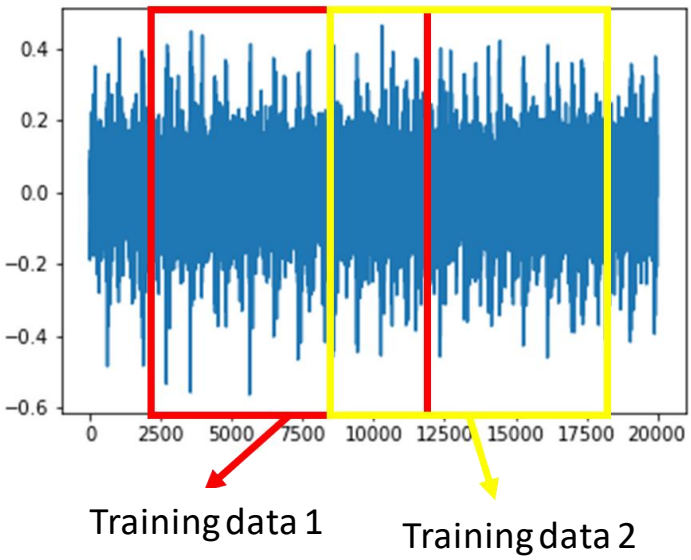


Fig. 3-5 Random sampling of input data

When acoustic data are used, it is common to apply a Fourier transform to the data in order to eliminate the variation caused by phase differences. In this analysis, in order to eliminate the difference in level caused by the varying distance between the pump and the microphone, data are first RMS-normalized, then the Fourier transform is applied. In order to reduce the dimension of the input data, the spectrum is then divided into 100 parts and the RMS of each part is calculated to build an input vector of 100 elements (see Fig. 3-7).

In addition, in order to evaluate the effect of the frequency range of the input data, three frequency ranges, 0-500Hz, 0-1000Hz and 0-5000Hz (see Fig. 3-6), are used for the learning phase and the accuracy of classification is evaluated for each range.

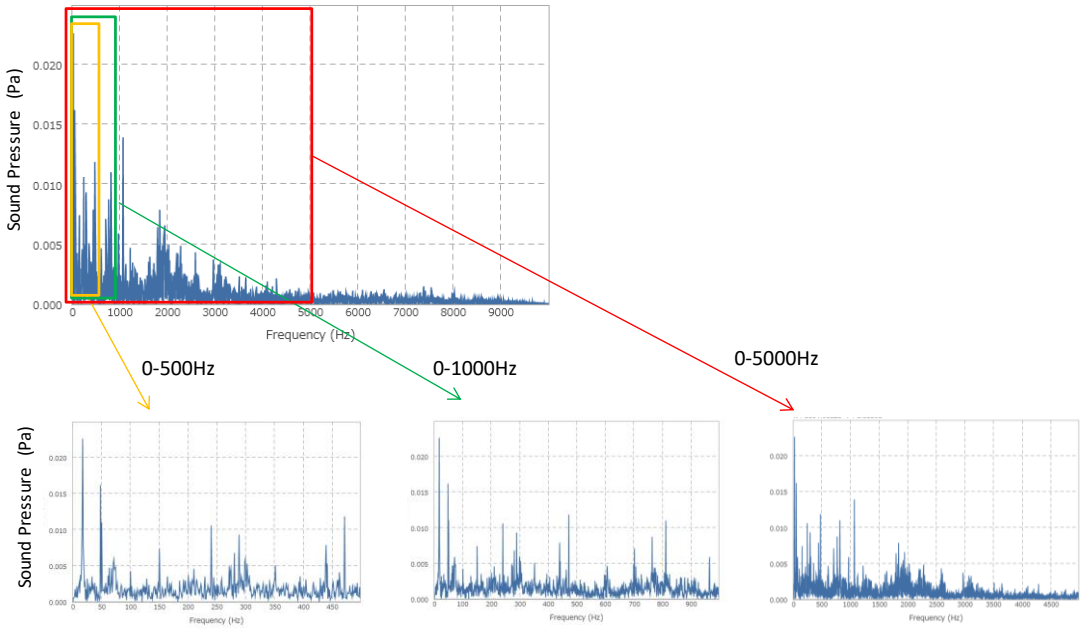


Fig. 3-6 Frequency spectrum (0-500Hz、0-1000Hz、0-5000Hz)

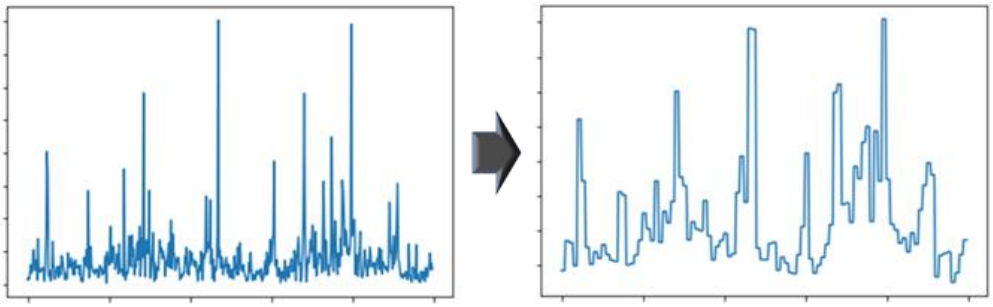
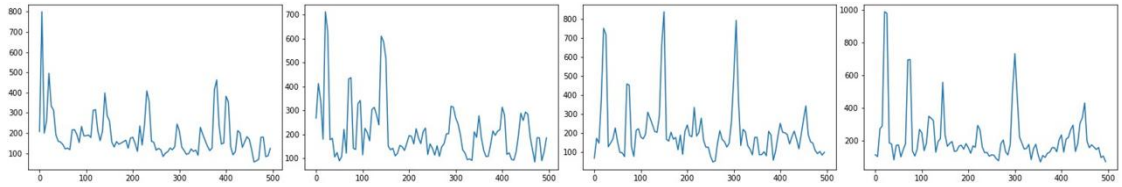


Fig. 3-7 Division of the frequency spectrum in 100 parts

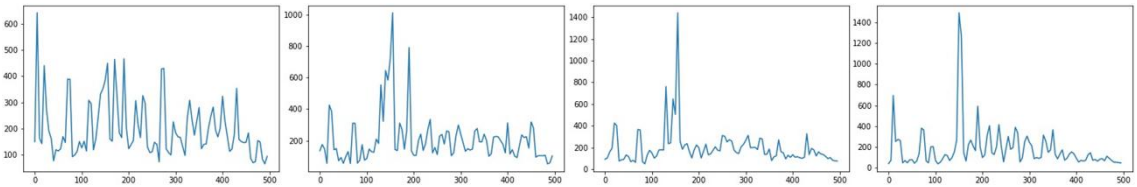
Fig. 3-8 shows examples of the input vector in "Normal", "Bearing Damage", and "Misalignment" conditions and for each frequency range.

Position of sensor

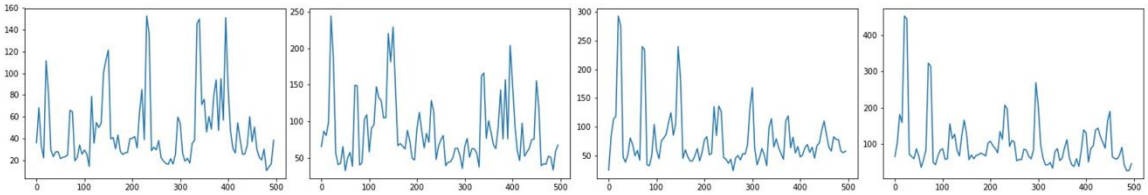
P1 P2 P3 P4



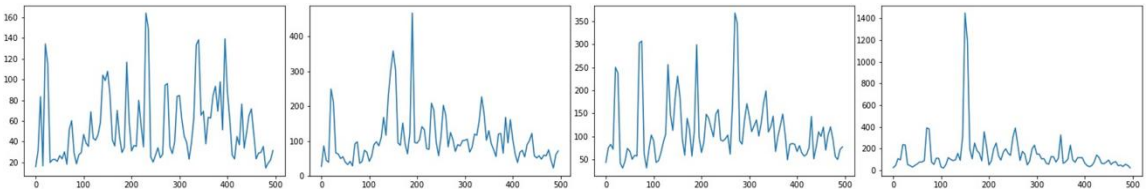
"Normal" 0-500Hz, Pump2 Stop



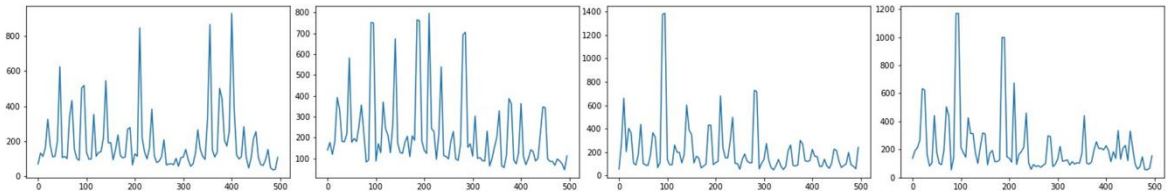
"Normal" 0-500Hz, Pump2 Run



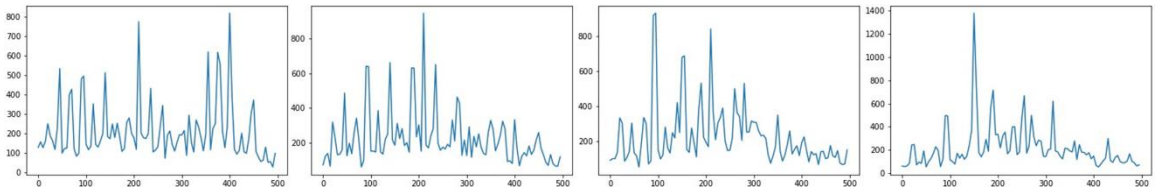
"Bearing Damage" 0-500Hz, Pump2 Stop



"Bearing Damage" 0-500Hz, Pump2 Run



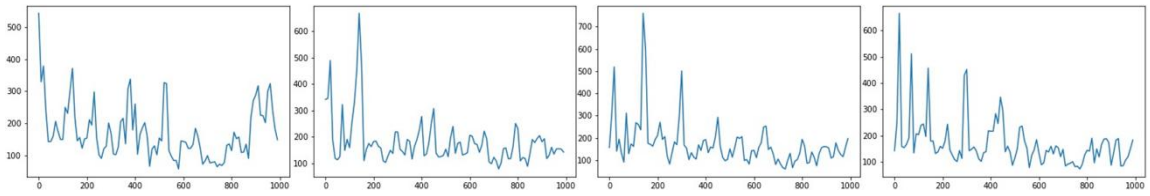
"Misalignment" 0-500Hz, Pump2 Stop



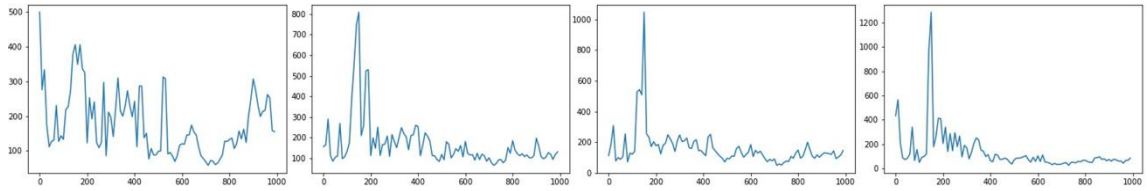
"Misalignment" 0-500Hz, Pump2 Run

Position of sensor

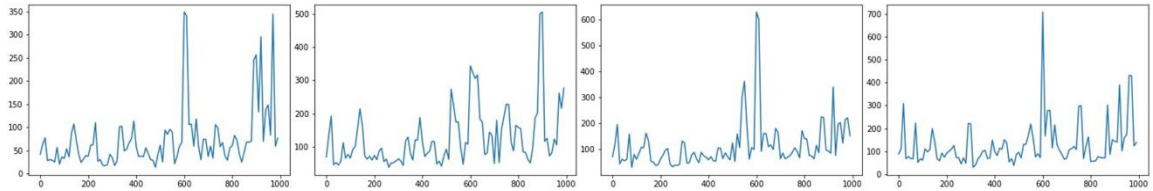
P1 P2 P3 P4



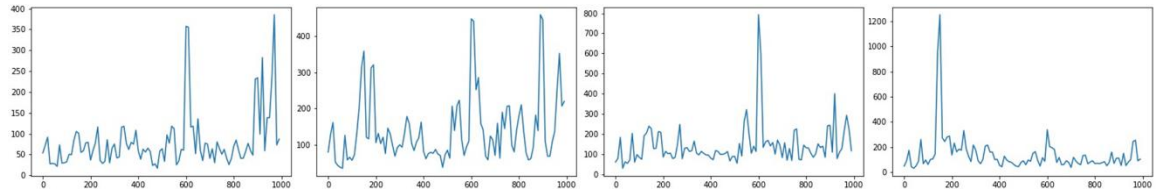
"Normal" 0-1000Hz, Pump2 Stop



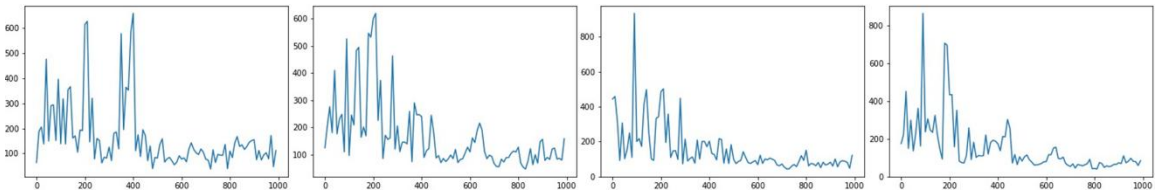
"Normal" 0-1000Hz, Pump2 Run



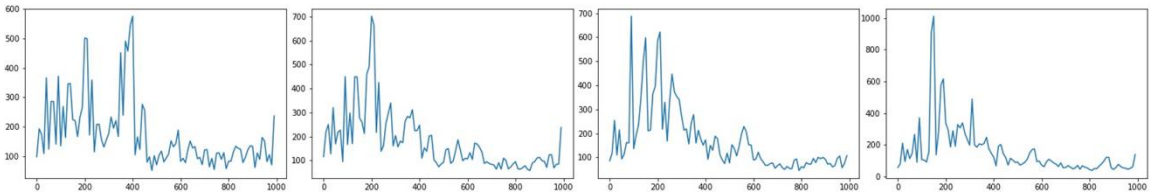
"Bearing Damage" 0-1000Hz, Pump2 Stop



"Bearing Damage" 0-1000Hz, Pump2 Run



"Misalignment" 0-1000Hz, Pump2 Stop



"Misalignment" 0-1000Hz, Pump2 Run

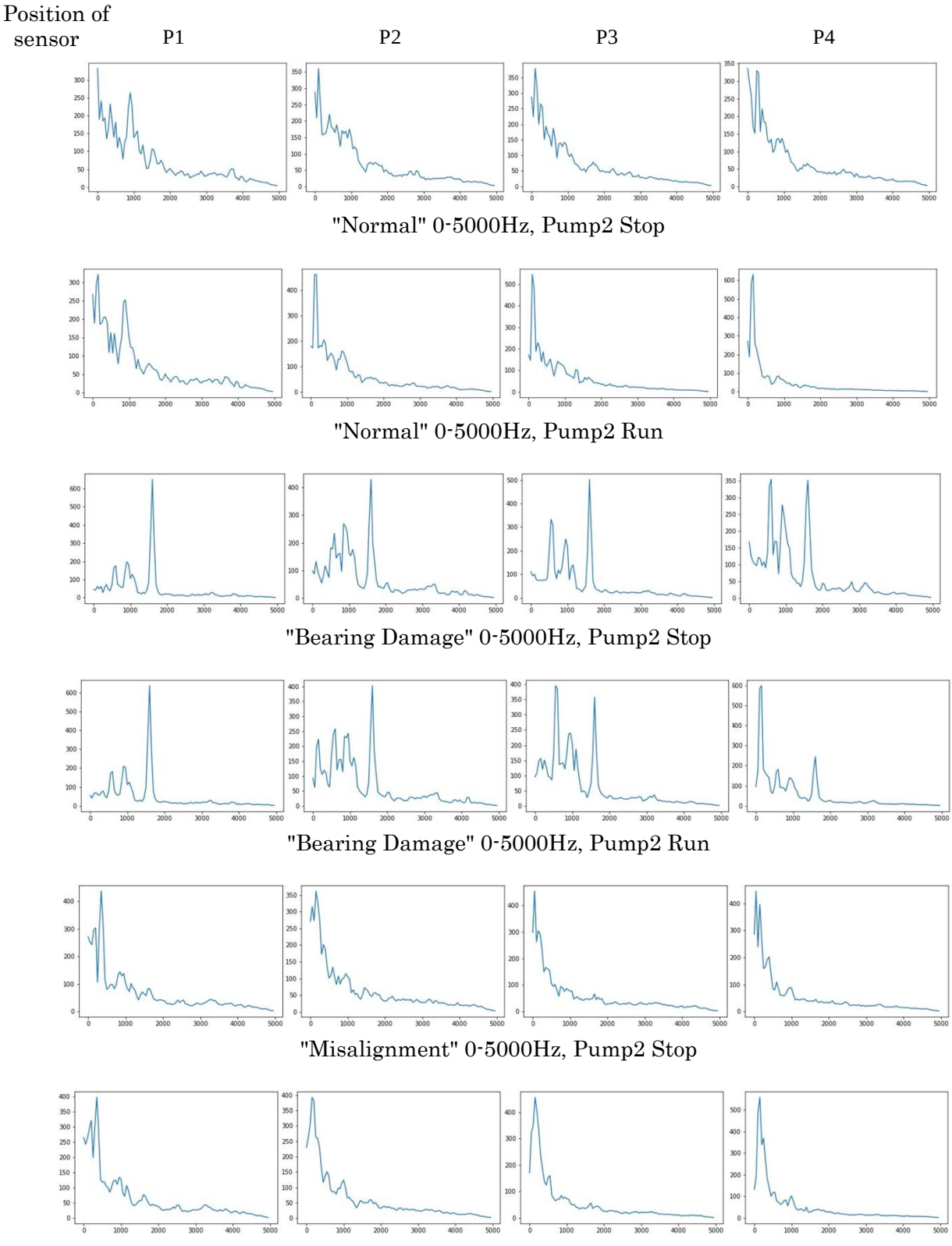


Fig. 3-8 Examples of input data

3.3.1.2. Calculation conditions

The neural network used for learning consists of an input layer with 100 nodes, two hidden layers with 32 and 8 nodes respectively, and an output layer with 3 nodes that corresponds to the 3 states "normal", "bearing damage" and "misalignment".

