Machine Learning Methods for Abnormality Detection in Hard Disk Drive Assembly Process

Mr. Masayuti Simongyi

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2018

การตรวจจับความผิดปกติในการประกอบฮาร์ดดิสก์ด้วยการเรียนรู้ของเครื่อง

นายมะสายุตี แสมงยี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561

Thesis Title	Machine Learning Methods for Abnormality Detection in Hard Disk Drive Assembly Process
By	Mr. Masayuti Simongyi
Field of Study	Computer Science
Thesis Advisor	Professor Prabhas Chongstitvatana, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

..... Dean of the Faculty of Engineering
(Associate Professor SUPOT TEACHAVORASINSKUN, Ph.D.)

THESIS COMMITTEE

..... Chairman
(Assistant Professor Sukree Sinthupinyo, Ph.D.)
..... Thesis Advisor
(Professor Prabhas Chongstitvatana, Ph.D.)
..... External Examiner
(Associate Professor Worasait Suwannik, Ph.D.)

มะสายุตี แสมงยี : การตรวจจับความผิดปกติในการประกอบฮาร์ดดิสก์ด้วยการเรียนรู้ของเครื่อง. (Machine Learning Methods for Abnormality Detection in Hard Disk Drive Assembly Process) อ.ที่ปรึกษาหลัก : ศ. ดร.ประภาส จงสถิตย์วัฒนา

งานวิจัยชิ้นนี้เสนอวิธีการตรวจจับความผิดปกติในการประกอบฮาร์ดดิสก์ ด้วยวิธีการการเรียนรู้ของเครื่องแบบต่าง ๆ วิธีการเรียนรู้ของเครื่องสามประเภทถูกเสนอเพื่อนำมาใช้ในการจำแนกประเภทของฮาร์ดดิสก์ที่ได้จากการประกอบ โดยจำแนกเป็นสองประเภทคืองานประกอบดีและงานประกอบเสีย โดยฮาร์ดดิสก์ที่จัดอยู่ในงานประกอบดีนั้น เป็นงานที่ทุกชิ้นส่วนในฮาร์ดดิสก์ถูกประกอบอย่างถูกต้องและถูกติดตั้งอย่างเหมาะสม ในขณะที่งานประกอบเสียคือ เป็นชิ้นงานที่ชิ้นส่วนบางอย่างในฮาร์ดดิสก์ถูกลืมประกอบ หรือประกอบแล้วแต่ถูกต้องตั้งอย่างไม่เหมาะสม กระแสขดลวดเสียงถูกนำมาใช้เป็นข้อมูลสอนและข้อมูลทดสอบ โดยกระแสขดลวดเสียงนั้นได้มาจากการวัดและเก็บค่าจากฮาร์ดดิสก์ที่อยู่ในสายการผลิต แต่เนื่องจากว่าจำนวนของงานประกอบเสียที่เก็บได้มีค่าน้อยกว่าจำนวนของงานประกอบดีมาก ซึ่งทำให้เกิดปัญหาความไม่สมดุลของตัวอย่างสอนในระหว่างกระบวนการสอนด้วยเทคนิคการเรียนรู้ของเครื่อง ดังนั้นในงานวิจัยชิ้นนี้จึงได้ทำงานทดลองปรับเปลี่ยนอัตราส่วนและจำนวนของข้อมูลสอนและข้อมูลทดสอบด้วย

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2561 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

5971019921 : MAJOR COMPUTER SCIENCE
KEYWORD: Hard Disk Drive, Voice Coil Motor, Detection, Support Vector Machine, Convolutional Neural Network, LSTM

Masayuti Simongyi : Machine Learning Methods for Abnormality Detection in Hard Disk Drive Assembly Process . Advisor: Prof. Prabhas Chongstitvatana, Ph.D.

The research proposes a method to detect abnormality in assembly of a hard disk drive. Three machine learning techniques are employed to classify the drives from assembly process into good and bad class. A good class represent a drive that all components are properly installed while a bad class represent an abnormal drive that some components are missing or improperly installed. The voice coil motor current motor is measured and collected from physical drives in assembly line for using as training and testing data set. Since the amount of bad drives in hard disk drive assembly process are much smaller than the good drive which introduce the imbalance problem during training process, this paper also set the experiment of varying amount of training data set that can satisfy the training in practical hard disk drive assembly process. Bidirectional Long Short-Term Memory, Wavelet transform with Convolutional Neural Network and Support Vector Machine are chosen as proposed machine learning models to classify this task. The comparison between each technique is discussed.

Field of Study:	Computer Science	Student's Signature .....
Academic Year:	2018	Advisor's Signature .....

ACKNOWLEDGEMENTS

I was really fortunate that got the scholarship from Western Digital Corporation to further my study on this very interesting research areas. This research is like my first invisible door that widen my perspective on the world of research, computer science and data technology at its very dawn age. After this, I believe that there would be many invisible doors of challenges and opportunities in these fields that waiting me to discover and dive into it deeper and deeper until I find what I would satisfy and return back to the initial state like depth-first search.

In this opportunity, I would like to express my deepest gratitude to my advisor Professor Dr.Prabhas Chongstitvatana for enlighten me to this research area and gave me a lot of valuable, insightful advises and guide me to correct and effective direction since my day one of the research. I also would like to thank Assistant Professor Dr.Sukree Sinthupinyo who taught me several courses during my graduate study, his lectures are always interesting and intuitive and Associate Professor Worasait Suwannik for giving their time to be the committees for my thesis proposal and gave a good comments and suggests that I can improve in my final works.

I also would like to give my sincere thanks to my supervisor Preeda Meekangvan and Talent Management and Organization Development team at WDC for helping and supporting me for this graduate study program.

In closing, I would like to send my heart-felt thank you to my family, friends, teachers and people at Western Digital Corporation who have supported me over the past few years.

Masayuti Simongyi

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

Chapter 1

Introduction

1.1 Background

In the process of manufacturing and assembling the hard disk drive. It involves the assembly of various hard disk drive components together. Hard disk drive consists of the following main components: voice coil motor (VCM), pivot, ramp and actuator arm, etc. In addition to the main components that make the hard disk drive work properly. The hard disk drive also contains components used to prevent damage to the read/write heads and the disks. For example, a latch has the function of preventing the read/write head from get off garage when the shock occurs. A breather filter acts to prevent leaks of helium gas while writing a servo signal. The breather filter is required to be installed and align with top cover properly otherwise it may be hit by a sliding actuator arm, which is frequently moved in and out, because of the breather filter is mounted on top cover nearby position of read/write head. The collision between the actuator arm and the breather filter not only cause the damage to the head but also produces the particle that circulate inside the hard disk drive. This particle has been proven to be a major cause of head and disk media collision while writing servo signal.

From above, it was found that the problems encountered in assembling the hard disk can be grouped into two categories. The first is that the parts are not completely assembled. Most of them are small pieces that are unnoticeable. For example, a latch which event though it has visual mechanic inspection with human eyes before closing hard disk drive top cover but in some cases, there still have a problematic hard disk drive that can pass through this check and continue to the next process. The second category of problem is that all parts are completely assembled but some parts are not properly installed, for example, installing a breather filter that mentioned previously. In this study, we label the drives that fall into these categories as a “Bad” class drive and label the remaining drives that pass server writing process as “Good” class, we excluded the drives that are completely and properly installed but fail during servo writing process from this study.

Three Machine learning techniques are employed for detecting this fault. The techniques are applied to the process of writing the servo signals which is done after the hard disk components are completely assembled from the clean room.

The first technique is Bidirectional Long Short-Term Memory (Bi-LSTM), this technique is well-suited to classifying, processing and making decisions based on time series data because it can learn long-term dependency between each step input of sequence data and as VCM current data set is the time series of current during load operation, so this technique is promising.

The second technique is Wavelet transform with convolutional neural network (CNN) using the transfer learning from GoogLeNet. In this study it is difficult to find a sufficient amount of drives to use as training data for a deep CNN since its requires a large amount of training data set and it is computational expensive when training it from scratch. So, we leverage the existing neural network GoogLeNet that have been train on large data set adapt to our classification technique as pretrained network for

image recognition. GoogLeNet is a deep CNNs originally design to classify image in 1000 categories. We reuse the network architecture of CNN to classify binary class of VCM current data based on the images from continuous wavelet transform of VCM current data. GoogLeNe is winner of imageNet Large Scale Visual Recognition Competition (ILSVRC) in 2014 so it is promising to use this pretrained network to our CNN classification.

The last technique that we proposed is Support Vector Machine (SVM) which is well-known for binary classification that give the largest margin between the classes. Usually SVM have a good classification on linearly separable data set and can use a kernel trick to solve that problem that not linearly separable.

Due to the difficulty of finding abnormality drives from assembly process because the failure rate is very small. So, this study uses much less number of bad drives compare to good drive for training the models. This introduce the imbalance problem to the training process. To tackle with this imbalance problem, we create three experiment groups which each group have varying number of training dataset and varying proportion of good and bad drives and we evaluate in in the precision and recall of both classes.