Multiple measurements under the same conditions are divided into training data and evaluation data, and the training data are used for the learning of the model. The accuracy of the classification is then evaluated by inputting the evaluation data into the trained model. Evaluation data are used to determine if overfitting occurs, which would impair generalization performance. It is expected that the accuracy of classification at the time of the implementation of the actual system will be close to the accuracy of the classification of the evaluation data.

In order to investigate whether or not the results vary depending on the ratio between the amount of training and evaluation data, two ratios are evaluated, 9:1 and 6:4. The number of input vectors for the learning phase is 432 with a ratio of 9:1 and 288 with a ratio of 6:4. The dropout ratio is 0.05 in both cases.

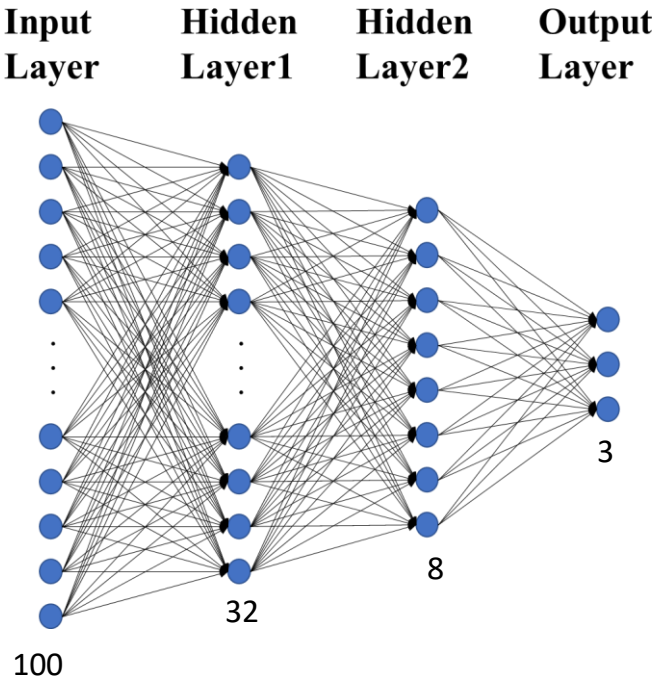


Fig. 3-9 Network structure

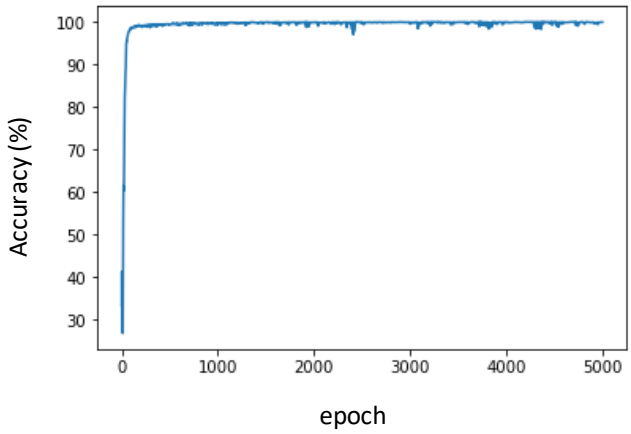
3.3.1.3. Calculation results

(1) Accuracy of classification

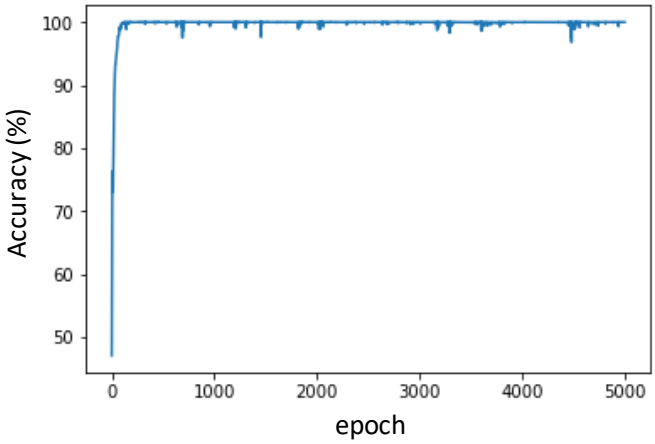
In Fig. 3-10 and 3-11, the horizontal axis represents the number of iterations of the learning phase and the vertical axis represents the accuracy of classification of the evaluation data. The accuracy of classification increases as the learning proceeds, and eventually reaches almost 100% in all cases. The accuracy at the end of the learning is summarized in Table 3-5.

The accuracy of classification is not affected by the ratio between the amount of training and evaluation data. This implies that the amount of training data was sufficient in both cases.

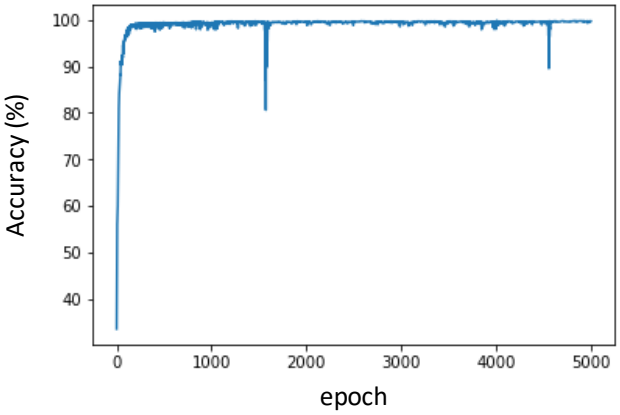
While the difference is very small, of the three frequency ranges that were evaluated, the frequency range 0-1000Hz gives the highest accuracy and the range 0-5000Hz the lowest.



(a) Input Data : 0-500Hz



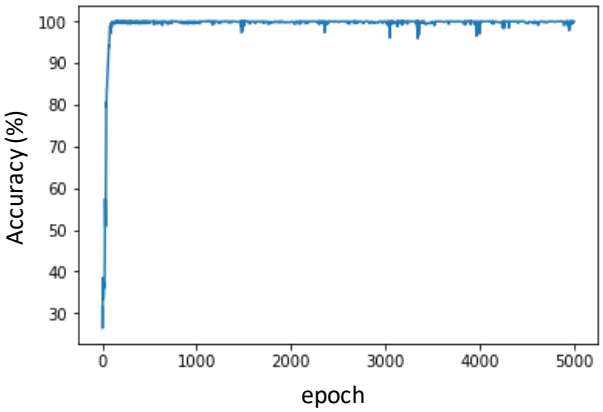
(b) Input Data : 0-1000Hz



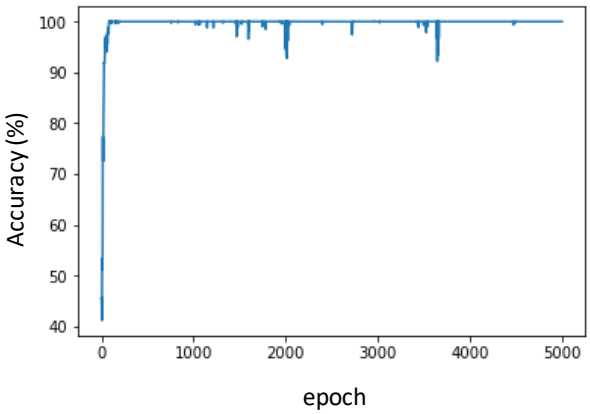
(c) Input Data : 0-5000Hz

Fig. 3-10 Classification accuracy depending on learning iterations

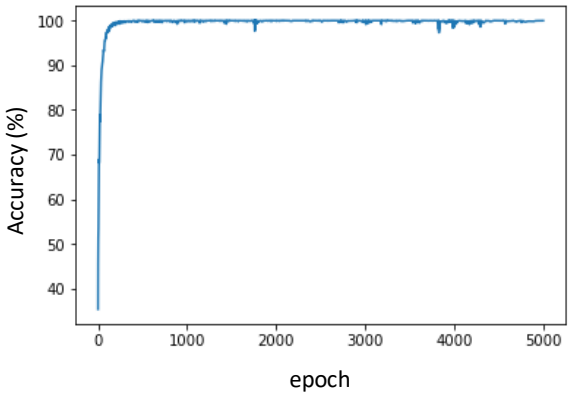
Ratio of training to evaluation data 6:4



(a) Input Data : 0-500Hz



(b) Input Data : 0-1000Hz



(c) Input Data : 0-5000Hz

Fig. 3-11 Classification accuracy depending on learning iterations
Ratio of training to evaluation data 9:1

Table 3-3 Classification Accuracy at the end of the learning

Ratio of training to evaluation data	Frequency Range		
	0-500Hz	0-1000Hz	0-5000Hz
6:4	99.9%	100%	99.8%
9:1	100%	100%	99.8%

3.3.1.4. Evaluation of the ability of extrapolation

In section 2.1, the ability of extrapolation was shown to be good for the classification problem with simulation data. In order to evaluate this ability with experimental data, learning is performed only with normal data and lightly abnormal data "Bearing Damage 2" and "Misalignment 2", excluding abnormal data "Bearing Damage 1" and "Misalignment 1" that are more severe. Using this trained model, how more severe abnormal data are classified is evaluated.

In this evaluation, the frequency range of the input data is set to 0-1000Hz, and the ratio of training to evaluation data is 9:1 (except for "Bearing Damage 1" and "Misalignment 1" data that are all evaluation data). Fig. 3-12 shows the accuracy of classification depending on the learning iterations. The left figure is the accuracy when the evaluation data consists of normal data and abnormal data "Bearing Damage 2" and "Misalignment 2". The right figure is the one when the evaluation data consists of normal data and abnormal data "Bearing Damage 1" and "Misalignment 1".

In the case of normal and lightly abnormal data, since the states of the evaluation data are included in the training data, the accuracy reaches 100%. On the contrary, in the case of more severe abnormalities, since these abnormal data are not included in the training data, the accuracy is lower. However it reaches 95%, thus exhibiting a good ability of extrapolation.

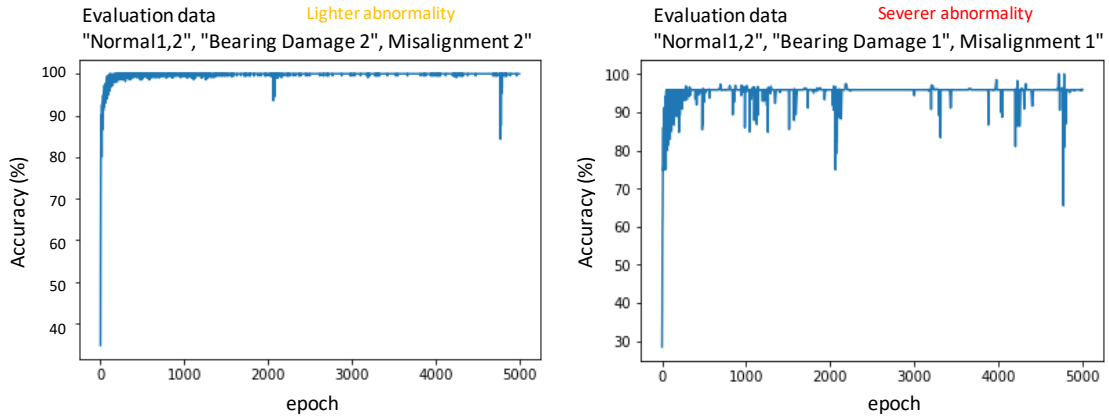


Fig. 3-12 Accuracy of classification by the model trained with lightly abnormal data (dropout 0.05)

The results when the dropout ratio of the learning phase is increased from 0.05 to 0.15 and the number of learning iterations is increased from 5000 to 20000, are shown in Fig. 3-13.

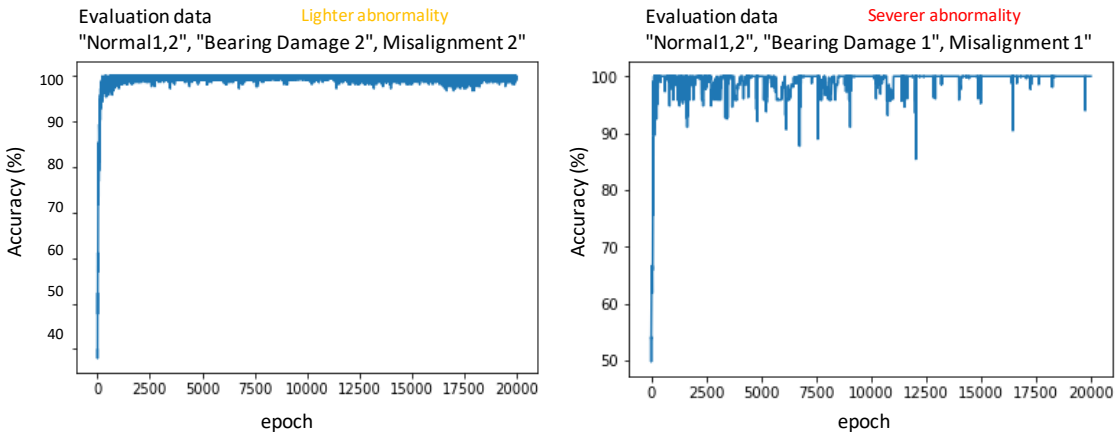


Fig. 3-13 Accuracy of classification by the model trained with lightly abnormal data (dropout 0.15)

The accuracy in the case of more severe abnormalities reaches almost 100% by increasing the dropout ratio and learning iterations. This implies that the ability of extrapolation can be increased by adjusting the dropout ratio.

3.3.1.5. Visualization of trained models

NN automatically perform extraction of features and classification, therefore it is not necessary for users to know what kind of features are extracted or how the model classifies the data. However, this knowledge can be useful in some applications.

For example, the frequency spectrum was used as input data, and to know the frequency bands the trained model is focusing on for classification can be useful to establish a method for monitoring equipment.

One of the visualization methods considered is to calculate the optimal input data. That is the input data that lead to output values that are the closest to the ideal output values for each state (see Fig. 3-3).

In NN, the update of weights is done using the gradient descent method. This is an algorithm that updates the weights in the opposite direction of the gradient, which is obtained by differentiating the error function with respect to the weights. The error function is the error between the output and the ideal output values for each state. When determining the optimal input data from the trained model, the gradient is obtained by differentiating the error function with respect to the input instead.

The optimal input data are shown in Fig. 3-14. The input data for the three frequency ranges are evaluated.