1.2 Research Purpose and Objectives

The research proposes a method to detect abnormality in assembly of a hard disk drive. The hard disk drives from assembly process are categorized into good and bad class. A good class represent a drive that all components are properly installed while a bad class represent an abnormal drive that some components are missing or improperly installed. The voice coil motor current motor is measured and collected from physical drives in assembly line for using as training and testing data set. Bi-directional Long Short-Term Memory, Wavelet transform with Convolutional Neural Network and Support Vector Machine are chosen as proposed machine learning models to classify this task.

1.3 Research Scope and Limitation

1. The research was experimented on the hard disk drive that have only two disk medias (2 platters). The result maybe varies with different number of platters.
2. The research was experiment with firmware and parameters that were being used at the time of experiment. The firmware and parameters can be changes over time.

1.4 Contribution

The research proposes to solve the problem that happen in practical hard disk drive assembly process and the result from this study may be use as feasibility assessment on applying machine learning techniques to detect the very rare defect that can be happened during hard disk drive assembly process.

Chapter 2

Basic Hard Disk Drive Knowledge and Related Works

2.1 Hard Disk Drive Component

Hard Disk Drive (HDD) is an electromechanical data storage device that uses magnetic storage to store and retrieve digital information using rigid rapidly rotating disks coated with magnetic material. Figure 1 show the component of Hard Disk Drive which have following main component parts.

1. **Read/Write Heads** are an interface between the magnetic disk where the data is stored and electronic components in the hard disk drive. The heads convert the information, which is in the form of bits to magnetic pulses when it is to be stored on the disk and reverses the process while reading.
2. **Actuator Arm** is a permanent magnet and moving coil motor which moves and control Read/Write Heads into desired position.
3. **Disk** is a circular metal that is mounted inside a hard disk drive. Several disk are mounted on a spindle motor to increase more data storage. The disk has a core made up of aluminum or glass substrate.
4. **Spindle Motor** is a high-precision, high-reliability electric motor that is used to rotate the shaft on which the disk are mounted in a hard disk drive (HDD).

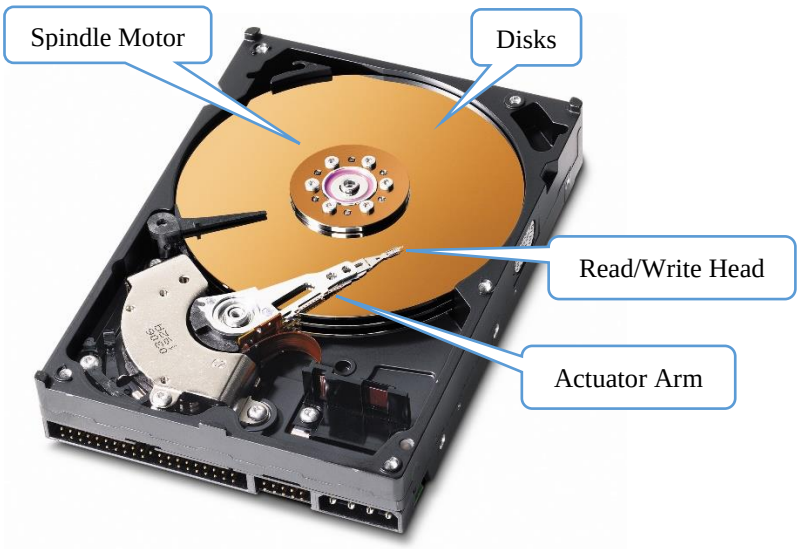


Figure 1 Basic Hard Disk Drive Components.

2.2 Hard Disk Drive Manufacturing Process

The hard disk drive assembly process must be done in a clean room. After the components are cleaned, the first step is to install the disk media and then the disk clamp to hold the disk media together. After that, it will be verified that if the hard disk drive media has been properly installed before installing the ram and other devices, as shown in Figure 2. It can be seen that in the hard disk drive assembly

process it is not only involve the installation of each component together, but it is also include of the verification of the component that has been installed. For example, the verification of disk media balance and the leak test but there are also a number of components that could not be verified. For example, a ram, voice coil motor, actuator arm, read/write head and a latch. This is because there is no proper way to verify these components better than visual mechanical inspection.

After all the hard disk drive components are assembled, they are closed with the top cover and passed to the next process with printed circuit board assembly (PCBA). It's ready for supply the voltage and the reader is pushed into the disk media to write a servo signal

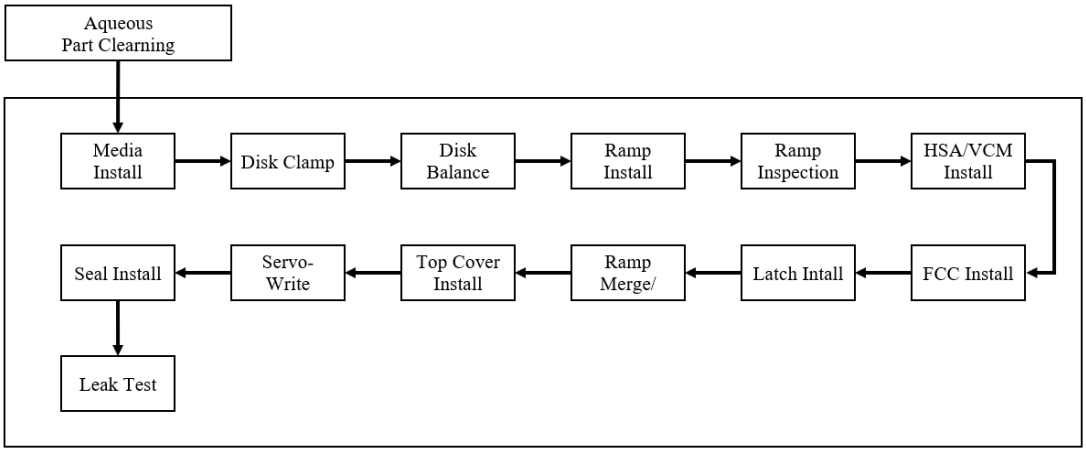


Figure 2 Process of assembly hard disk drive in clean room.

2.3 Head Loading Operation in Hard Disk Drive

To read and write servo signal on disk media, the first step is to spin up spindle motor to make it spin up at constant velocity and then loading the heads onto the disk. If the head is loaded while the spindle motor is not spinning at our target speed it has the potential to cause the read/write head crash with the disk media. Generally, in the firmware of the hard disk drive, it is always necessary to have this verification step before loading head onto the disk.

When the spindle motor is spinning at speed, the current is supplied to voice coil motor so that the actuator arm is pushed onto the disk. This process requires a different level of current because the force that need to push actuator out of garage until get actuator onto the disk is varying and because of the voice coil motor itself is a like coil which generate the back EMF when supply the current through it. So, it is important to have close loop system built in as shown in Figure 3 to accurately control the speed of the actuator.

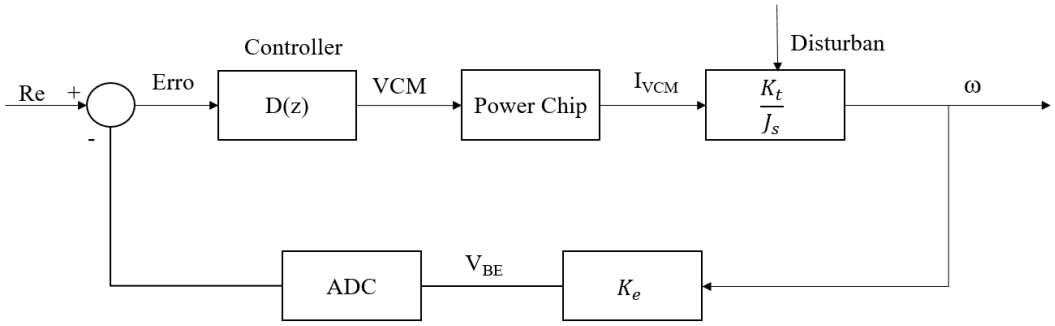
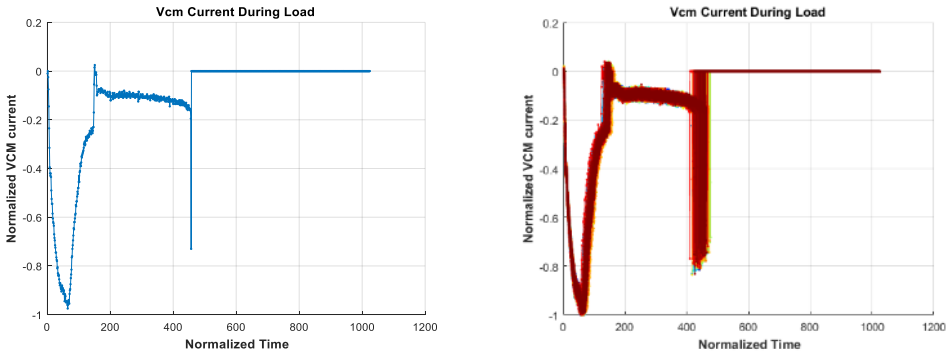


Figure 3 Closed-loop control system of actuator arm.

2.4 Voice Coil Motor Characteristic

The current that supply to voice coil motor since head parking on the garage until head successfully load onto disk media is plot and shown in Figure 4(A). At the beginning, enough amount of current need to be supplied to make actuator out of garage and latch and then the current needed is gradually reduce until close to zero when head is on disk media and detect the crash stop.