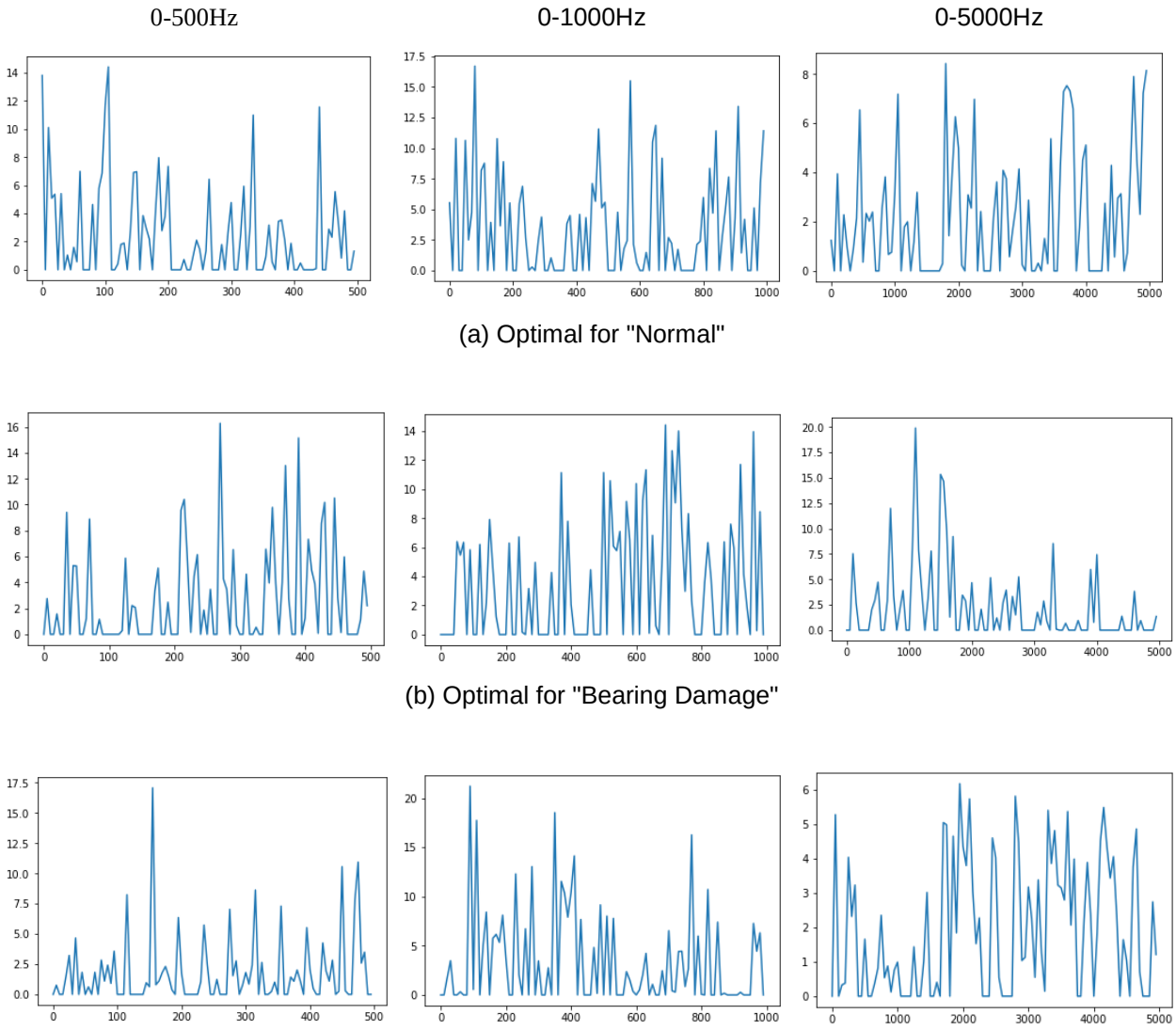


Fig. 3-14 Optimal input data

These optimal input data are different from the actual input data and do not represent sounds produced by the pump. This is due to the fact that input data acquired in different conditions such as different microphone positions, operation or not of pump 2, are still classified as the same state. However, the trained model classifies these optimal sounds as the corresponding state with the strongest certainty.

For the frequency range 0-500Hz (see Fig. 3-14), the magnitude of the spectrum is relatively high in the low frequency region in "Normal" conditions and in the high frequency region in "Bearing Damage" conditions. Moreover, periodical peaks appears in the spectrum in "Misalignment" conditions, the interval of which coincides with the rotation frequency of the pump. This is consistent with the common knowledge about

the vibration spectrum when misalignment occurs.

3.3.2. Convolutional Neural Network (CNN)

CNN is a neural network that includes not only fully connected layers but also convolutional layers and pooling layers. A convolutional layer has inputs from only some of the nodes of the previous layer. This is inspired by the animal visual cortex and is especially suitable for classifying images. Pooling layers, that combines the outputs of several nodes of a layer into a single node in the next layer, are used for reducing the dimension of data.

Forward propagation and back propagation in convolutional layers in CNN are calculated as follows.

- Forward propagation

For an M×N dimension input image and an m×n dimension kernel, the output image is expressed as

$$a_{ij}^{(k)} = \sum_{s=0}^{m-1} \sum_{t=0}^{n-1} w_{s\,t}^{(k)}\, x_{(i+s)(j+t)} + b^{(k)}$$

x : Input image
a : Output image
w : Kernel
k : Index of kernel
b : Bias

- Back propagation

The parameters of the model to be optimized are w and b. Let E be the error function. Each gradient is expressed as follows.

$$\frac{\partial E}{\partial w_{s\,t}^{(k)}} = \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\partial E}{\partial a_{ij}^{(k)}} \frac{\partial a_{ij}^{(k)}}{\partial w_{st}^{(k)}} = \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\partial E}{\partial a_{ij}^{(k)}}\, x_{(i+s)(j+t)}$$

$$\frac{\partial E}{\partial b^{(k)}} = \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\partial E}{\partial a_{ij}^{(k)}} \frac{\partial a_{ij}^{(k)}}{\partial b^{(k)}} = \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\partial E}{\partial a_{ij}^{(k)}}$$

When multiple convolutional layers exist, back propagation from the convolutional layers is required, and the error of a convolutional layer is expressed as

$$\frac{\partial E}{\partial x_{ij}} = \sum_{s=0}^{m-1}\sum_{t=0}^{n-1} \frac{\partial E}{\partial a^{(k)}_{(i-s)(j-t)}} \frac{\partial a^{(k)}_{(i-s)(j-t)}}{\partial x_{ij}} = \sum_{s=0}^{m-1}\sum_{t=0}^{n-1} \frac{\partial E}{\partial a^{(k)}_{(i-s)(j-t)}} w^{(k)}_{st}$$

In the above equation , $\frac{\partial E}{\partial a^{(k)}_{(i-s)(j-t)}} = 0$ when $i - s < 0$ or $j - t < 0$.

3.3.2.1. Input Data

In order to apply CNN to acoustic data, 10000 contiguous samples are selected randomly and reshaped as a 100 × 100 matrix that constitutes the input data (see Fig. 3-9). In order to eliminate the influence of the varying level of the sound due to the distance between the microphone and the pump, data are first RMS-normalized (see section 3.3.1).

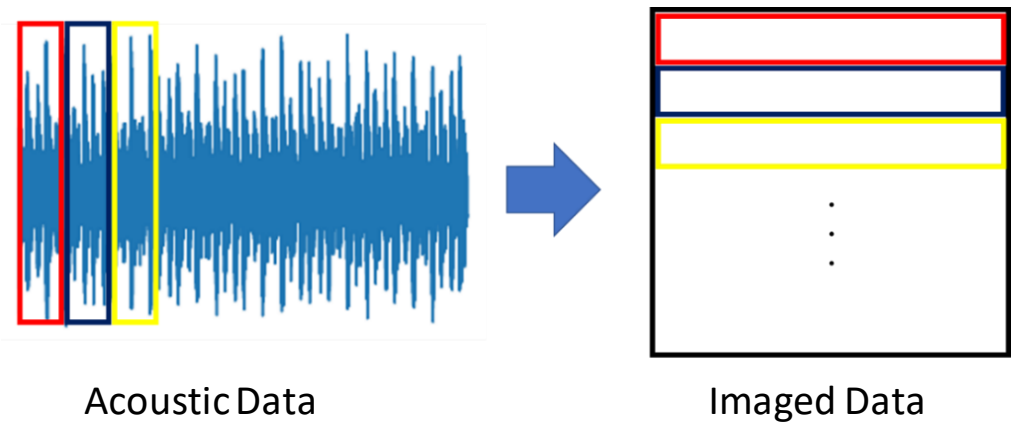


Fig. 3-15 Creation of input matrix

Examples of the input matrix for each condition of the pump are shown in Fig. 3-16. As it can be seen, each matrix has a unique pattern. For this reason, it is expected that classification by CNN is possible.

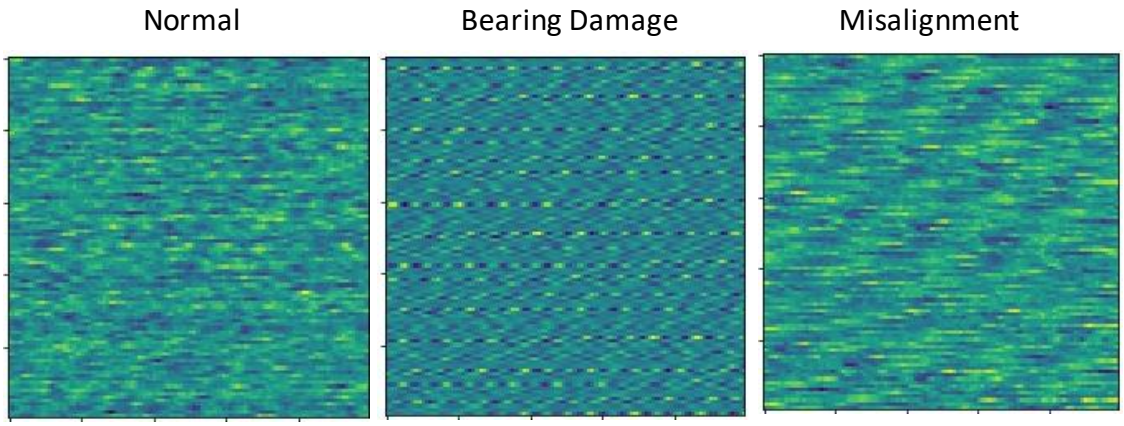


Fig. 3-16 Input matrix examples

3.3.2.2. Calculation conditions

The CNN model used for this evaluation consists of an input layer with 10000 nodes, 4 convolutional layers, 3 fully connected layers and an output layer with 3 nodes. For convolutional layers, the size of the convolution kernel is 10, 8, 5, 2 respectively, and the stride is set to 2 for a kernel size of 10 and 1 for the other kernel sizes. The stride is the value of shifting when applying the convolutional filter. The number of kernels is 50, 100, 200, and 500 respectively. The output of the convolutional layers is 500 feature maps of dimension 1×1 (see Fig. 3-17). The obtained feature maps are then the inputs of the fully connected layers. The number of nodes of the fully connected layers is 500, 128, and 32 respectively (see Fig. 3-19). Two different ratios, 9:1 and 6:4, between the amount of training and evaluation data are evaluated (see section 3.3.1).

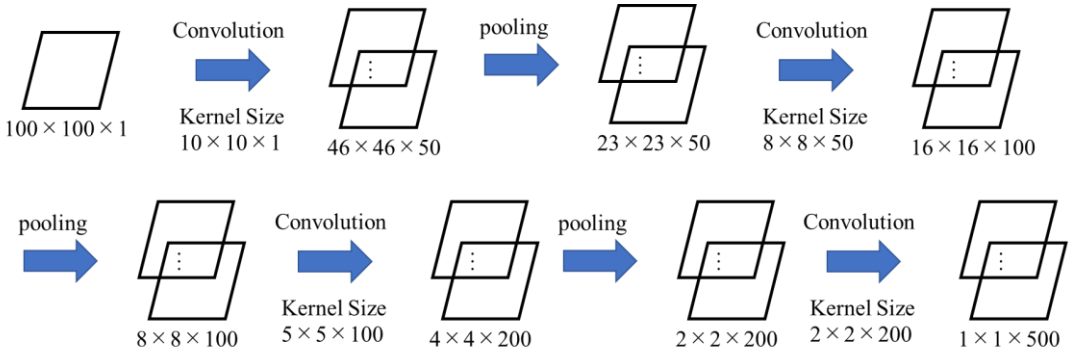


Fig. 3-17 Convolutional layers

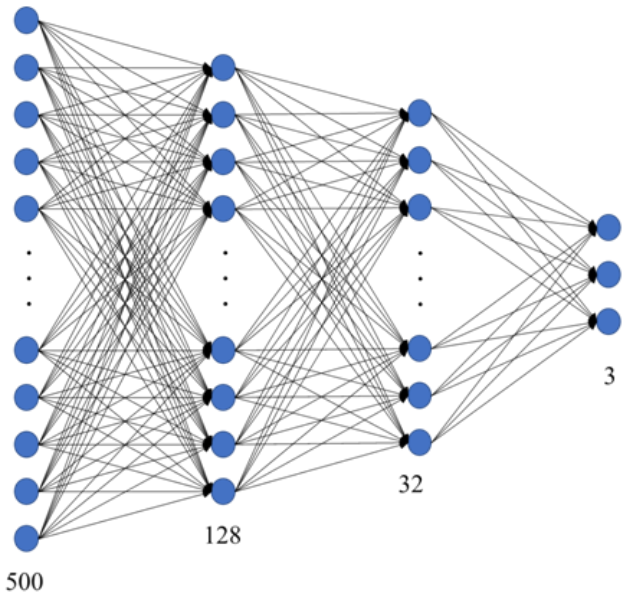


Fig. 3-18 Fully connected layers

3.3.2.3. Calculation results

Fig. 3-19 and Fig. 3-20 show the accuracy of classification by CNN depending on the learning iterations. The accuracy at the end of learning reaches 99.8% and 100% depending on the ratio, 6:4 and 9:1 respectively.

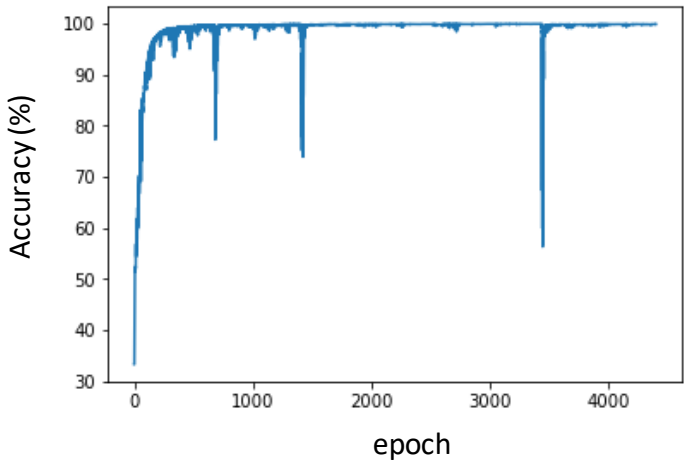


Fig. 3-19 Accuracy of classification by CNN
Ratio of training to evaluation data 6:4

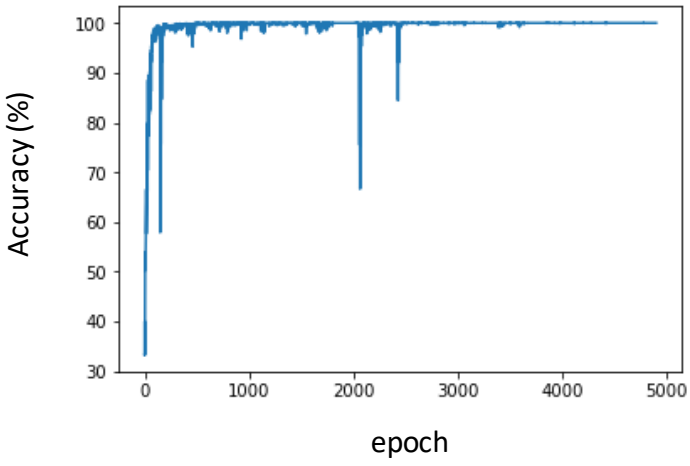


Fig. 3-20 Accuracy of classification by CNN
Ratio of training to evaluation data 9:1

Table 3-4 Classification Accuracy at the end of the learning by CNN

Ratio of training to evaluation data	Accuracy of classification
6:4	99.8%
9:1	100.0%

3.3.2.4. Evaluation of the ability of extrapolation

The ability of extrapolation of classification by CNN is evaluated. Learning is performed only with normal data and abnormal data "Bearing Damage 2" and "Misalignment 2", excluding the more severe abnormal data "Bearing Damage 1" and "Misalignment 1". Using the trained model, how the more severe abnormal data are classified is evaluated.

Fig. 3-12 shows the accuracy of classification depending of the learning iterations.

In the case of normal data and abnormal data "Bearing Damage 2" and "Misalignment 2", since the states of the evaluation data are included in the training data, the accuracy reaches 100%. On the contrary, in the case of more severe abnormal data "Bearing Damage 1" and "Misalignment 1", since these data are not included in the training data, the accuracy is somewhat lower. However, it still reaches 99.2% and exhibits a better ability of extrapolation than in the case of FCNN.

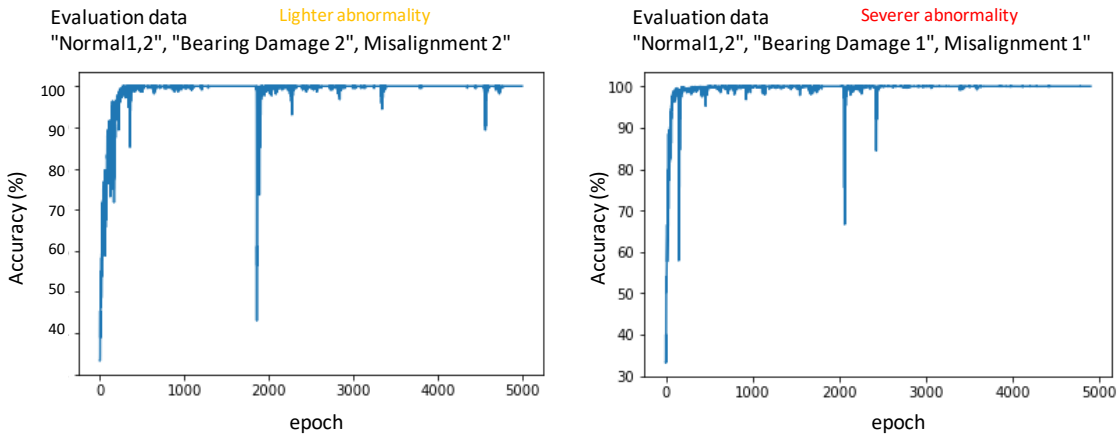


Fig. 3-21 Accuracy of classification by the CNN model trained with lightly abnormal data

Fig. 3-23 shows the results when the learning iterations are increased 4 times (20000 iterations). In this case, the accuracy at the end of the learning reaches 100% in both

cases. These results show that if light abnormalities are learned as abnormal, the CNN model classifies more severe abnormalities as abnormal too.

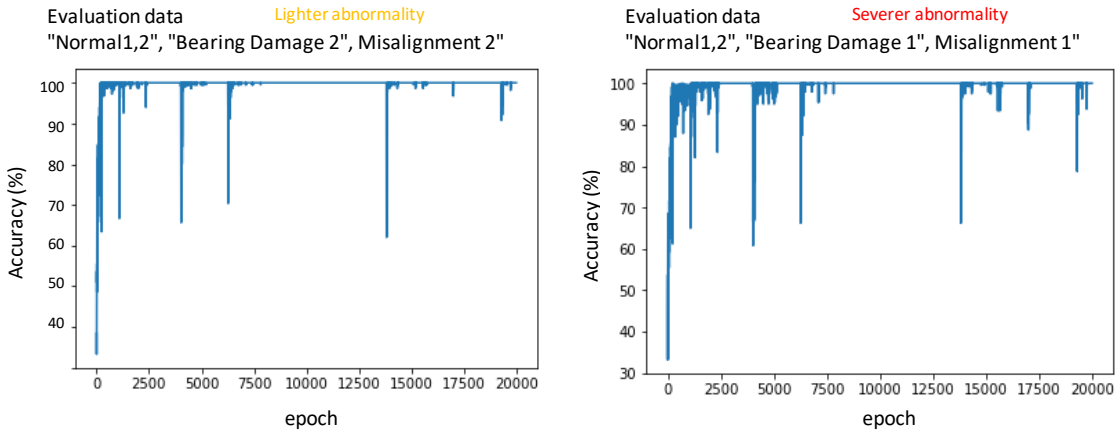


Fig. 3-22 Accuracy of classification by the CNN model trained with lightly abnormal data (20000 epochs)

3.3.3. Summary of state classification of a pump by Neural Networks

State classification of a pump was performed by applying two different types of neural networks, Fully Connected Neural Networks and Convolutional Neural Networks, to acoustic data.

Although each state of the pump includes data in various conditions, such as the position of the microphone and the presence of noise, and that it is difficult to identify the state by the signal level only, it was shown that the states of the pump can be identified with very high accuracy by both types of neural networks.

In the evaluation by FCNN, with the frequency spectrum as input, the optimal input data were calculated in order to obtain knowledge about the internal learning process. Knowledge about the characteristics of the spectrum was obtained that can be used to develop or improve a method for monitoring equipment.

Classification by NN was shown to have a good ability of extrapolation, that is, it is possible to classify abnormal data as abnormal even if this type of abnormal data were not included in the learning process. This is a desirable characteristic for the application of the method to condition monitoring.

This research introduced a newly developed method that reshapes time series data into matrices and applies CNN techniques of image recognition. Using this method, it was possible to classify the states of a pump with very high accuracy without having to use the frequency spectrum as input data. Although verification is necessary, since the frequency spectrum assumes a stationary signal, this method may be superior to the conventional method using the frequency spectrum even if the actual data are not stationary.

3.4. Abnormality detection by model trained with only normal data

Although it has been shown that the state of the pump can be classified with high accuracy by NN trained with data of each of the states "Normal", "Bearing Damage" and "Misalignment", it is usually difficult to obtain actual abnormal data from equipment in operation. Therefore, the possibility of detecting abnormalities with a model trained only with normal data would be useful.

From this point of view, the performance of Auto Encoder, a method that makes use of neural networks and that can be applied to this problem, is evaluated.

3.4.1. Auto Encoder

Auto Encoder is a method of dimensionality reduction of data by a neural network. Fig. 3-23 shows a schematic diagram of Auto Encoder.

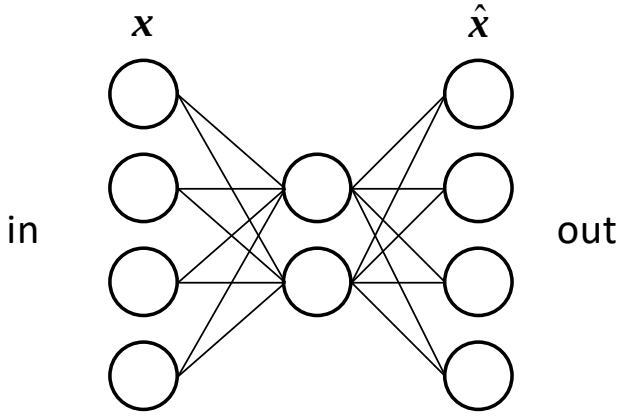


Fig. 3-23 Auto Encoder

The dimension of the input layer (the number of nodes) and the dimension of the output layer are equal, but the dimension of the hidden layers is smaller. Learning is done so that the difference between the input and output is minimized. If the input data can be accurately reproduced even after passing through the lower dimension hidden layers, it means the dimension reduction is successful. The compressed data in the hidden layers can be regarded as the main features extracted from the input data.

In order to detect abnormalities using Auto Encoder, first a model is trained with normal data that are compressed and reproduced. Then, arbitrary data are input to the trained model and the difference between the input and output data is evaluated. If the data are normal, the difference is expected to be small, because the model is trained so that normal data can be correctly reproduced. If the data are abnormal, the data cannot be accurately reproduced after compression, and the difference is expected to be large. Let $\boldsymbol{x}$ be input and $\hat{\boldsymbol{x}}$ be the output obtained from a trained model, the residual is

evaluated as follows.

$$\text{residual} = |\boldsymbol{x} - \hat{\boldsymbol{x}}|$$

3.4.2. Calculation conditions

An Auto Encoder trained only with normal data is used to detect abnormalities using actual experimental data and its performance is evaluated. The experimental data are the same as in section 3.3. For comparison, not only the Auto Encoder but also the Similarity Based Modeling (SBM) and the Mahalanobis distance are evaluated. It is expected that these methods can detect abnormalities using a model trained only with normal data. Details of each method are included in last year's report.

Another method, the Kernel Density Estimation (KDE), that is applicable to this problem and was among the methods studied last year, was considered too. However, its results are omitted as they were not meaningful. This is due to the fact that the dimension of the input vector used for this evaluation is 100, which is much larger than in the case of last year's study. KDE is a method that assumes an N dimensional Gaussian distribution centered on each input data in the N dimensional space and that evaluates the sum of these distributions as a probability density in the whole space. As the dimension of data increases, setting the appropriate width of the Gaussian distributions becomes more challenging.

3.4.2.1. Input data

Time series of duration 0.1 second (2000 samples) are first RMS-normalized and their frequency spectrum is determined. The spectrum is then divided into 100 parts and the RMS of each part is calculated to build a 100 dimensional input vector. This processing is common to all three methods (Auto Encoder, SBM and Mahalanobis distance).

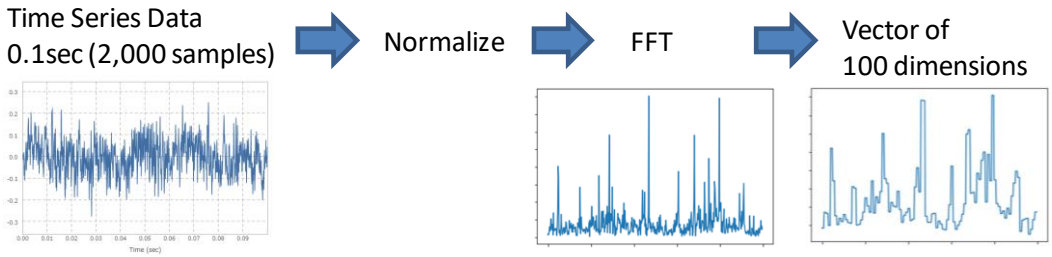
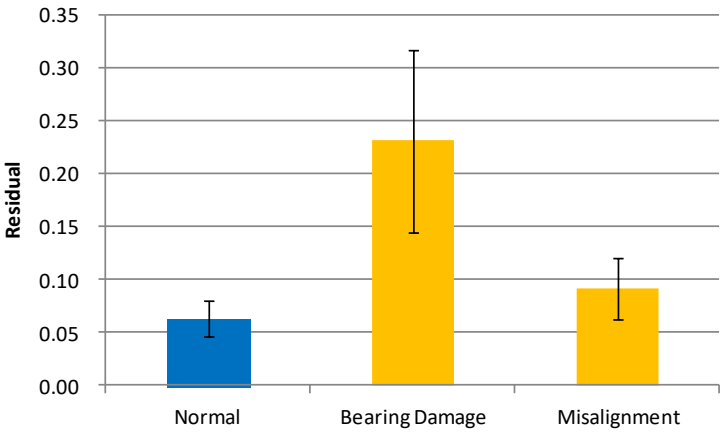


Fig. 3-24 Processing of time series

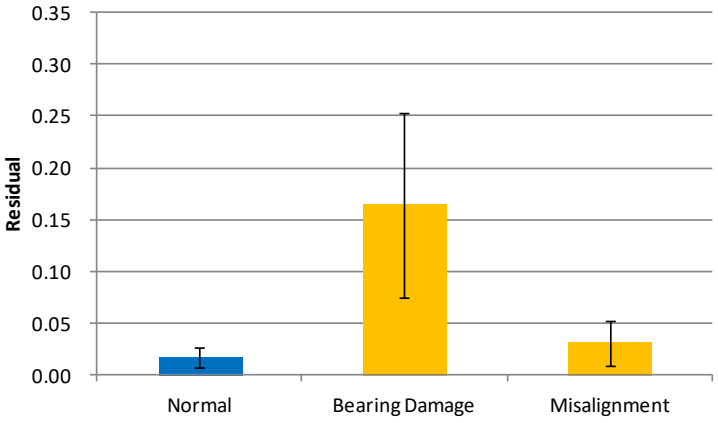
Learning is performed using 60% of normal data, the remaining 40% of normal data and all abnormal data being used as evaluation data. The amount of normal data (input vectors) for learning is 960, and 640 for evaluation. The amount of abnormal data is 1600 for each abnormal state "Bearing Damage" and "Misalignment".

3.4.3. Calculation results

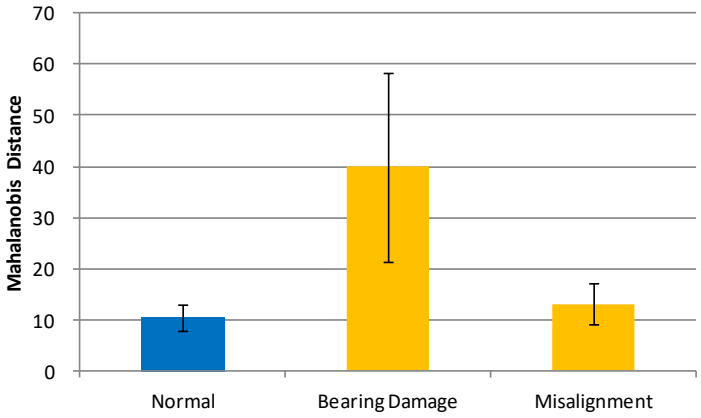
Fig. 3-25 shows the results of evaluation of each method. In the case of Auto Encoder and SBM, the Euclidean distance between the input and output vectors is calculated. For the Mahalanobis distance, the Mahalanobis distance between the input and output vectors is calculated. Error bars in the figures corresponds to ±1 standard deviation. As a reference, the signal level for each state is also shown in Fig. 3-26.



(a) Auto Encoder



(b) Similarity Based Modeling



(c) Mahalanobis Distance

Fig. 3-25 Evaluation results with model trained only with normal data

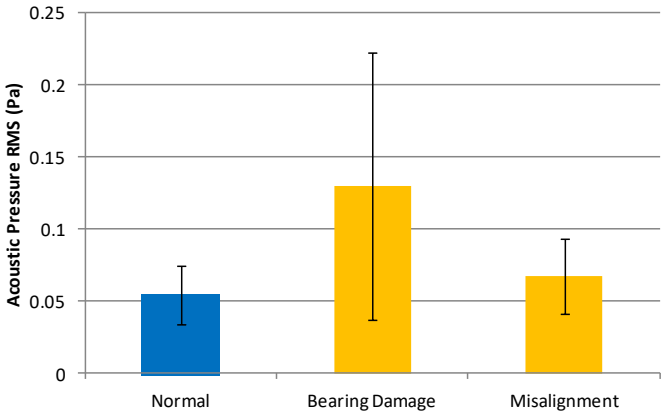


Fig. 3-26 Signal level (RMS)

Similar results are obtained with the three methods (see Fig. 3-26). The difference between bearing abnormality and normal states is large and detection is easy. On the contrary, the difference between the misalignment and normal states is small and detection is difficult.

Although the same can be said from the signal level (see Fig. 3-26), the difference between normal and abnormal states is not as large as with the three methods and standard deviation is larger.

The threshold for deciding that the state is abnormal is defined as the average value plus twice the standard deviation in the normal state. The state is judged abnormal if the relevant value of each method exceeds this threshold. The percentage of data judged abnormal, bearing damage or misalignment, is summarized in Table 3-5. There are no significant difference in the ability of detection of bearing damage between the three methods. Although detection of misalignment is difficult for any method, Auto Encoder exhibits the highest detectability.

Table 3-5 Accuracy of detection of abnormality

	Accuracy of detection of Bearing Damage	Accuracy of detection of Misalignment
Auto Encoder	94.8%	40.3%
Similarity Based Modeling	97.8%	27.8%
Mahalanobis Distance	93.8%	22.1%
Signal Level (RMS)	48.1%	12.5%

3.4.4. Summary of abnormality detection using only normal data

When only normal data are used for training the model, detection of bearing damage is possible with high accuracy, but detection of misalignment is more difficult. This is because the change in the frequency spectrum from the normal state is more subtle in the case of misalignment than in the case of bearing damage. Auto Encoder showed the highest detectability of misalignment compared to SBM and Mahalanobis distance. This implies that Auto Encoder is able to extract more relevant features from the frequency spectrum than the other methods.