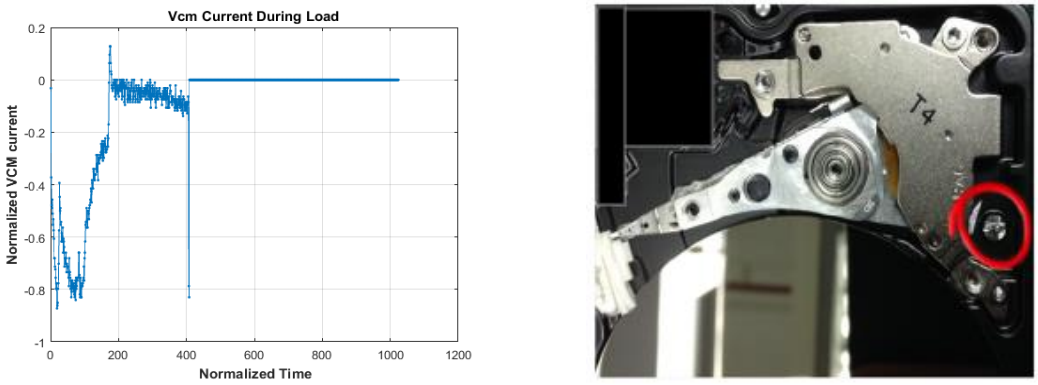
When we collect voice coil motor current during loading head operation over 1000 times, the voice coil motor current has a consistent shape as shown in Figure 4(B) this is because the current that supply to the coil has feedback control as described above.



(A) (B)

Figure 4 (A) Single VCM current (B) 1000x overlay of VCM current.

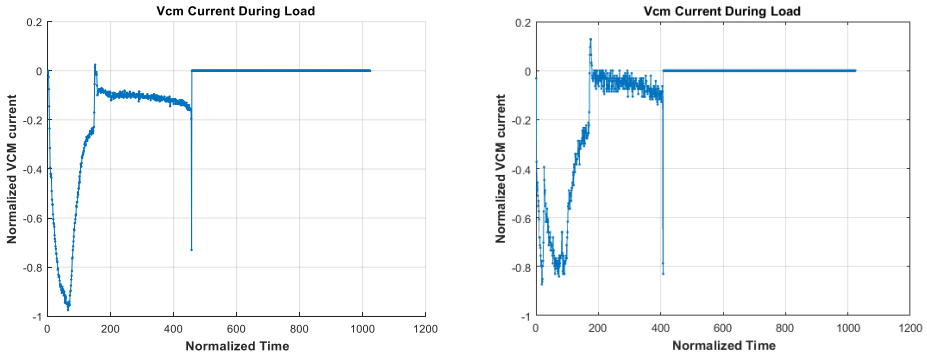
The current that supply to the voice coil motor is continuous value but the current that collect for experiment is sampling from continuous value. This sampling time is equal to the interrupt time for close loop feedback control system. In this study, we allocate memory of 1024 values for each load operation. This will include interval time of 1024 time unit which enough for our validation.



(A) (B)

Figure 5 VCM current of missing latch drive (B) Missing Latch position.

From the experiment with the problematic drives it is found that the form of voice coil motor current is change when experiment with incomplete assembly drive. For example, when latch is not installed as shown in Figure 5(A) the shape of the voice coil current is different from Figure 4(A). Figure 5(B) show the position where latch is not installed. Figure 6 show the comparison of VCM current between good drive, Figure 6(A) and bad drive, Figure 6(B). From the figure, the shape of VCM current show big difference in the time unit from 0 to 200 which is the time when heads get off the garage and load onto media. Later on, the difference become small during heads fly onto the media and find crash stop. This is because the missing components of the drive such as latch influence the VCM current only on early time during heads get off from the garage, it holds actuator to prevent actuator unexpectedly load heads onto the disks. After actuator can get off from garage, latch has no longer influence to the VCM current, so the differences are decreased.



(A) (B)

Figure 6 (A) VCM current of good drive. (B) VCM current of bad drive.

2.3 Related Work

There are many works that use machine learning techniques to identify defects, faults and abnormalities in systems. Support Vector Machine is one favorited techniques that have been chosen. The algorithm was developed by C. Cortes and V. Vapnik at AT&T Bell Labs [1] to tackled two-group classification problems by computing the best hyperplane that separate all data points of one class from another class. A. Kumar and R. Kumar propose a method to detect defect from vibration signal of centrifugal pump using time-frequency analysis using support vector machine [2]. M. Ebrahimi, M. Khoshtaghaza, S. Minaei, and B. Jamshidi uses SVM technique to detect pest in strawberry greenhouses [3]. Joseph F. Murray, Gordon F. Hughes and Kenneth Kreutz-Delgado [4] use machine learning method for predicting failure in hard disk drive using Self-Monitoring and Reporting Technology (SMART) data. Many techniques have been used in this study including Support Vector Machines and Multiple Instance Naïve Bayes (mi-NB) and they found that SVM with redial base function kernel gave high detection rate with low false alarm when compare to other methods.

Long Short-Term Memory was proposed by S. Hochreiter and J. Schmidhuber [5] which was designed to avoid long-term dependency problem and solve the issue of gradient explosion or diminish in learning process of RNN. LSTM is among the best in learning sequence and time series data and this technique have been applied in various application. J. Li and Y. Shen [6] use Bi-LSTM to generated images description. A. M. Ertugrul and P. Karagoz [7] used Bi-LSTM to classified movie gene from plot summaries. A. H. Mirza and S. Cosan use sequence LSTM to detect computer network intrusion. Cheng Feng et al [8] use LSTM network and package signature to implement multi-level anomaly detection in Industrial Control System (ICS). Y Wang et all [9] use LSTM to predict the water quality. Y. Heryadi et al [10] experiment with various techniques including Stacked LSTM to recognize transaction amount for fraudulent transaction.

A deep convolution neural network is the widely used for image recognition. Y. Le Cun et all [11] proposed to use the technique for handwritten digit recognition. C. Szegedy, W Liu et [12] proposed a deep convolutional neural network architecture code name Inception to classify and detect image in Imagenet Large-Scale Visual Recognition challenge 2014 (ILSVRC14) and won the prized. I. Haberal and H. Ogul [13] use a deep CNN network to make a prediction of protein metal binding-site.
Wavelet transform can be used as feature extraction technique in data. Q. Zhao and L. Zhang [14] use wavelet transform to extract features from The electrocardiogram (ECG) signals. R. F. Leonarduzzi [15]. T. Li and M.Zhou [16] classified EGC signals using Wavelet Packet Entropy and Random Forests.

Chapter 3
Machine Learning Methods

In this research, we propose three machine learning techniques to classify the good and bad drives from hard disk drive assembly process. These techniques are Bidirectional Long Short-Term Memory, Wavelet transform with Convolutional Neural Network and Support Vector Machine. However, in this section, beside from the propose techniques we mentioned earlier we also introduce the basic knowledge of deep learning and CNN technique as it is the fundamental knowledge to Wavelet transform with Convolutional Neural Network technique that we propose.

3.1 Training and Validation Process in Matlab®

In this study we use Matlab® as our tool for training and classifying machine learning techniques. So, in this section we discuss the fundamental methodology use in machine learning, especially in Deep Learning, and the basic Matlab® command use in this methodology as well as its implementation.

3.1.1 Training Options

TrainingOptions is a Matlab® command for specify training option for training deep neural network such as CNN, LSTM. The Syntax for this command is

```
options = trainingOptions(solverName,Name,Value)
```

SolverName is the solver for training neural network, currently Matlab® provides three solvers which are the stochastic gradient descent with momentum (SGDM), the RMSProp optimizer and the Adam optimizer. In this study, we focus only on one solver which is the stochastic gradient descent with momentum. So, we specify ‘sgdm’ as our solver name in the command argument, name-value pair argument can be used to specify the momentum of this solver. Other two solvers are beyond the scope of our study, we left it as our future work.

Name-Value Pair Arguments specify optional comma-separated pairs of Name,Value arguments. Name is the argument name and Value is the corresponding value. There are a number of name-value that Matlab® provided. Code fragment below show the example on how to specify trainingOptions in Matlab® with SGDM solver name.

The first argument is ‘sgdm’ as we want to us SGDM as our solver and then the next argument is in name-value pair argument format. We input ‘momentum’ and 0.9 which specify that we use momentum equal to 0.9 in this training. If this ‘momentum’ field is omitted, and we choose to use SGDM as our solver then Matlab® will use default value (0.9) for computation. If we use another solver this field is not necessary.

```
options = trainingOptions('sgdm', ...
    'momentum',0.9, ...
    'ExecutionEnvironment','cpu', ...
    'MaxEpochs',100, ...
    'MiniBatchSize',10, ...
    'Shuffle','never', ...
    'Plots','training-progress');
```

ExecutionEnvironment field is used to specify the hardware resource use for training network. In this study, we use CPU as our hardware resource for training network. Malab® also provide GPU option as a resource for training the network but it's required the hardware that support CUDA® enabled NVIDIA® GPU with compute capability 3.0 or higher. Since the computer that we use in this study is not support CUDA® so training with CUP is only our option. In our study, it's take very long time to train Wavelet-CNN with CPU. In the future work of improving Wavelet-CNN we may need to look for the hardware that support CUDA® to accelerate the training.