4. Comprehensive study

4.1. Summary of the study of neural networks for abnormality detection

In this research, we evaluated the applicability of neural networks to the detection of abnormalities in equipment and developed a method using simulation and experimental data.

4.1.1. Ability of extrapolation

First, using simulation data, we evaluated how neural networks classify or estimate data deviating from the training data. Conclusions were drawn for both the classification and regression problems.

1) For the classification problem, neural networks exhibit a good ability of extrapolation. This is a desirable characteristics for the application to the detection of abnormalities in equipment, as it can be expected that abnormal data that were not included in the training data will be still classified as abnormal.
2) For the regression problem, neural networks exhibit a poor ability of extrapolation. Care must be taken when using neural networks for such applications.

The reason for the difference of the ability of extrapolation is that in the case of the classification problem, the result is determined by comparing the output value of all the nodes of the output layer, whereas in the case of the regression problem, the output value of the single node of the output layer itself is the result.

The ability of extrapolation in the case of the classification problem was verified with experimental data (see sections 3.3.1.4 and 3.3.2.4): when learning the states using only small abnormalities, larger abnormalities were classified as abnormal too with high accuracy.

4.1.2. Abnormality detection by neural networks

Neural networks were applied to acoustic data of a pump, and the detectability of abnormalities was evaluated.

(1) Classification problem

It was shown that when the frequency spectrum of the acoustic data is used as the input and the model is trained with data of each state, "Normal", "Bearing damage" and "Misalignment", the state can be determined with very high accuracy regardless of the

presence or not of noise or of the distance between the microphone and the pump.

A newly developed method, which applies CNN, a technique commonly used in image recognition, to time series converted into images, also showed high accuracy of classification of normal and abnormal data.

The method using the frequency spectrum as input assumes a stationary signal. On the contrary, since the newly developed method uses time series as input, it is expected to be more effective with non-stationary signals and more complex problems such as abnormality detection of equipment operating condition of which dynamically changes

(2) Abnormality detection by model trained only with normal data

When only normal data are used for training the model, the accuracy of detection of abnormalities is lower than in the case when abnormal data are used for training too. When using experimental data, bearing damages can be detected with high accuracy, but misalignment is more difficult to detect. Auto Encoder (neural network) showed the highest detectability of misalignment compared to SBM and Mahalanobis distance. This implies that neural networks extract more relevant features from the input data than the other methods.

4.2. Summary

4.2.1. Applicability of neural networks

When data of each state of equipment is available, neural networks are able to identify the state with very high accuracy. When only normal data are available, although the accuracy decreases, it is still possible to detect abnormalities with a higher accuracy than with other machine learning methods. This makes neural networks a powerful technique for the detection of abnormalities in equipment.

4.2.2. Appropriate usage for each method

During the course of this research that started last year, six machine learning methods were evaluated and their characteristics were investigated. The appropriate usage of each of these methods is summarized as follows.

- Linear Regression, Kernel Regression

It is suitable to use regression analysis when data can be distinguished between explanatory variables and objective variables. In the case of equipment monitoring, the explanatory variables could be the operating conditions of equipment that are not the target of monitoring. An objective variable is a parameter to be monitored such as vibration level.

Linear regression is suitable for learning simple relationships, and kernel regression is suitable for learning more complicated relationships.

If there is no distinction between explanatory variables and objective variables, and all variables are to be monitored, it is more suitable to use the following methods than regression analysis.

- Mahalanobis Distance

Compared to the other methods, a model that extract simpler features from the data is obtained. Moreover, even if there are no parameters to be adjusted, the model is lightweight and relatively effective in many cases.

- Kernel Density Estimation

Although a model that reflects the distribution of data accurately can be obtained, it becomes difficult to adjust the parameters when the dimension of the data increases. It is effective when a complex model is needed and the dimension of data is relatively low.

- Similarity Based Modeling

A complex model of high dimensional data can be trained, and its applicability to abnormality detection is high. Moreover, unlike neural networks, there is no convergence calculation during the training of the model.

- Neural Networks

Although convergence calculation is necessary during learning, a complex model of high dimensional data can be trained, and the size of the model can be relatively small. In this study, neural networks showed better performance than SBM in the frame of abnormality detection with a model trained only with normal data. It can be said that their suitability to abnormality detection is very high.

References

[1]Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of Machine Learning Research 12.Jul (2011): 2121-2159.
[2] Tijmen Tieleman; G. Hinton (2012). Lecture 6.5 - rmsprop, COURSERA: Neural Networks for Machine Learning.
[3] Zeiler, Matthew D. "ADADELTA: an adaptive learning rate method." arXiv preprint arXiv:1212.5701 (2012).
[4] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[5]Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. The Journal of Machine Learning Research, Volume 15 Issue 1, January 2014 Pages 1929-1958

Investigation of change detection and forecasting methods for predictive maintenance

Final Report 2
Change detection and forecasting of time series images

NIHON NO SHORAIWO KANGAERU KAI

Dec.18, 2017

Contents

This is the final report 2 on the basic research for AOARD entitled "Investigation of change detection and forecasting methods for predictive maintenance"

Section 1. Data forecast and abnormal analysis methods for predictive maintenance

1.1 Background and Introduction

An urgent lesson learned from Fukushima Daiichi nuclear power plant accident is what can happen by natural disaster also can be made to happen by human design. The accident raised a fear that terrorists could cause a similar accident by acts of sabotage against a nuclear power plant (NPP) [1], and it is noticeable that threats of terrorism for nuclear security increased after the accident.

In addition, when considering sabotage, due attention should be paid to insiders. Generally, insiders are the individuals with authorized access to nuclear facilities in transport who could attempt unauthorized sabotage. They could take advantage of their access authority and knowledge, to bypass dedicated physical protection elements or other provisions [2]. Thus, we should value the catastrophic consequences of the act of insider sabotage which may lead to loss of safety functions of NPP.

Currently, major countermeasures for insider sabotage are as below:

- Confirmation of the individual trustworthiness of NPP employees;
- Two-man rule;
- Physical protection system (PPS).

However, just consider how to carry out background checks within the context of the strict Personal Information Protection Law in Japan is not enough. And even though the “two-man rule” has some certain effects in restricting access to sensitive areas at NPP by an un-accompanied employee but can be compromised by a trained terrorist who can “take out” an unsuspecting partner [3].

In addition, considering current countermeasures of PPS to insider sabotage, the most significant challenge is how to distinguish ordinary maintenance behaviours and sabotage behaviours since the following difficulties:

- Deterrence is invalid for insider;
- It is difficult to distinguish malicious insider from ordinary maintenance worker and it is not practical to monitor everyone in the NPP;
- Normal behaviours of ordinary maintenance work are complicated and diverse;
- Some sabotage behaviours may be hidden in ordinary maintenance behaviours.

It appears that hand motion has high contribution to human activity and a significant portion of maintenance behaviours and sabotage behaviours such as modification to safety circuit board by using tools and virus contamination on control computer by using USB can be detected through hand motion monitoring.

Thus, the usual protections against insider sabotage are of limited effectiveness. To enhance insider sabotage prevention and maintain safety functions of NPP, a method needs to be developed that automatically and accurately detects sabotage behaviours by hand motion analysis.

1.2 Hand Motion Analysis

1.2.1 Hand Motion Capturing

Given the advancements in computer vision, fingertips position can be used to detect hand motion and recognize hand gesture. The fingertips calculation algorithm in the research based on RGB-D image analysis and Microsoft Kinect v2 is proposed to be used as the surveillance video camera in detection of hand motion for insider malicious behaviors in NPP.

The real-time hand motion capturing system in the research consists of three main components: hand region classification, fingers segmentation and fingers identification. The system is developed using Visual Studio 2015 with C# as programming language. The roadmap of hand motion capturing system development can be demonstrated in Figure 1.1.

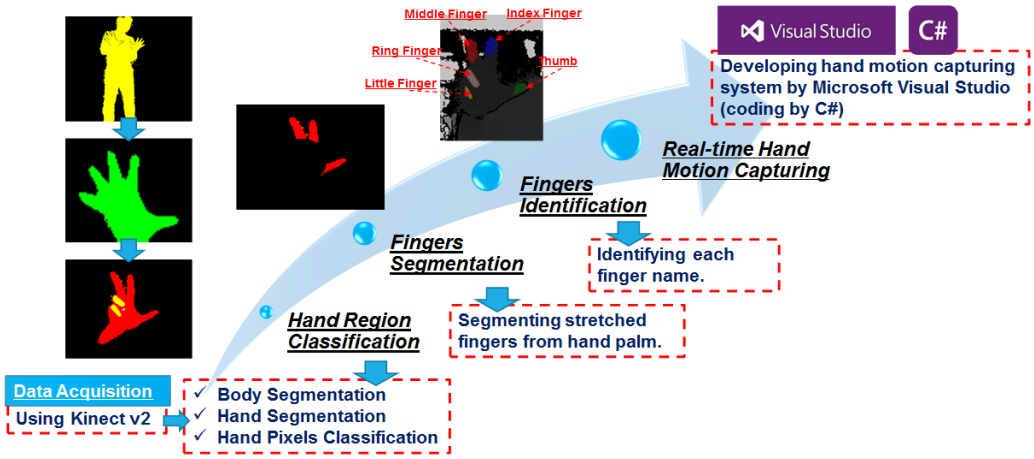


Figure 1.1: Roadmap of Hand Motion Capturing System

1.2.2 Hand Region Classification

The first step is to acquire hand region from each frame of data captured by Kinect v2 and each pixel in hand region should be classified into different parts. Kinect body index frame is used to segment body region from the background. Body index frame is an improved function provided by Kinect v2 which represents a frame with 512×424 resolution that indicates which pixels belong to tracked people and which do not [4]. Each pixel in body index frame will be marked by an index with the following rules:

1) If the pixel belongs to the region of human body, the index is 0~5;
2) If the pixel belongs to the background, the index is 255.

Thus, human body can be segmented from background by using body index frame.

Following this, detailed hand region classification will be performed. With regard to reduce computing complexity, a reasonable small region of interest (ROI) should be extracted. Thus, we proposed a simple algorithm called "Rectangle Limitation Algorithm" to extract ROI and capture specific regions of hands. The algorithm can be summarized as follow:

1) Get centre points of left hand $C_l(x_{C_l}, y_{C_l})$ and right hand $C_r(x_{C_r}, y_{C_r})$ using Kinect skeleton frame.
2) Generate four rectangles from centre point with width W and height H for each hand (The values of W and H depends on the distance from detected body to Kinect camera).
3) Only the pixels within this region can be classified as hand pixels. Thus, left hand pixels $P_l(x, y)$ and right hand pixels $P_r(x, y)$ can be given as follows:

$$P_l(x, y), \left(x \in \left(x_{C_l} - W, x_{C_l} + W\right), y \in \left(y_{C_l} - H, y_{C_l} + H\right)\right) \tag{1.1}$$

$$P_r(x, y), \left(x \in \left(x_{C_r} - W, x_{C_r} + W\right), y \in \left(y_{C_r} - H, y_{C_r} + H\right)\right) \tag{1.2}$$

For each hand, stretched hand region and bend hand region should be divided. For this purpose, we proposed an algorithm based on the Kinect depth frame. Basically, the depth frame can be seen as a device that returns (x, y, z)-coordinates of 3D objects [5]. Hand region classification algorithm can be demonstrated in Figure 1.2. Let $\mathrm{D}(n)$ be the set of depth of left hand pixels $P_l(x, y)$ and right hand pixels $P_r(x, y)$ in depth frame, $\mathrm{I}(n)$ be the set of body index of left hand pixels $P_l(x, y)$ and right hand pixels $P_r(x, y)$ in body index frame, d_{max} be the wrist point which is in the farther distance to the Kinect Camera, d_{min} be the wrist point which is in the closer distance to the Kinect Camera and d is the threshold to calculate bend hand pixels. For each hand pixel's depth value $\mathrm{D}(i)$ in $\mathrm{D}(n)$:

1) If $\mathrm{D(i)} \leq \mathrm{d_{max}}$ and $\mathrm{D(i)} > \mathrm{d_{max}} - \mathrm{d}$, meanwhile, $\mathrm{I(i)} \neq 255$, then I(i) can be classified as the farther distance stretched hand pixels;
2) If $\mathrm{D(i)} \leq \mathrm{d_{max}} - \mathrm{d}$ and $\mathrm{D(i)} > \mathrm{d_{min}}$, meanwhile, $\mathrm{I(i)} \neq 255$, then I(i) can be classified as the farther distance bend hand pixels;
3) If $\mathrm{D(i)} \leq \mathrm{d_{min}}$ and $\mathrm{D(i)} > \mathrm{d_{min}} - \mathrm{d}$, meanwhile, $\mathrm{I(i)} \neq 255$, then I(i) can be classified as the closer distance stretched hand pixels;
4) If $\mathrm{D(i)} \leq \mathrm{d_{min}} - \mathrm{d}$, and $\mathrm{D(i)} \neq 0$, meanwhile, $\mathrm{I(i)} \neq 255$, then I(i) can be classified as the closer distance bend hand pixels.

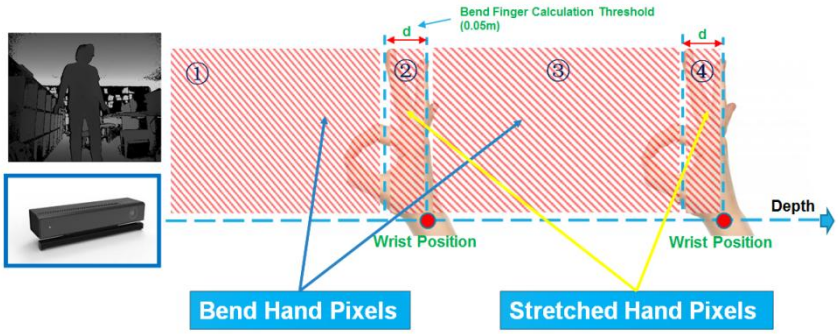


Figure 1.2: Hand Region Classification by Depth

Finally, pixels can be divided into four regions: left stretched hand pixels, left bend hand pixels, right stretched hand pixels and right bend hand pixels. The results of hand region classification are shown in Figure 1.3. (red pixels: stretched hand region; yellow pixels: bend hand region; black pixels: background)

Figure 1.3: Results of Hand Region Classification

1.2.3 Fingers Segmentation

Following hand region classification, the objective of fingers segmentation is to segment all fingers from the palm. Bend fingers region has already been separated from hand palm in hand region classification step (yellow pixels in Figure 3), which allows us just to focus on dealing with stretched fingers in this step. The algorithm of fingers segmentation can be summarized as follow:

1) Firstly, a circle of radius R is draw in each hand, the centers of these circles are set in left hand center point $C_l(x_{C_l}, y_{C_l})$ and right hand center point$C_r(x_{C_r}, y_{C_r})$. The pixels within this area are considered as palm (Figure 1.4-A), others are considered as stretched fingers pixels for further analysis;
2) In practical experiment, some noises may also exist as shown in Figure 1.4-B, but these can be filtered by analyzing body index data and depth data of Kinect;

3) Finally, stretched fingers can be segmented as demonstrated in Figure 1.4-C. Left stretched finger pixel $P_{S_l}(x, y)$ and right stretched finger pixel $P_{S_r}(x, y)$can be given as follows:

$$P_{S_l}(x, y)\left(\sqrt{\left(x - x_{C_l}\right)^2 + \left(y - y_{C_l}\right)^2} < R\right) \tag{1.3}$$

$$P_{S_r}(x, y)\left(\sqrt{\left(x - x_{C_r}\right)^2 + \left(y - y_{C_r}\right)^2} < R\right) \tag{1.4}$$

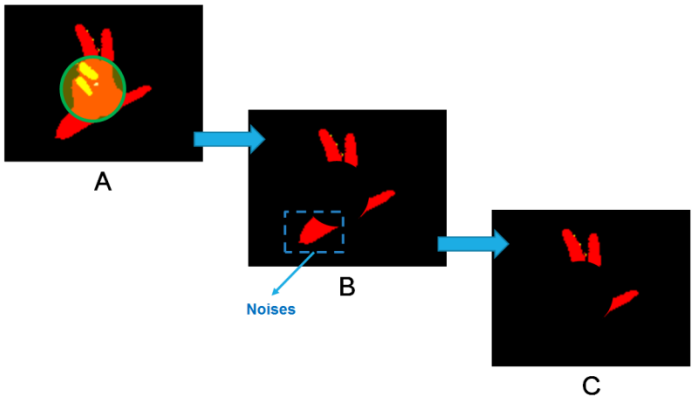


Figure 1.4: Algorithm of Fingers Segmentation

The results of fingers segmentation can be seen in Figure 1.5. We can see that all fingers (both stretched fingers and bend fingers) can be successfully segmented from palm.