An epoch is the full pass of the training algorithm over the entire training data set. The maximum number of epochs to use during training can be specified in 'MaxEpochs' field. In our study we set MaxEpochs to 100 as we found that the accuracy and loss of training are steady with this value.

MinibatchSize field specify the size of the mini-batch to use for each training iteration. A mini-batch is a subset of the training set that is used to evaluate the gradient of the loss function and update the weights accordingly. An iteration is one step taken in the gradient descent algorithm towards minimizing the loss function using a mini-batch. In this study, the mini-batch size we use is vary for each experiment groups. Since each experiment group have different training data set. For example, experiment group # 1 use 500 training data set. We use mini-batch size of 10 to accelerate our training, lower sample size can use smaller number of minibatch.

Plot field specify if we want to show the process during training by specified ''training-progress' value the training process pops up as shown in figure 13. The gradient and an update of the network parameters is calculated in every iteration and show in the plot as show in Figure 7

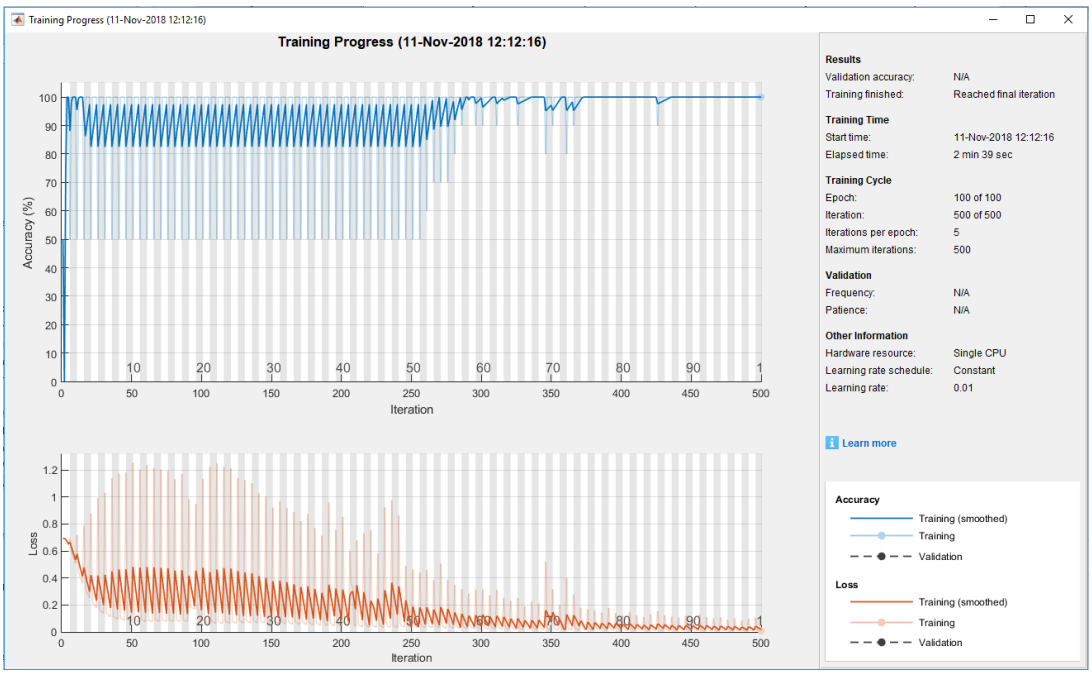


Figure 7 Training Process Window.

3.1.2 Monitoring Training Process

Training accuracy is the accuracy of classification on each individual mini-batch. For 500 training data set with mini-batch size set to 10. The number of mini-batch is 50. This mean that in each epoch the network will iterate 50 times thought each mini-batch. For 100 number of epochs, the total iteration for training is 5000.

Training loss is the loss on each mini-batch. The loss function that use for this classification is the cross entropy loss which used for softmax outputs or classification task as show in Equation (1).

$$L = -\sum_n y_n log q_n(x, \theta) \quad (1)$$

Where y_n is 1 if data x class n and 0 otherwise.

The training process is also shown in Matlab® command windows as depicted in Figure 8.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:00	50.00%	0.6934	0.0100
10	50	00:00:12	100.00%	0.0764	0.0100
20	100	00:00:23	100.00%	0.0622	0.0100
30	150	00:00:35	100.00%	0.0378	0.0100
40	200	00:00:46	100.00%	0.0234	0.0100
50	250	00:00:59	100.00%	0.0174	0.0100
60	300	00:01:10	100.00%	0.0126	0.0100
70	350	00:01:22	100.00%	0.0091	0.0100
80	400	00:01:34	100.00%	0.0066	0.0100
90	450	00:01:45	100.00%	0.0050	0.0100
100	500	00:01:57	100.00%	0.0041	0.0100

Figure 8 Training Process in Matlab® command window.

3.1.3 Train the Network

After the network option training option is set, we can start training the network by invoking `trainNetwork` Matlab® command. This command can be used to train convolutional neural network, a long short-term memory (LSTM) network and a bidirectional long short-term memory (Bi-LSTM) for deep learning classification.

```
net = trainNetwork(XTrain,YTrain,layers,Trainingoptions)
```

where XTrain is the training input data. The training input data can be either the input of images if we use to train CNN network or the sequence of time series input if we use to train LSTM or Bi-LSTM. YTrain is the categorical label that corresponds to the XTrain data. TrainingOptions is the training option that we have discussed in preceding section. The layer parameter is the layer of the network. For example, CNN layer that we discuss in section 3.2 can be specified as.

```
layers = [
    imageInputLayer([224 224 3])
    convolution2dLayer(5,20,'Padding',1)
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(5,40,'Padding',1)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer];
```

3.1.4 Classify the Network

After the network have been trained, we then can use the model to classify class on our testing data set using this Matlab® command.

```
[YPred,scores] = classify(net,XTest)
```

Where ‘net’ is model or the trained network that we get from training and ‘XTest’ is our testing data set. ‘YPred’ is the classification output of the XTest and ‘scores’ is the probability scores of each classification output. ‘YTest’ is our actual class of testing data. From the Performance measurement that we discus in section 4.4, we can calculate accuracy, recall, precision and F-measure with Matlab®. As show in code fragment below.

```
TP = 0; % True Possitive
TN = 0; % True Negative
FP = 0; % False Possitive
FN = 0; % Fasle Negative

for i = 1:size(YPred,1)
    if (YPred(i) == categorical(0) && YPred(i) == YTest(i))
        TP = TP + 1;
    end

    if (YPred(i) == categorical(1) && YPred(i) == YTest(i))
    TN = TN + 1;
    end

    if (YPred(i) == categorical(0) && YPred(i) ~= YTest(i))
        FP = FP + 1;
    end

    if (YPred(i) == categorical(1) && YPred(i) ~= YTest(i))
        FN = FN + 1;
    end
end

% Accuracy
Accuracy = (TP + TN)/(TP+TN+FP+FN);

% Precision
Precision = TP/(TP+FP);

% Recall ,Sensitivity
Recall = TP/(TP+FN);

% F-Measure
FMeasure = 2*TP/(2*TP+FP+FN);
```

3.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm, a type of machine learning algorithms that is most popular for classifying image, video, text and sounds. The technique is very useful in finding pattern in image to recognize the objects, screens and faces by directly learn from the image data and using pattern to classify images with no need of manual feature extraction.

The network architecture of CNN compound of many network layers. Starting from the input data from the input layer pass through many layers to classification output layer. There are many variations of the CNN architectures depending on the types and number of layers included which depends on the particular application or

data. For example, if the output is categorical, the network must have a Softmax layer and classification layer whereas if the output is continuous the network must have regression layer at the end of the network. Figure 9 show the basic of CNN network architecture that use in Matlab®.

A number of network layer also depends on the quality and number of images that are used for classifying. For example, a small number of gray image might be sufficient for learnt and classified by a small network with one or two convolutional layers. On the other hand, we might need a more complicated network with multiple convolutional and fully connected layers for more complex data with millions of colored images.

Image Input Layer inputs the image into the network and applied data normalization by using input size argument. For example, in our study we input the image into the form of 224-by-224-by-3 which mean 224 in height, 224 in width and number of channel for color image is 3.