Figure 1.5: Results of Fingers Segmentation

1.2.4 Fingers Identification

Fingers can be identified by their specific position. For this purpose, all finger pixels should be classified into different clusters at first. Then fingertips can be calculated and identified.

To conduct finger pixels clustering, K-means clustering algorithm is used in the research. K-means is a clustering method that aims to find the positions $\mu_i (i = 1 \cdots k)$ of the clusters that minimize the distance from the data points to the cluster [6]. As demonstrated in Figure 1.6, the implementing steps can be summarized as follows:

1) K initial "means" μ_i (in this case K=5) are randomly generated within the hand region pixels;

$$\mu_i = r(i = 1 \cdots 5) \tag{1.5}$$

where r is a random value generated for "means" initialization.

3) 5 clusters c_i are created by associating every observation with the nearest mean;

$$c_i = \{j: d(x_j, \mu_i) \leq d(x_j, \mu_l), l \neq i, j = 1 \cdots n\} \tag{1.6}$$

4) The centroid of each cluster μ_i becomes the new mean:

$$\mu_i = \frac{1}{|c_i|} \sum_{j \in c_i} \underline{x}_j \,, \forall i \tag{1.7}$$

where $|c_i|$ is the number of elements in the cluster c_i.

6) Steps 2 and 3 are repeated until convergence has been reached.

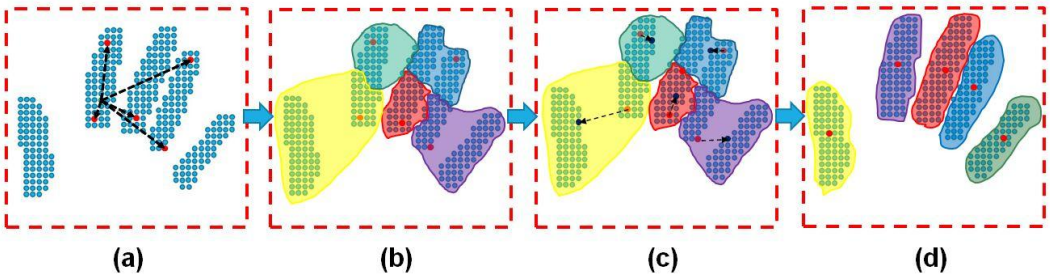


Figure 1.6: Steps of K-means Clustering Algorithm

By analyzing positions of the centroids of each cluster and mating permutation of five fingers to human hand model, fingers can be identified. For each cluster of finger pixels, the fingertip is the pixel which has closest distance to camera.

Finally, a real-time hand motion detection system is developed and positions of each fingertip can be captured with the frame rate of 29.8fps.

1.3. Behavior Recognition

Even though hand motion can be captured, a method to distinguish different motions and recognize sabotage behaviors in NPP is still necessary. From the point of view of pattern recognition, different motion should be classified into different patterns for the purpose of behavior recognition.

Conventional researches of abnormal behavior detection commonly use Static Data Analysis (SDA). However, it is difficult to distinguish malicious behaviors from ordinary maintenance behaviors by static image analysis since some frames of these two types of motions may share some similarity. Compared with SDA, Time-Series Data Analysis (TSDA) can detect more scenes, more detail information and time variation information. However, the large computational complexity of TSDA is an obstacle for real-time calculation. To solve this problem, extracting the features and conducting dimensionality reduction is

necessary. Commonly, a simple method named Principal Components Analysis (PCA) is widely used. However, if nonlinear correlations between variables exist, PCA would be unable to describe data accurately. So, can we find a nonlinear principal component analysis method? Fortunately, high-dimensional data can be converted to low-dimensional codes by Deep Learning (DL) [7].

DL can be considered as an important branch of neural network and the basic architecture of DL is Deep Neural Network (DNN). DNN is simply a feedforward network with many hidden layers. Compared with "non-deep" feedforward neural network, the primary advantage of DNN is that it can have significantly greater representational power.

A Stack Auto-Encoder (SAE) is a DNN consisting of multiple layers of sparse auto-encodes in which the outputs of each layer are wired to the inputs of the successive layer [8]. For each sparse auto-encode, when the number of neurons in the hidden layer is less than the size of the input, the sparse auto-encode learns a compressed representation of the input. Finally, the probability of each pattern can be output by a softmax classifier and different behaviors can be recognized.

In order to recognize different malicious behaviors in NPP, the time-series data of malicious behaviors by hand motion will be used as the trainset to train a SAE and this trained SAE can be used to recognize malicious behaviors.

1.4 Experimental Validation

Equipped with the ability to capture hand motion as well as the recognition algorithm for different behaviors, now it is time to apply this method to a real experiment.

1.4.1 Training Data Capture

Recently, we assumed five malicious motions in NPP for behavior recognition experiment as shown in in Figure 1.7.

1) Cutting motion (by using scissor, etc.);
2) Patting motion (control panel, etc.);
3) Turning motion (switch, etc.);
4) Grasping motion (tools);
5) Pushing motion (buttons).

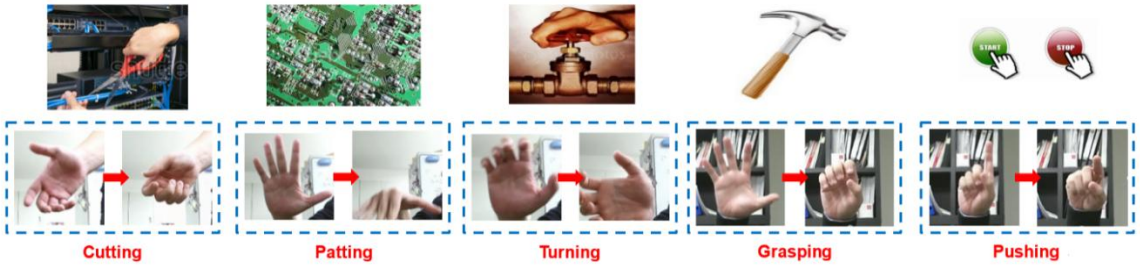


Figure 1.7: *Assumed Malicious Motions*

To capture hand motions for trainset, five people are asked to perform each of the five motions for 20 times with 30bpm (beat per minute) in different distance and angle to camera and each frame of these captured motion data includes 3D positions of five fingers.

1.4.2. Time-Series Data Conversion

All captured hand motion data and testing data will be converted to time-series format. For each frame, the captured hand motion data includes 3D positions of five fingers which can be given by the following 15-dimensional vector:

$$f_i = \begin{bmatrix} x_{i,thumb} \\ y_{i,thumb} \\ z_{i,thumb} \\ \vdots \\ x_{i,little} \\ y_{i,little} \\ z_{i,little} \end{bmatrix} \tag{8}$$

where, i is frame number. Variables of 3D positions (x, y, z) of five fingers are arranged by their order (thumb, index finger, middle finger, ring finger, little finger). In addition, each motion was performed with 30 bpm and the frame rate of our real-time hand motion capturing system is 29.8fps, that is, each cycle of motion formed by 60 frames. Thus, each training sample of the trainset includes fingertip positions from 60 successive frames with 15 variables (3D positions of five fingers) for each frame as follows:

$$s_1 = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ \vdots \\ f_{60} \end{bmatrix}, s_2 = \begin{bmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \\ \vdots \\ f_{61} \end{bmatrix}, s_3 = \begin{bmatrix} f_3 \\ f_4 \\ f_5 \\ f_6 \\ \vdots \\ f_{62} \end{bmatrix}, \cdots, s_j = \begin{bmatrix} f_j \\ f_{j+1} \\ f_{j+2} \\ f_{j+3} \\ \vdots \\ f_{j+59} \end{bmatrix} \tag{9}$$

Conversion from captured hand motion data to time-series format SAE trainset can be demonstrated as Figure 1.8.

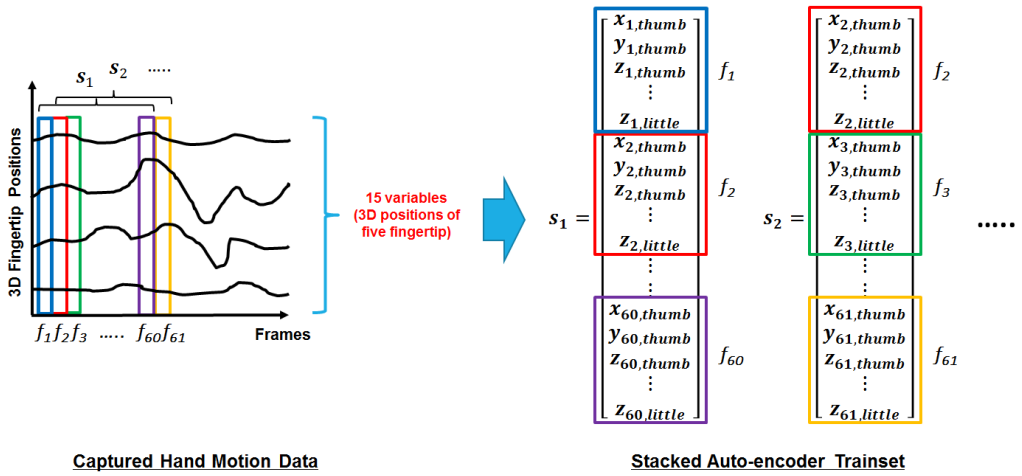


Figure 1.8: Conversion from Captured Hand Motion Data to Time-series Format Trainset

1.4.3. Time-Series Data Conversion

As illustrated in Figure 1.9, a SAE with 2 hidden layers is generated for classification of malicious motions in current experiment.

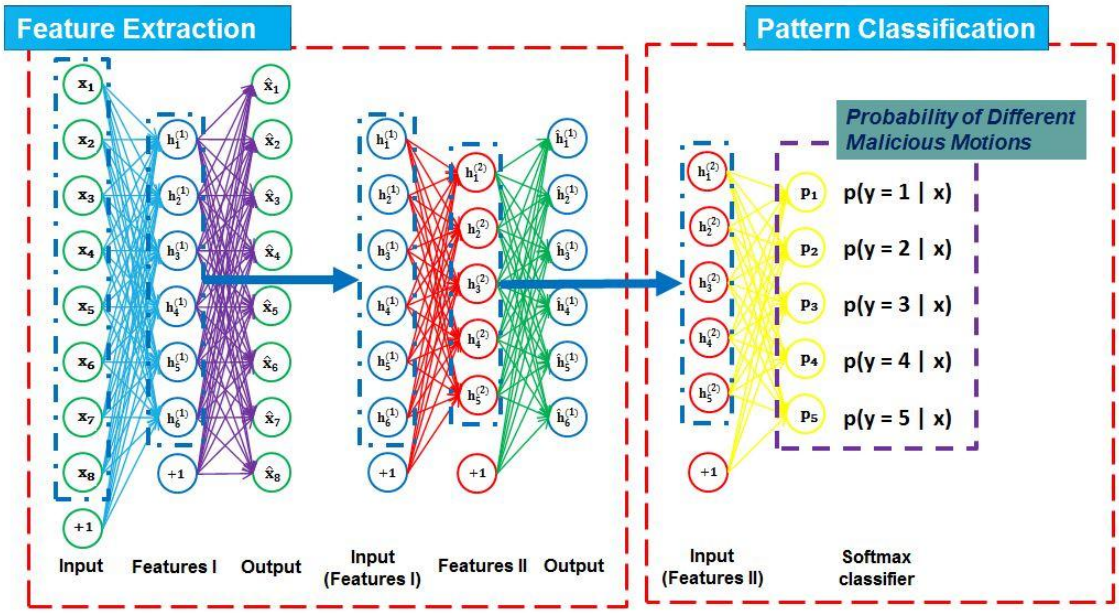


Figure 1.9: Structure of Stacked Auto-Encoder Conducted in Experiment

First, a sparse auto-encoder is trained on the raw inputs $x^{(l)}$ (time-series format SAE trainset, 900 neurons in each sample) to learn primary features $h^{(1)(l)}$ (450 neurons) on the raw input. Next, these primary features $h^{(1)(l)}$ are feed into the second sparse auto-encoder to obtain the secondary feature activations $h^{(2)(l)}$ (200 neurons) for each of the primary features. Following this, these secondary features $h^{(2)(l)}$ are treated as "raw input" to a softmax classier (5 neurons), and it will be trained to map secondary features to digit labels. Finally, all three layers are combined together to form a SAE with 2 hidden layers and a final softmax classifier layer capable of classifying malicious motions.

1.4.4. Detection Results

To verify detection ability of the trained SAE, a time-series testing data is also captured. This testing data are contained with different motion (both malicious motions and ordinary normal motions) in order (Pushing-Normal-Grasping-Normal-Cutting-Normal-Patting-Normal-Turning). Each motion in this testing data is performed 5 times with 30bpm. This testing data of each frame is feed into the SAE to get the output of pattern probability in the softmax classifier.

The accuracy of each motion is calculated and shown in Table 1. For "Pushing Motion", "Grasping Motion" and "Patting Motion", the accuracy is nearly 100%, and the behaviors are perfectly recognized. Also, "Cutting Motion" and "Turning Motion" are recognized with the accuracy of 93.305% and 83.575%. Besides, "Normal Motion" (or "undefined motion" which are the randomly captured motions to demonstrate human ordinary normal behavior) can be recognized with the accuracy of 71.849%. The overall accuracy is 82.555%. In addition, the average calculating time for each frame is 0.0025s, thus real-time calculating is possible in future work.

Table 1. Accuracy of Behavior Recognition

Pushing	Grasping	Cutting	Patting	Turning	Normal
100%	100%	93.305%	100%	83.575%	71.849%

1.5 Robustness Verification

In practice, data loss is a common and serious problem in abnormal detection. So the question remains, can we still achieve satisfied detection results when some frames of data are lost, or the behavior recognition method we proposed is robust or not?

To verify the robustness of the behavior recognition method, different datasets will be generated with randomly selection of the lost frames based on the captured testing data. Two type of data loss problems are considered in verifying method:

1) Discontinuous frames loss problem. A single frame is lost in several continuous frames because of sudden data transmitter failed. This problem may cause intermittent information loss in data acquisition.
2) Continuous frames loss problem. Successive frames are lost in several continuous frames because of motion detection failed (e.g. some parts of human body is hidden in obstacle).

This problem may cause continuous information loss in data acquisition and change pattern of data.

1. Discontinuous Frames Loss Problem

To generate the verification datasets for discontinuous frame loss problem, one frame will be randomly selected as lost frames from each n continuous frames and data of this selected lost frame is replaced by data of the previous frame of this selected lost frame. Three datasets will be generated to verify varying degrees of data loss and 1000 samples will be generated for each dataset:

1) 20% of data loss. Randomly selecting 1 frame from each 5 continuous frames.
2) 33% of data loss. Randomly selecting 1 frame from each 3 continuous frames.
3) 50% of data loss. Randomly selecting 1 frame from each 2 continuous frames.

Figure 1.10 gives a plot of the overall accuracy decreasing in discontinuous frames loss problem. The accuracy decreases only 2.368% (82.555% to 80.187%) in 50% data loss. It is clear that no significant accuracy decreasing showed in verifying results and the behavior recognition method maintains its robustness in discontinuous frames loss problem.