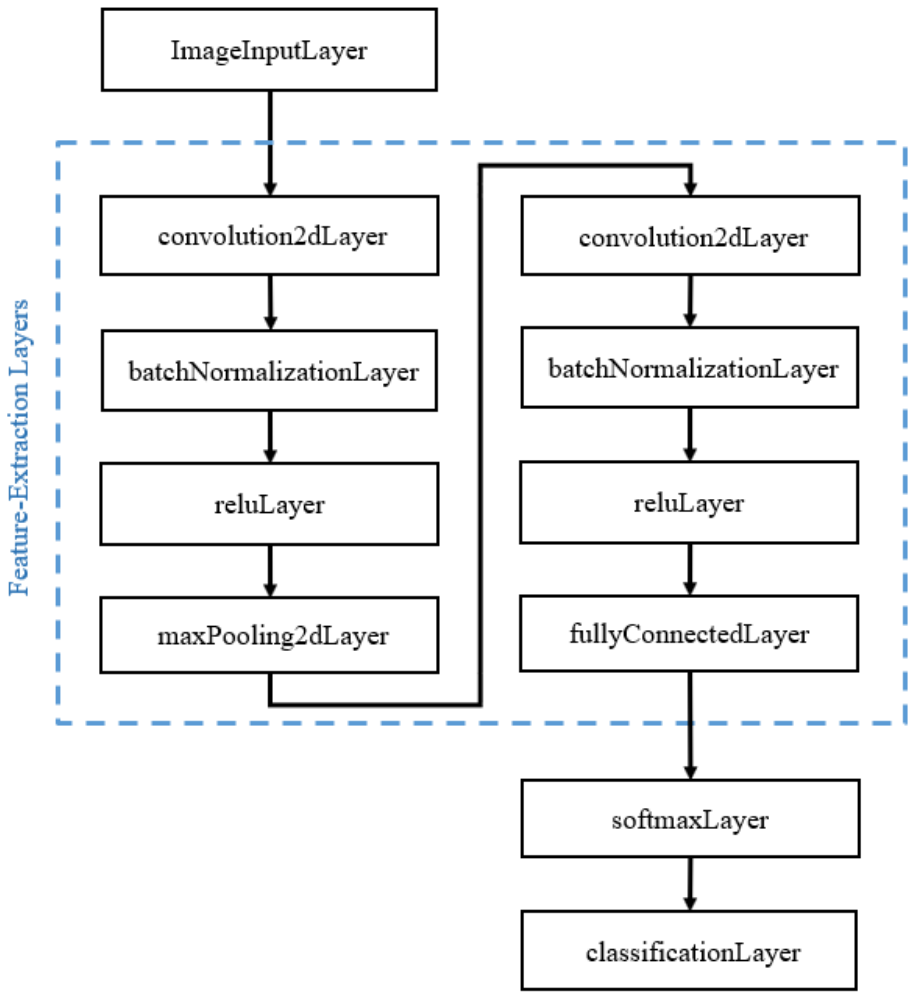


Figure 9 Basic CNN Network Architecture.

In Matlab ®, A 2-D convolutional layer applies sliding convolutional filters to the input image. This layer consists various components such as filter, stride, feature map etc. which can be configured by using convolution2dLayer function.

Filter is a subregions of the input image where the network uses this filter to learn the feature localized by these subregions by moving throughout an images in vertical and horizontal direction and compute a dot product of weight and the input.

Feature map is the output from the filter that use different set of weight and bias, Therefor, the number of feature map is equal to the number of filter. Convolution of an image with different feature map can perform operation such as blur, edge detection and sharpen the image.

Batch Normalization Layer is designed to speed up the training of convolutional neural networks and reduce the sensitivity to the network initialization by normalize each input channel across a mini-batch. This layer should put between the convolutional layers and non-linearities layer.

ReLU Layer is a nonlinearity activation function layer that perform threshold operation to each element of the input where any input less than zero is set to zero

Max Pooling Layers is designed to reduce the number of parameters when the images are too large by dividing the input into rectangular pooling regions and computing the maximum of each region whereas Average Pooling Layers is down-sampling by computing the average value of each region.

Fully Connected Layer computes class scores that we'll use as output of the network by multiplies the input with a weight matrix and then adds a bias vector. This layer summarizes all of the features learned by the previous layers across image to identify the important patterns. So all neurons in a fully connected layer connect to all the neuron in the previous layer. the last fully connected layer combines the features to classify the images.

A softmax layer applied softmax function which calculate the probability of each output from input from previous layer, usually from fully connected layer, and then pass it to classification layer for final classification.

3.3 Bidirectional LSTM (Bi-LSTM)

Long Short-Term Memory (LSTM) network is a special kind of Recurrent Neural Network (RNN) which can learn long-term dependency between each step input of sequence data and it is designed to avoid long-term dependency problem and solve the issue of gradient explosion or diminish due to long time lags during backpropagated error in learning process of RNN. So, Bi-LSTM is well-suited to classify and predict long-time series data.

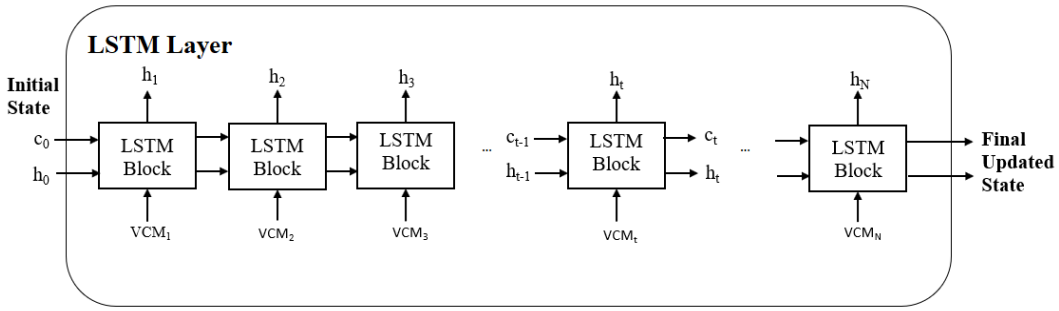


Figure 10 LSTM Layer.

In this study, we use Bidirectional LSTM (Bi-LSTM) which is the variation of LSTM network that can learn both forward information and backward information of VCM current data at any particular time. For the forward pass, the VCM current input is fed into LSTM network of *N* hidden units as show in Figure 10, where h_t and c_t represents the output and cell state of LSTM cell at time *t* respectively. The backward pass is similar to forward pass but in opposite direction. To predict the class label, the network end with fully connected layer, a softmax layer and classification output layer as depicted in Figure 11.

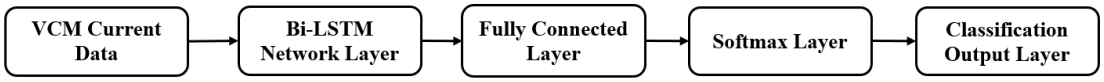


Figure 11 LSTM Network Architecture

In LSTM, each of the layer consists of the *hidden* state and cell state. The hidden state at time step *t* keeps the output of the LSTM layer for this time step. The cell state keeps information learned from the previous time steps. At each time step, the layer adds or removes information from the cell state, where gate is used as the layer controls these updates. Figure 12 show the flow of data at each time step *t*, where *i*, *f*, *g* and *o* represent Input gate, Forget gate, Cell candidate gate and Output gate respectively.

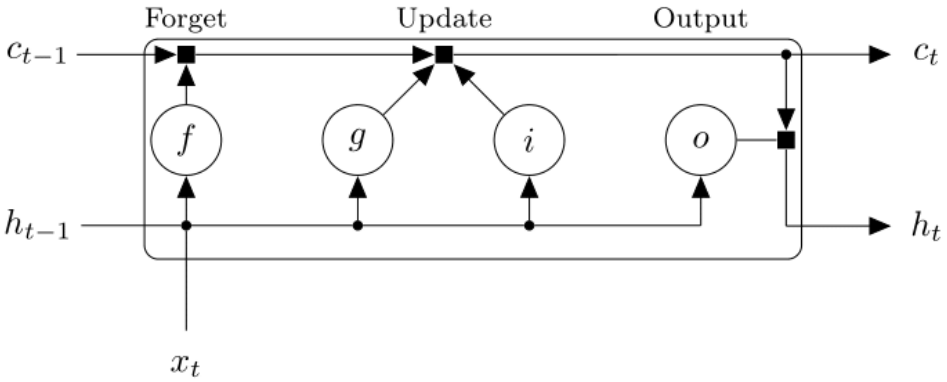


Figure 12 LSTM cell state [17].

The Matlab® command use to specify the Bi-LSTM layer and the training options is shown in code fragment below.

```
maxEpochs = 100;
miniBatchSize = 1;
numHiddenUnits = 100;

layers = [ ...
    sequenceInputLayer(inputSize)
    bilstmLayer(numHiddenUnits,'OutputMode','last')
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer]

options = trainingOptions('sgdm', ...
    'ExecutionEnvironment','cpu', ...
    'GradientThreshold',1, ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'SequenceLength','longest', ...
    'Shuffle','never', ...
    'Verbose',0, ...
    'Plots','training-progress');
```

In this study, The Bi-LSTM network use stochastic gradient descent with momentum during training with the momentum value of 0.9 and initial learning rate of 0.01.

Since the number of training samples are vary for each experiment groups. The first experiment group which handle only 50 training sample can be train individually. However, for the larger training sample we need to group some sample as mini batch to be a subset of the training set to use in each iteration. We choose the number of maximum epoch to100 as the loss for this value is close to zero, to prevent underfitting model.

Increasing the maximum epoch more than 100 may cause the problem of overfitting. One epoch is a full pass of the training algorithm over the entire training.

3.4 Wavelet Transform with Convolutional Neural Network

In this study we apply a deep convolutional neural network (CNN) GoogLeNet, a pretrained network for image recognition to classify VCM current data of good and bad drives base on a time-frequency representation using the continuous wavelet transform (CWT), we shortly call this method as Wavelet-CNN

3.4.1 Wavelet-CNN Network Architecture

In this study we apply a deep convolutional neural network (CNN) GoogLeNet, a pretrained network for image recognition to classify VCM current data of good and bad drives base on a time-frequency representation using the continuous wavelet transform (CWT), we shortly call this method as Wavelet-CNN

GoogLeNet is a deep convolutional neural network designed for classify images of 1,000 categories. In this study we leverage this CNN to classify the abnormality of the hard disk drive assembly process based on scalogram images from CWT of VCM current data.