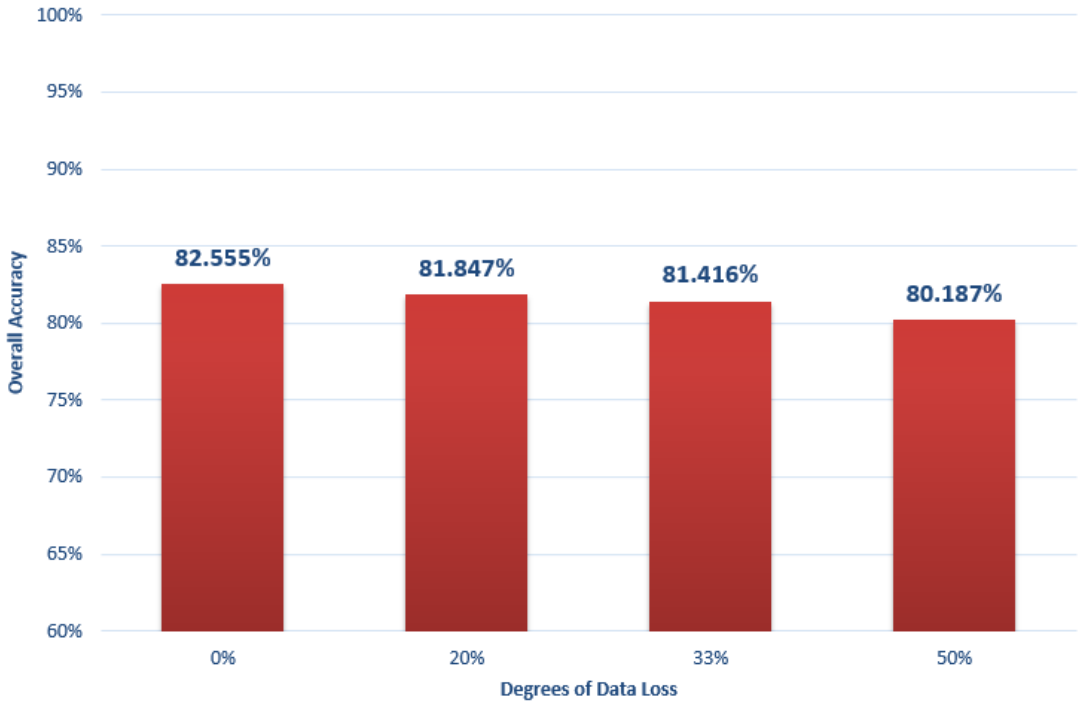


Figure 1.10: Overall Accuracy Decreasing in Discontinuous Frames

Loss Problem

2. Continuous Frames Loss Problem

For continuous frame loss problem, one frame will be randomly selected as lost frames from each n continuous frames and data of this selected lost frames will be replaced by data of the previous frame of the first selected lost frame. Also, following datasets will be generated to verify varying degrees of data loss:

1) 10% of data loss. Randomly selecting 10 continuous frames from each 100 continuous frames.
2) 20% of data loss. Randomly selecting 20 continuous frames from each 100 continuous frames.
3) 30% of data loss. Randomly selecting 30 continuous frames from each 100 continuous frames.
4) 40% of data loss. Randomly selecting 40 continuous frames from each 100 continuous frames.
5) 50% of data loss. Randomly selecting 50 continuous frames from each 100 continuous frames.

The overall accuracy decreasing in continuous frames loss problem can be demonstrated in Figure 1.11. Even though 50% of continuous frames are lost, the accuracy decreases only 19.444% (82.555% to 63.111%). It can be said that the behavior recognition method maintains its robustness in continuous frames loss problem.

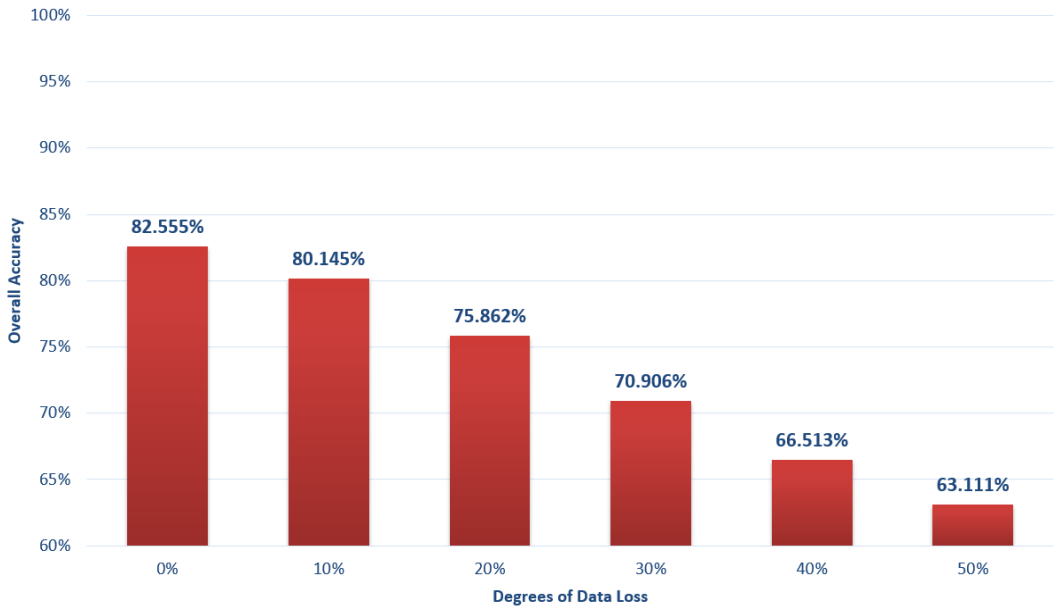


Figure 1.11: Overall Accuracy Decreasing in Continuous Frames

Loss Problem

1.6 Conclusion

For insider sabotage behaviors detection for nuclear security, the following objectives have been accomplished in this research.

1) A new hand motion detection algorithm was proposed based on RGB-D image analysis and both stretched fingers and bend fingers can be detected. Based on this algorithm, a real-time hand motion detection system was developed using Kinect v2 and 3D position of each fingertip can be captured with 29.8fps. Because of the low computational complexity of this algorithm, extra GPU implementation is unnecessary in achieving real-time purpose.

2) A new TSDA based behavior recognition method was developed. We trained a SAE with two sparse auto-encoder and a softmax classifier using time-series format captured hand motion data as trainset. The trained SAE can be used to detect different malicious motions in NPP. In current experiment, all five assumed malicious motions can be classified into different patterns by this trained SAE and detected with high accuracy (82.555% in overall) and speed (0.0023 seconds per frame). Thus, a real-time detection purpose can be achieved in practical implementation.

Section 2. Additional Algorithm for time series image analysis

So far, we used various data available in our database which includes human actions images like hand motion. We also developed some additional algorithm for some advanced motions using image data of organ motion where we can estimate not only the motion but also the image deformation. Hence, in this section we discuss medical image analysis for prediction and anomaly detection.

2.1 Background and Introduction

Last few decades have witnessed significant advances in medical imaging and computerized image processing. These have led to the acquisition of two dimensional, three dimensional or multi-dimensional imaging modalities that have become important tools in diagnosis, planning the treatment and treating the patient.

In our previous granted project, “Dynamic tumor tracking in Radiation Therapy”, we had number of data sets available for several lung cancer patients obtained in the hospitals for diagnostic and treatment procedures. These images consisted of 4D computed tomography images and 2D projection images which were obtained for treatment. In this research, we focus on the prediction of such images and anomaly detection that might occur during the patient treatment [9].

Image guided radiation therapy plays a major role in modern radiation therapy. Modern day treatment machines consists of one or two x-ray source and detectors for in-treatment monitoring. However, organ

motion is still a major challenge. Particularly, in case of tumors like lung tumor, intra-fraction motion caused by patient breathing can affect the treatment outcome. Prediction is important along with the tracking in order to compensate the time lag between during the beam delivery and the target motion. This enables to reduce the exposure of unnecessary radiation to the normal tissues.

Case studies have reported that same patient can have different breathing pattern on different days of the treatment. Breathing pattern changes from large amplitude to small or vice versa can affect the target dose coverage significantly [10]. Hence, it becomes important to find out any abnormality. Along with the intra-fraction motion, patient immobilization also leads to target position uncertainty during the treatment. While it is important to track the moving tumor to reduce the area of irradiation, abnormalities due to irregular breathing motion, sudden cough, and sneeze during the treatment or some patient movement should also be considered as illustrated in Figure 2.1.

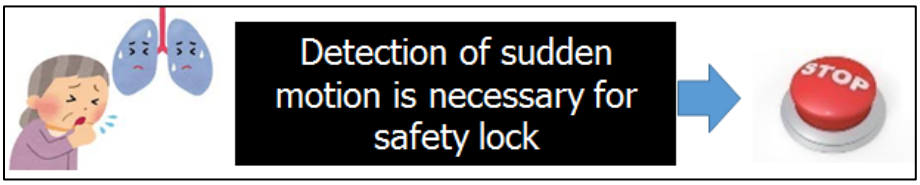


Figure 2.1. Abnormal motion and safety lock system for radiation therapy

Elekta Synergy provides 4D kV imaging which allows 4D volume imaging of moving tumor just prior to the treatment and during the treatment. This allows moving targets to be treated aggressively without compromising the safety of surrounding critical structures.

In this research, we develop a criteria for motion monitoring using the prior kV images to create a boundary for analyzing the abnormal motion due to breathing irregularities or sudden movement during the treatment.

2.2 Image prediction using Principal Component Analysis and Singular Spectral Analysis

Principal component analysis also known as eigen analysis is a multivariate statistical technique used by many scientific disciplines [11]. It is a technique used for data classification and dimensionality reduction without much loss of information. Information means the variation in the data. It is the way of identifying patterns in the data. After identifying the patterns, PCA allows the compression of data by reducing the number of dimensions without much loss of information. To achieve its goal, PCA computes new variable known as principal components which are obtained by linear combination of the original variables. The first principal component has the largest possible variance. The second component is computed under the compulsion of being orthogonal to the first component to have the largest possible variation in the data. The main objective of PCA can be classified as follows.

A. Extracting the most important information.

B. Reducing the size of the data by retaining the most important information.

C. Simplification of the description of data

2.2.1 Image prediction Algorithm

(i). Principal component calculation on 2D images: : The 2D images *I(t)* are converted into 1D image vectors and a single matrix can be constructed as shown in the following equation:

$$X = [X_1, X_2 \ldots\ldots X_n] \tag{2.1}$$

Where *n* is the total number of images. Next, auto-correlation is performed as shown below:

$$Y = [YY^T] \tag{2.2}$$

Where T is the transpose.
(ii) Eigen analysis is then performed in *Y* as shown below; Where Y is represented only by the most dominant Eigen value and vectors of dimension *1 to i*

$$Y = \begin{pmatrix} V_1 & V_2 & Vi \end{pmatrix} \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_i \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \\ \\ V_i^T \end{pmatrix} \tag{2.3}$$

The *i* eigen vectors $V_1 \sim V_i$ can be then arranged to two dimensional matrices to perform the principal component image as explained by the following equation:

$$I(t) = \sum_{j=1}^{i} a_j(t) V_j = a_1(t)V_1 + a_2(t)V_2 + \quad a_r(t)V_r \tag{2.4}$$

Where $a_i(t)$ also known as principal coefficient, can be calculated as following

$$a_i(t) = I(t).V_i^T$$

(ii) Singular spectral analysis (SSA) [12] is a non-parametric modeling method that applies PCA to time series. This method consists in the decomposition of the time series within several components that usually can be identified as trends, oscillatory components or noise components. The SSA algorithm basically depends upon two stages, decomposition and the reconstruction. For the decomposition stage embedding decomposes the original time series data into trajectory matrices and singular value decomposition turns the trajectory matrix into decomposed trajectory matrices which will turn into the trend[13].

SSA here is for the future prediction of $a_i(t)$. $a_i(T\text{-}L+1)$, … , $a_i(T\text{-}L+M)$, …$a_i(T)$ are rearranged perpendicularly to make the vectors $\boldsymbol{b}_{i,1}$, $\boldsymbol{b}_{i,2}$, …, $\boldsymbol{b}_{i,L\text{-}M+1}$ as equation (2.5).

$$\boldsymbol{b}_{i,1} = \begin{pmatrix} a_i(T-N+M) \\ a_i(T-N+M-1) \\ \\ a_i(T-N+1) \end{pmatrix}, \quad \boldsymbol{b}_{i,2} = \begin{pmatrix} a_i(T-N+M+1) \\ a_i(T-N+M) \\ \\ a_i(T-N+2) \end{pmatrix}, \ldots, \quad \boldsymbol{b}_{i,N-M+1} = \begin{pmatrix} a_i(T) \\ a_i(T-1) \\ \\ a_i(T-M+1) \end{pmatrix} \tag{2.5}$$

where M is an embedding dimension that is the size of each vector $\boldsymbol{b}_{i,1}$, $\boldsymbol{b}_{i,2}$, …, $\boldsymbol{b}_{i,L\text{-}M+1}$. These are arranged as equation (2.6) to make B, which is the matrix of R*M rows and L-M+1 lines.

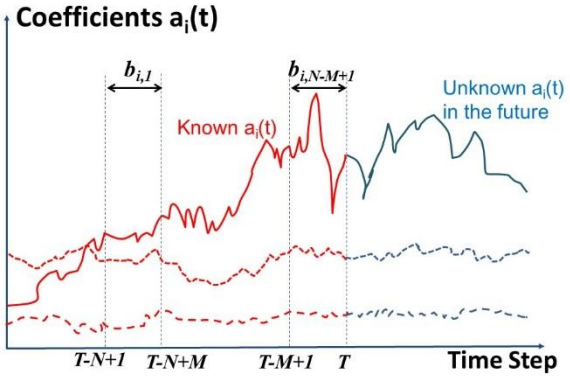


Figure 2.2. Known time-Series of the Coefficients $a_i(t)$ and predicted unknown $a_i(t)$

$$B = \begin{bmatrix} \boldsymbol{b}_{1,1} & \boldsymbol{b}_{1,2} & \boldsymbol{b}_{1,N-M+1} \\ \boldsymbol{b}_{2,1} & \boldsymbol{b}_{2,2} & \boldsymbol{b}_{2,N-M+1} \\ \\ \boldsymbol{b}_{R,1} & \boldsymbol{b}_{R,2} & \boldsymbol{b}_{R,N-M+1} \end{bmatrix} \tag{2.6}$$

The eigen analysis in the autocorrelation matrix BB^t gives the eigen vectors $\boldsymbol{u}_j$ and the eigen values. The r eigen vectors $\boldsymbol{u}_{1, \ldots,} \boldsymbol{u}_r$ with the largest eigen values are arranged horizontally as equation (2.6) to obtain the matrix U,

$$U = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & & \boldsymbol{u}_r \end{bmatrix}. \tag{2.6}$$

The r scores at the future time step t = T+1, $a_1(T+1)$, $a_2(T+1)$, … , $a_r(T+1)$ arranged in column as equation (2.7) is set to the vector $\boldsymbol{A}$.

$$\boldsymbol{A}_{T+1} = \begin{pmatrix} a_1(T+1) \\ a_2(T+1) \\ \\ a_r(T+1) \end{pmatrix}. \tag{2.7}$$

The condition which the vector $\boldsymbol{A}_{T+1}$ should fulfill here is assumed as equation (2.8). This means that the vector $\boldsymbol{A}_{T+1}$ exists closest from the plane by the vectors $\boldsymbol{u}_{1, \dots,} \boldsymbol{u}_r$.

$$\left\{ \left(Q\boldsymbol{A}_{T+1} + \boldsymbol{S} \right) - UU^t \left(Q\boldsymbol{A}_{T+1} + \boldsymbol{S} \right) \right\}^2 \to \min . \tag{2.8}$$

The solution of equation (2.7) gives the vector $\boldsymbol{A}_{T+1}$ as equation (2.9),

$$\boldsymbol{A}_{T+1} = \left(E_r - \boldsymbol{Q}^t UU^t \boldsymbol{Q} \right)^{-1} \boldsymbol{Q}^t UU^t \boldsymbol{S} \tag{2.9}$$

$$Q = \begin{pmatrix} 1 & 0 & & 0 \\ 0 & & & \\ & & & \\ & 1 & & \\ & 0 & & \\ & & & 1 \\ & & & 0 \\ & & & \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ a_1(T) \\ \\ a_1(T-M+2) \\ 0 \\ a_2(T) \\ \\ a_r(T-M+2) \end{pmatrix}, \quad E_r = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ & & \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.10}$$

where Q is the matrix of r*M lines and r columns, $\boldsymbol{S}$ is the vector of r*M lines, and E_r is the unit matrix of size r*r expressed as equation (2.10). The further future of the score, $a_i(T+2)$, $a_i(T+3)$, $a_i(T+4)$, $\dots a_i(T+n)$, can be obtained by repeating calculation from equation (2.7) to (2.10) by n times by substituting the obtained future coefficients into equation (2.7). After repeating the calculation from equation (2.7) to equation (2.10) by n times, the calculation start again from equation (2.1) in order to refresh the principal component vectors and to minimize the prediction error.

2.2.2 Prediction Result

Figure 2.2 and Figure 2.3, shows the comparison of the original time series images for the whole CT images and the region of interest (ROI) respectively. It has successfully been predicted with minimum

error. The calculation time for the whole CT image is around 0.6 sec while for the ROI case is 0.1 sec which is enough to compensate the time lag for the prediction. Cross correlation analysis between the predicted and the original image and the accuracy was up to 99% in each case.