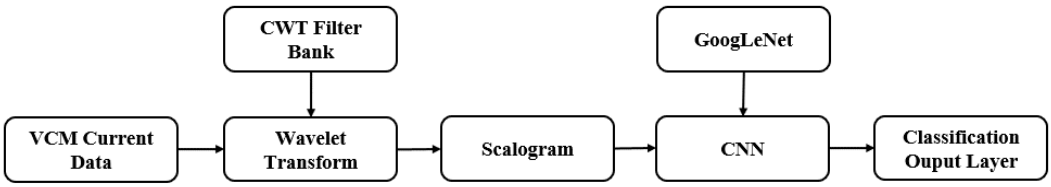


Figure 13 Wavelet-CNN Network Architecture

3.4.2 Continuous Wavelet Transform.

CWT is a time-frequency transformation that is widely use in image compression because it is provides significant improvements in picture quality at higher compression ratios over conventional techniques. It is also use in acoustics processing and pattern recognition because it good to detect abrupt changes in the signal.

The CWT is a complex-valued function of scale and position. If the signal is real-valued and by comparing the signal to the mother wavelet at various scales and positions i.e. for a scale parameter a>0, position, b and the analyzing function wavelet, ψ. the CWT of signal x(t) is shown in Equation (2), where * denotes the complex conjugate.

$$C\big(a,b;x(t),\psi(t)\big) = \int_{-\infty}^{\infty} x(t)\,\frac{1}{a}\psi * \left(\frac{t-b}{a}\right)dt \qquad (2)$$

There are many popular wavelet functions use in signal processing such as analytic Morse wavelet, analytic Morlet wavelet and Bump wavelet. Each wavelet function has different characteristic that can be employed depending on signal features that we want to detect. For abrupt discontinuities signal we can use one

wavelet to detect. Figure 14 show different characteristic between Morse and Bump wavelet.

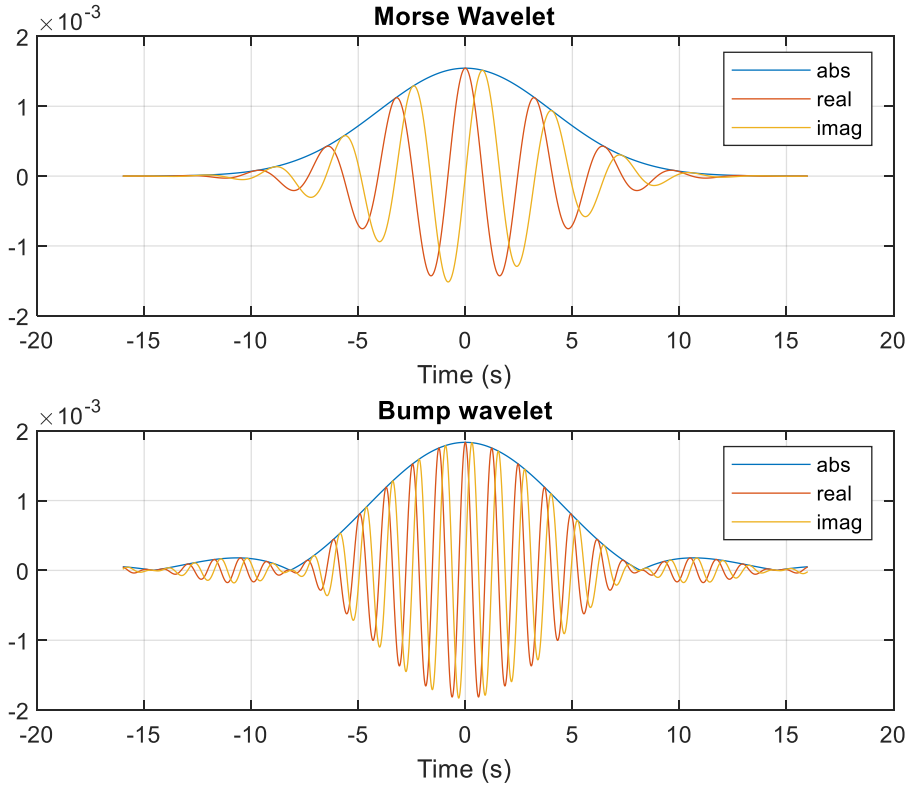


Figure 14 Characteristic comparison between Morse and Bump wavelet.

To create time-frequency representations of VCM current, we need to transform the data into representations call “scalogram”. A scalogram is the absolute value of the CWT coefficients of the data which can be precompute by a CWT filter bank. Computing CWT using CWT filter bank is the good way to improve computational efficiency when analyzing multiple signal in time-frequency.

After create CWT filter bank with 1024 sample. We use the filter bank to take the CWT of the first 1024 samples of VCM current data and obtain the scalogram from the coefficients.

Different filter bank gives different frequency response and resulting in different color in scalogram as shown in Figure 15 where Figure 15A show the frequency response and scalogram using Morse wavelet and Figure 15B show the frequency response and scalogram using bump wavelet.

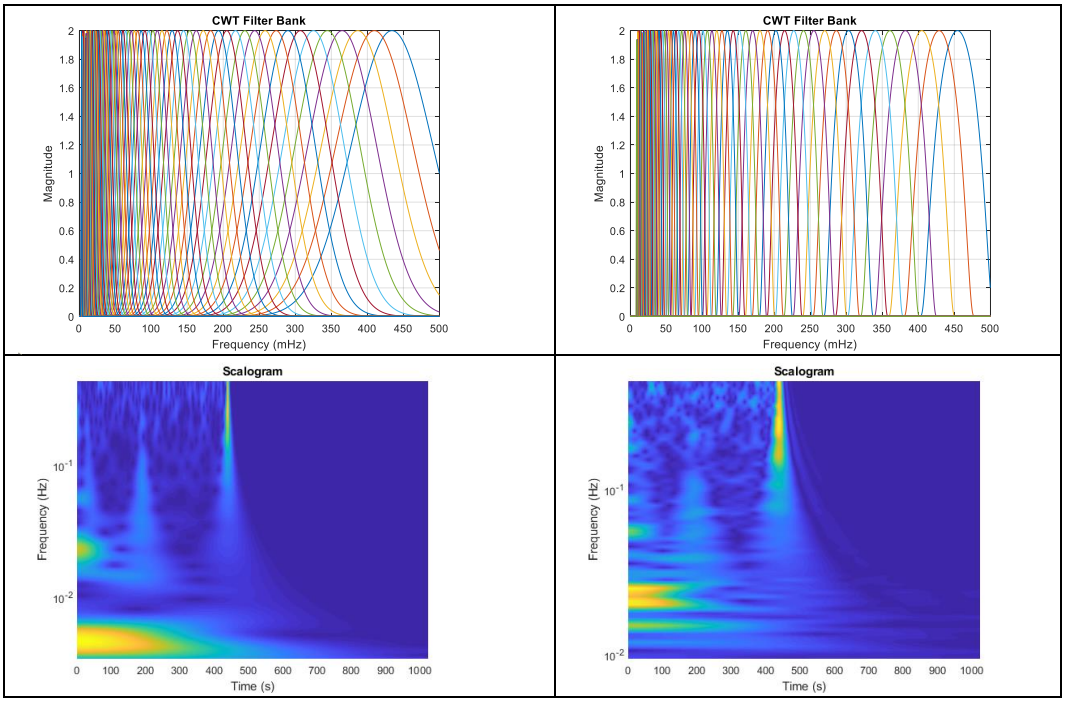
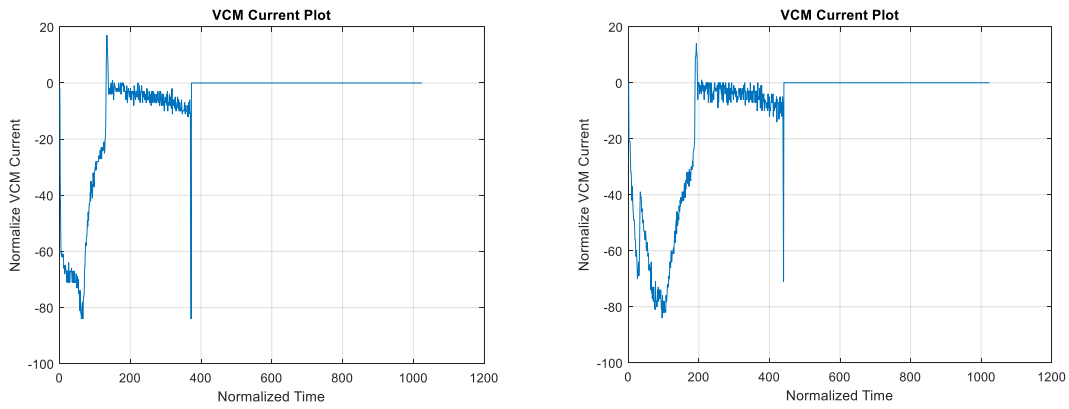