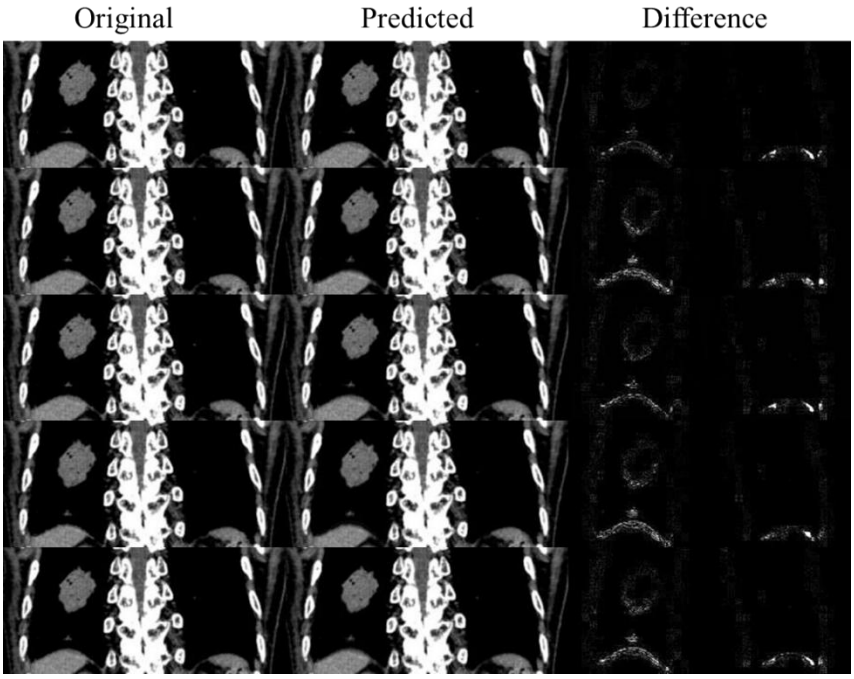


Figure 2.2: Prediction results for whole 4DCT image (left: Original, center: Prediction and right: Difference)

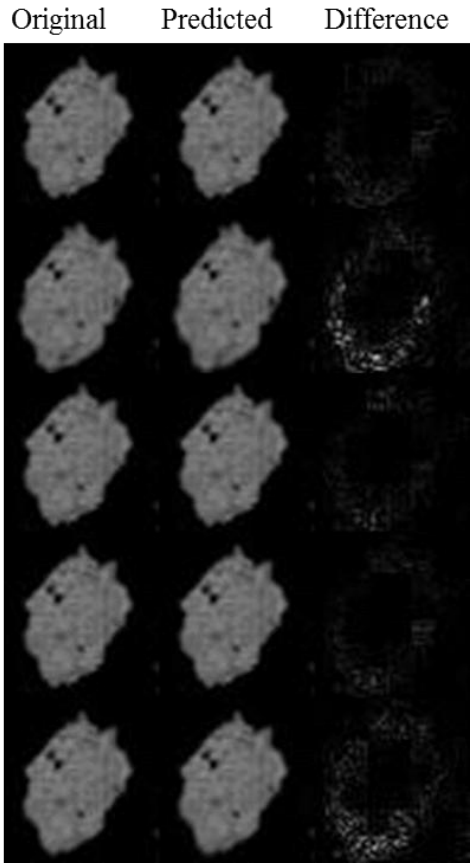


Figure 2.3: Prediction results for ROI 4DCT image (left: Original, center: Prediction and right: Difference)

2.2.3. Limitation

The images has been successfully predicted with minimum error however, with the increase in the number of frames the error probably due to noise keeps on increasing. To consider the use for real-time, this limitation should be overcome probably with some noise reduction method.

So far, this system is only suitable for the fixed x-ray source and detector. In case of rotating gantry system, this algorithm needs to be modified and the angles need to be considered. The algorithm needs to be modified based on the system requirement.

2.3 Anomaly Analysis using cross correlation

After the successful prediction of images, an algorithm for monitoring the organ motion was created in order to apply the beam stop or safety lock system in the case of sudden physical motion or abnormal breathing.

2.3.1 Data Monitoring Algorithm

The developed algorithm for monitoring the motion of the moving tumor is divided into four steps as explained below. (i). Acquisition of prior images: Prior to the treatment images of few lung cancer patients are acquired offline. (ii). creating a boundary: The maximum and the minimum boundary is created using the relative cross-correlation of the prior images. We eliminate the low pass component in the correlation signal using the high pass filter. Maximum and minimum boundaries are acquired by creating a threshold and dividing the signals into different channels.

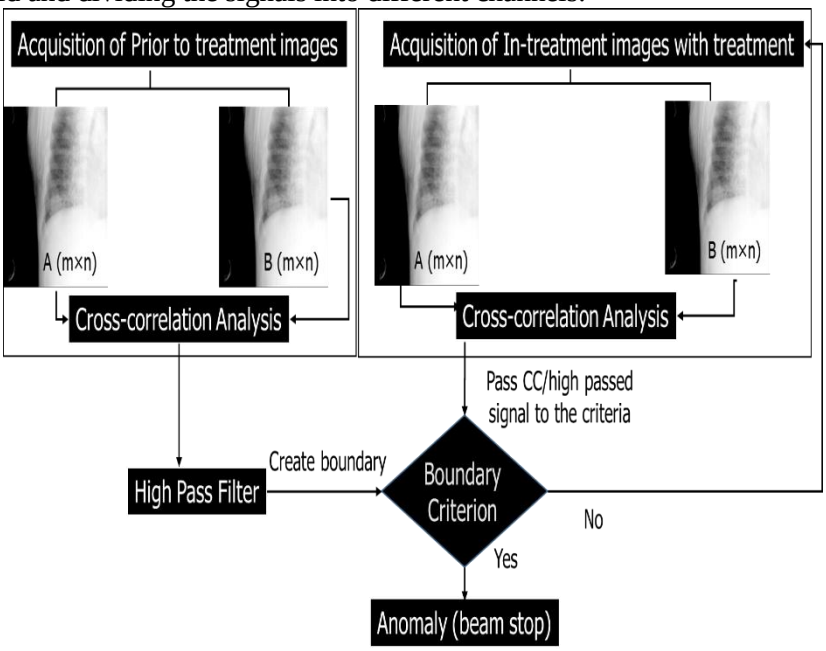


Figure 2.4: Motion monitoring algorithm for detection of any abnormal motion

(iii). Cross-correlation for in-treatment images: Once the boundary is acquired, relative cross correlation is performed on the kV images acquired during the treatment. Set of image datas already available in our database has been used for the time series image analysis . Cross-correlation has been widely used to identify the similarities between the two images. The equation for cross correlation coefficient is shown as :

$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \overline{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \overline{B})^2\right)}} \quad (2.11)$$

Where A_{mn} and B_{mn} are time-series images respectively. Figure 6 shows the algorithm for the relative cross correlation analysis in the time-series where if the cross correlation (CC) value exceeds a certain threshold, it is identified as anomaly or an outlier. This cross-correlation signal passes through the high pass filter and then is passed through the boundary created earlier in step (ii).(iv) Determine the abnormal motion: Any abnormality in the signal can be located if the in-treatment signal crosses the threshold constructed from the prior to the treatment images. The flow of these above mentioned steps in the algorithm is illustrated in Figure 2.4

2.3.2 Monitoring Result

Given the advancements in computer vision, fingertips position can be used to detect hand motion and recognize hand gesture. The fingertips calculation algorithm in the research based on RGB-D image analysis and Microsoft Kinect v2 is proposed to be used as the surveillance video camera in detection of hand motion for insider malicious behaviors in NPP.

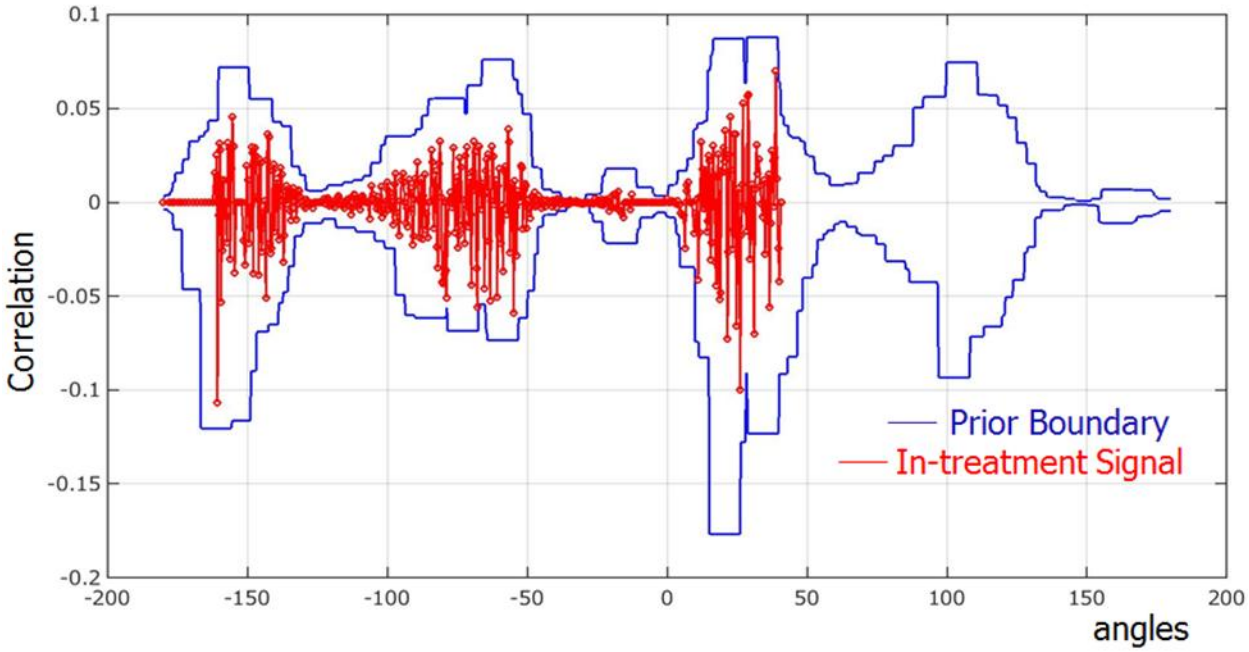


Figure 2.5: Creation of boundary and passing the cross-correlation signal from in-treatment images to the boundary for a normal case

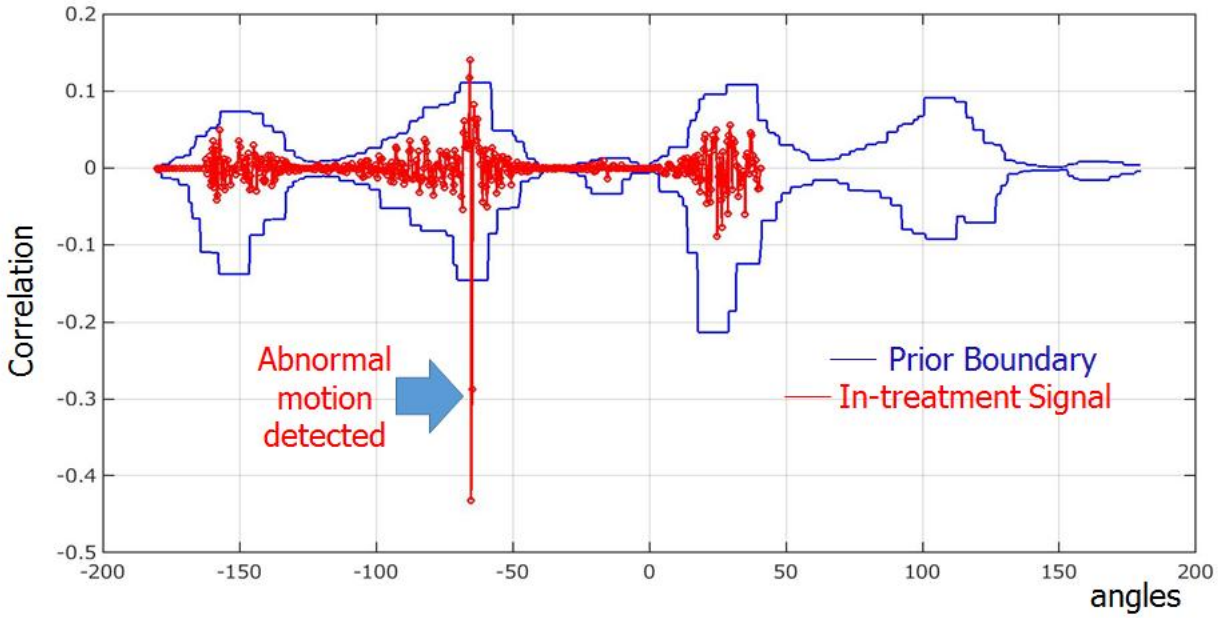


Figure 2.6: Creation of boundary and passing the cross-correlation signal from in-treatment images to the boundary for abnormal case

Number of patient's breathing was analyzed for four days using the prior to the treatment and in-treatment images. The criteria or the boundaries were successfully created using the prior to treatment images, the in-treatment signals are then passed to the boundary as shown in Figure 2.5. The prior boundary is extended to 60% in order to consider the margin concept in radiation therapy.

The number of patients analyzed in the database doesn't show big abnormal motion, hence we created an artificially induced sudden motion in order to verify our algorithm. This artificial sudden motion was induced by changing the order of the images in the data set. The result for the artificially created motion is illustrated in Figure 2.6 where there is an abnormal motion is detected around angle -60 degrees. The motion was well detected once the in-treatment signal passed outside the prior boundary as shown in Figure 2.6.

2.4 Conclusion and future work

This research focuses on the development of algorithms for the forecasting the movement of advanced images such as human lung and to test the feasibility of organ motion monitoring algorithm to detect the anomaly during the treatment in near real time. All our work are performed offline with the available images in our data base.

In case of forecasting using PCA/MSSA, further calculation time reduction a can be achieved by applying combination with other methods such as Optical flow by which only few components can be tracked for the tracking of the entire region of displacement in the image. The combination is also believed to reduce the noise appearing in longer sequences.

In case of the anomaly detection, the results shows the development of criteria for monitoring any anomaly during the treatment process. Several factors such as normalization during beam on/off, individual patient based parameter optimization will be considered next.

References

[1] N. AKIYAMA, H. SATO, K. NAITO, Y. NAOI, T. KATSUTA, The fukushima nuclear accident and crisis management-lessons for Japan-US alliance cooperation, The Sasakawa Peace Foundation, 2012.

[2] M. L. GARCIA, The Design and Evaluation of Physical Protection Systems, Butterworth-Heinemann (2007).

[3] P. L. LEVENTHAL, M. M. HOENIG, Nuclear terrorism: Reactor sabotage and weapons proliferation risks, Contemporary Economic Policy 8.3 (1990): 106-121.

[4] https://msdn.microsoft.com/en-us/library/windowspreview.kinect.bodyindexframe.aspx, (accessed 2016-11-29).

[5] M. R. ANDERSEN, T. JENSEN, P. LISOUSKI, A. K. MORTENSEN, M. K. HANSEN, T. GREGERSEN, P. AHRENDT, Kinect depth sensor evaluation for computer vision applications, Electrical and Computer Engineering Technical Report ECE-TR-6 (2012).

[6] J. MACQUEEN, "Some methods for classification and analysis of multivariate observations", Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 14. 1967.

[7] G. E. HINTON, R. R. SALAKHUTDINOV, Reducing the dimensionality of data with neural networks, science 313.5786 (2006): 504-507.

[8] P. VINCENT, H. LAROCHELLE, I. LAJOIE, Y. BENGIO, P. A. MANZAGOL, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, Journal of Machine Learning Research 11. Dec (2010): 3371-3408.

[9] R.. B. CHHATKULI, K. DEMACHI, M.UESAKA , K. NAKAGAWA AND A. HAGA , Markerless tumor tracking using prior cone beam computed tomography for phase recognition in lung cancer radiation therapy, Journal of Radiation Research (to be submitted).

[10] PHAM D, KRON T, FOROUDI F, SIVA S. Effect of different breathing patterns in the same patient on stereotactic ablative body radiotherapy dosimetry for primary renal cell carcinoma: a case study. Medical Dosimetry. 2013 Nov 30;38(3):304-8.

[11] M. TURK AND A. PENTLAND, Eigenfaces for Recognition, Journal of Cognitive Neuro- science, Vol.3, No.1, pp.71-86, 1991.

[12] R. VARTARD AND M. GHIL, Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series Physica D, 35, pp.395-424, 1989.

[13] K. DEMACHI, H. SAKAKIBARA AND R. B. CHHATKULI , Development of Future Prediction Method for Video by Principal Component Analysis of Images and Singular Spectral Analysis of Time Series Data, Signal Image and Video Processing Journal, (under review).