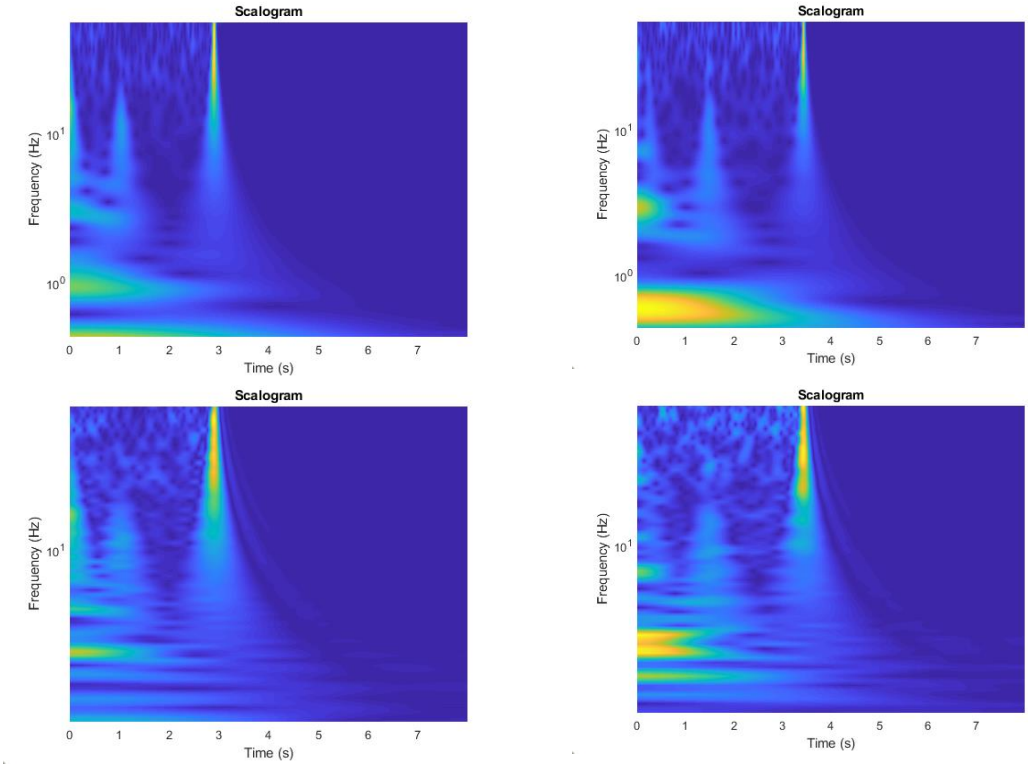
Figure 15 Comparison of scalogram that is derived from (A) Morse wavelet filter bank and (B) Bump wavelet filter bank.

(A) (B)

The comparison of scalogram between Morse wavelet and bump wavelet is shown in Figure 16 where the top of Figure 16(A) show the VCM current data of good drive the middle figure show the scalogram that was generated with Morse wavelet and the bottom figure show the scalogram that was generated with bump wavelet. Figure 16(B) show the behavior of the bad drive according.

From the figure 16, using the Morse wavelet give more distinctive color on the scalogram than Bump wavelet. So in this study, we use Morse wavelet as our mothor wavelet function to generate scalogram to input to a deep convolutional neural network (CNN) GoogLeNet.





(A) (B)

Figure 16 Comparison of scalogram of (A) good drive and (B) bad drive that was generated from different wavelet function.

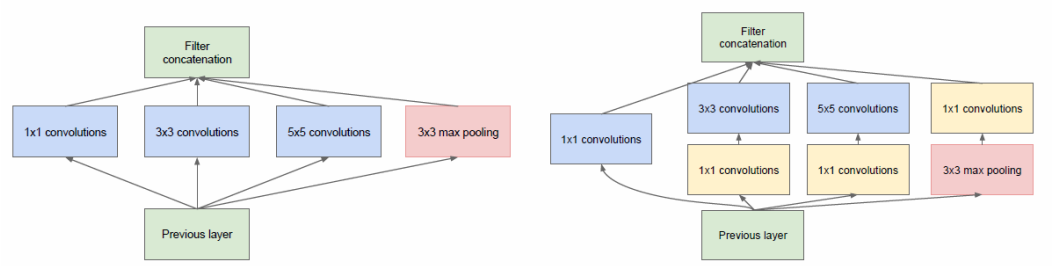
In Matlab®, we can generate scalogram from VCM current data with code fragment below. Where Fs is the sampling frequency which in this case is 1 because the VCM data is sampled every 1-time unit in 1024 data length (DataLenght).

First, the filter bank (fb) was generate using *cwtfilterbank()* function and then the filter bank is pass through *wt()* function to generate CWT with filter bank. The scalogram was plot using pcolor function with time, frequency and the absolute value of CWT coefficient. To be compatible with the GoogLeNet architecture the images were converted to RGB of size 224-by-224-by-3.

```
%Create Time-Frequency Representations
Fs = 1;
DataLength = 1024;
fb = cwtfilterbank('SignalLength',DataLength,...
    'SamplingFrequency',Fs,...
    'VoicesPerOctave',12);
sig = VCMTrainData.Data(1,1:DataLength);
[cfs,frq] = wt(fb,sig);
    t = (0:DataLength - 1)/Fs;
figure;
pcolor(t,frq,abs(cfs))
set(gca,'yscale','log');shading interp;axis tight;
title('Scalogram');xlabel('Time (s)');ylabel('Frequency (Hz)')
```

3.4.3 GoogLeNet

The main architecture in GoogLeNet is called Inception where it concept is to cover a big picture concept but still preserve to keep a detail of small area on image as well. So it convolve data from previous layer in parallel with many different sizes of convolutional filter from the most accurate detailing (1x1) to a bigger one (5x5) as shown in Figure 17,where Figure 17(A) show the naïve version of Incpetion Module and figure 17(B) show the Inception module with dimension reduction.



(A) (B)

Figure 17 GoogLeNet Inception Module (A) naïve version (B) dimension reductions version[12].

In Matlab®, we can load the pretrained GoogLeNet using code fragment below. This requires Depp Learning Toolbox™ Model and need to install GoogLeNet Network support packages from Matlab® Add-On Explore. After execution, the code generate the GoogLeNet Layer graph as shown in Figure 18.

```
net = googlenet;
lgraph = layerGraph(net);
numberOfLayers = numel(lgraph.Layers);
figure('Units','normalized','Position',[0.1 0.1 0.8 0.8]);
plot(lgraph)
title(['GoogLeNet Layer Graph: ',num2str(numberOfLayers),' Layers']);
```

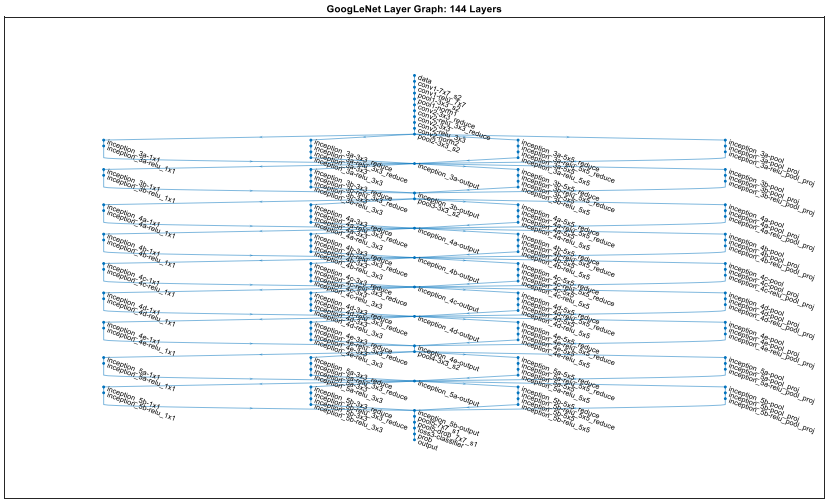


Figure 18 GoogLeNet Layer Graph.

In GoogLeNet Network layer, the earlier layers handle more common feature of images such as edge and color. Since these layers are common for all images so we don't need to change it. However, we need to modify the later layer of the network which focus more on specific feature to well-suited our needs.

We modified the dropout layer 'pool5-drop_7x7_s1' which randomly set input elements to zero with a given probability to help prevent overfitting of training data. The default probability value of this layer is 0.5 we change it to 0.6. We set the fully connected layer from default value of 1000 (1000 categories of images) to 2 which equal to our classification lass (Good and Bad Class). The code fragment for doing this task is shown below.

```
lgraph = removeLayers(lgraph,{'pool5-drop_7x7_s1','loss3-
classifier','prob','output'});

numClasses = numel(categories(allTrainImages.Labels));
newLayers = [
    dropoutLayer(0.6,'Name','newDropout')
    fullyConnectedLayer(numClasses,'Name','fc',...
        'WeightLearnRateFactor',5,'BiasLearnRateFactor',5)
    softmaxLayer('Name','softmax')
    classificationLayer('Name','classoutput')];
lgraph = addLayers(lgraph,newLayers);

lgraph = connectLayers(lgraph,'pool5-7x7_s1','newDropout');
inputSize = net.Layers(1).InputSize;
```

During training, stochastic gradient descent algorithm is used. In each iteration the gradient of loss function is evaluated, and the weights are updated. We set our initial learning rate to 0.0001 and momentum to 0.09. The full training option is shown in code fragment below. Some layer of GoogLeNet network is shown in Table 1.

```
options = trainingOptions('sgdm',...
    'MiniBatchSize',10,...
    'MaxEpochs',100,...
    'InitialLearnRate',1e-4,...
    'Verbose',1,...
    'ExecutionEnvironment','cpu',...
    'Plots','training-progress');

trainedGN = trainNetwork(allTrainImages,lgraph,options);
```

Table 1 GoogLeNet Network Layer.

Layer #	Lyaer Name	Layer Type	Layer Description
1	'data'	Image Input	224x224x3 images with 'zerocenter' normalization
2	'conv1-7x7_s2'	Convolution	64 7x7x3 convolutions with stride [2 2] and padding [3 3 3 3]
3	'conv1-relu_7x7'	ReLU	ReLU
4	'pool1-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 1 0 1]
5	'pool1-norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
6	'conv2-3x3_reduce'	Convolution	64 1x1x64 convolutions with stride [1 1] and padding [0 0 0 0]
7	'conv2-relu_3x3_reduce'	ReLU	ReLU
8	'conv2-3x3'	Convolution	192 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
9	'conv2-relu_3x3'	ReLU	ReLU
10	'conv2-norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
11	'pool2-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 1 0 1]
…			
137	'inception_5b-pool_proj'	Convolution	128 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]
138	'inception_5b-relu_pool_proj'	ReLU	ReLU
139	'inception_5b-output'	Depth concatenation	Depth concatenation of 4 inputs
140	'pool5-7x7_s1'	Average Pooling	7x7 average pooling with stride [1 1] and padding [0 0

			0 0]
141	'newDropout'	Dropout	60% dropout
142	'fc'	Fully Connected	2 fully connected layer
143	'softmax'	Softmax	softmax
144	'classoutput'	Classification Output	crossentropyex with classes '0' and '1'

Event each layer of a CNN produces an activation to an input image but only few layer that are suitable for image feature extraction for example the layer at the beginning of the network that capture the edges and blobs of the image. We can use Matlab® command below to visualize the network filter weights from the first convolutional layer. The result is shown in Figure 19.

```
wghts = trainedGN.Layers(2).Weights;
wghts = rescale(wghts);
wghts = imresize(wghts,5);
figure
montage(wghts)
title('First Convolutional Layer Weights')
```

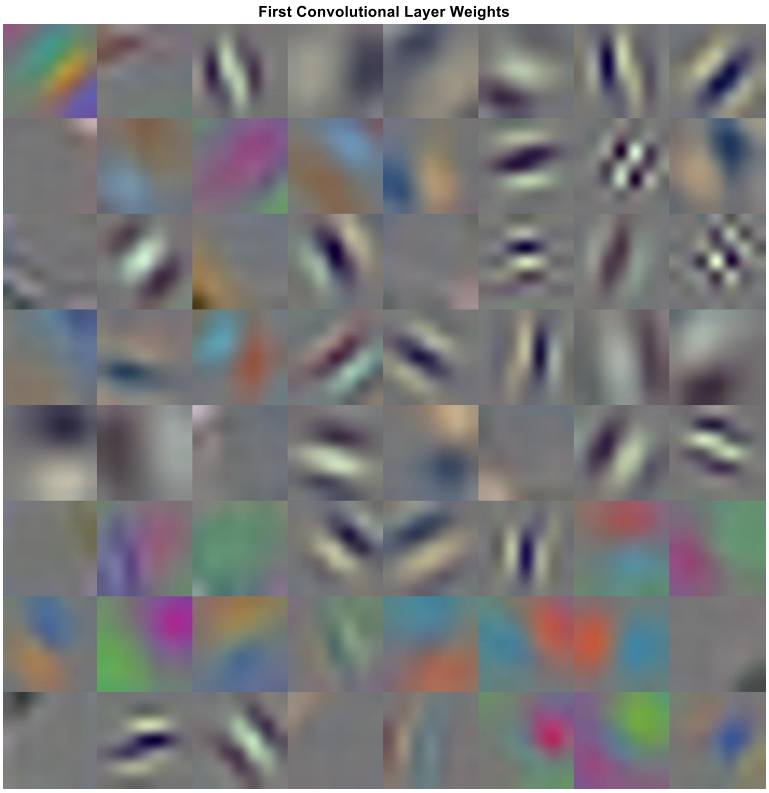


Figure 19 Visualization of the filter weights from the 1st convolutional layer.

Since the number of training samples are vary for each experiment groups. The first experiment group which handle only 50 training sample can be train

individually. However, for the larger training sample we need to group some sample as mini batch to be a subset of the training set to use in each iteration. We choose the number of maximum epoch to100as the loss for this value is close to zero, to prevent underfitting model.

Increasing the maximum epoch more than 100 may cause the problem of overfitting. One epoch is a full pass of the training algorithm over the entire training set.

3.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a pattern classification algorithm which can be used for classification and regression. SVM classifieds data by calculating the optimal hyperplanes that separate all data points of one class from another class. The optimal hyperplane for SVM is the one that have largest margin between the two classes, which is the maximum width of the parallel line in the hyperplane that has no interior data points as depicted in Figure 20, which show the classification of Good and Bad class using SVM.

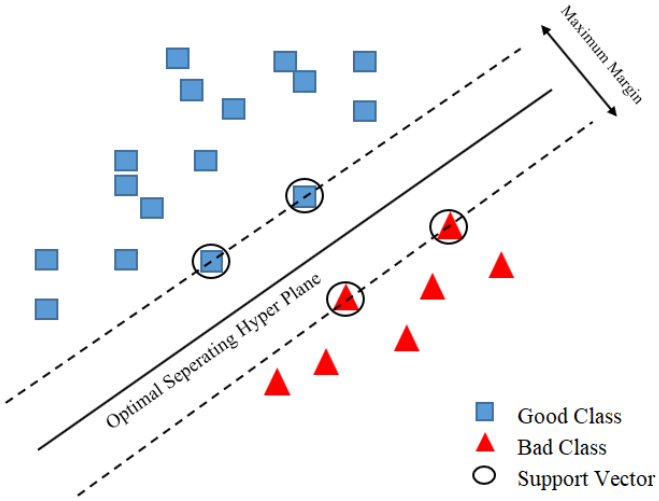


Figure 20 Support Vector Machine for classification.

If the data is linearly separable then the optimal separating hyper plane is exist and can be described as in Equation (3) where N is the number of training data of data point (x_i, y_i) , x_i is the input data and $y_i \in \{+1, -1\}$ corresponds to its target value of the output class, i.e. Good class equal to +1 and Bad class is equal = -1. The parameter w and b is the weight and bias of the SVM respectively.

$$f(x) = w^T x + b = \sum_{i=1}^{N} w_i x_i + b = 0 \quad (3)$$

$$f(x) = w^T x + b > 0 \; on \; Good \; Class(+1) \quad (4)$$

$$f(x) = w^T x + b < 0 \; on \; Bad \; Class(-1) \quad (5)$$

$$y_i(w^T x + b) - 1 \geq 0 \quad (6)$$

The margin can be calculated by equation X, where M is the largest margin.

$$w^T x^{+1} + b = 1$$
$$w^T x^{-1} + b = -1$$
$$w^T (x^{+1} - x^{-1}) = 2$$
$$M = \left(\frac{w}{||w||}\right)^T (x^{+1} - x^{-1})$$
$$M = \frac{2}{||w||} \quad (7)$$

In case that the data is non-linear in lower dimensional space it can be transform to higher dimensional space using kernel trick, which is incorporate kernel function for computing dot product of mapping function. The kernel function is $K(x_i, x) = x_i^T x$ for Linear Kernel $K(x_i, x) = (x_i^T x + 1)^d$ for Polynomial Kernel and $K(x_i, x) = e^{(-\gamma||x - x_i||^2)}$ for Radial basis function Kernel (RBF).

In Matlab®, SVM can be trained using *svmtrain()* Matlab® command. Where 'Training' is matrix of training data and 'Group' specifies the group of the corresponding row of training data with its actual label. In this study, group have only two class label which is Good class and Bad class. Name-Value pair argument can use to specify training option for SVM.

```
SVMStruct = svmtrain(Training,Group,Name,Value)
```

Kernel function for training SVM can be specified in name-value pair argument using name field 'kernel_function'. Matlab® provide several kernel functions such as Linear kernel, which is set as the default value, Quadratic kernel, Polynomial Kernel, Gaussian Radial Basis Function kernel, Multilayer Perceptron kernel and Function handle to a kernel function.

In this study, we have experimented with many kernels and found that Linear Kernel give the best result for this classification. So, we use SVM with Linear Kernel as our technique to compare with other machine techniques that we proposed in this paper. 'autoscale' is a Boolean value specifying whether svmtrain automatically centers the data points at their mean and scales the data points to have unit standard deviation, before training, by default this option is set to true.

The method used to find the separating hyperplane can be specified in 'method' field name where 'QP' value is the Quadratic programming method and 'SMO' is Sequential Minimal Optimization.

The SVMStruct output argument containing information about the trained SVM classifier as shown in Figure 21.

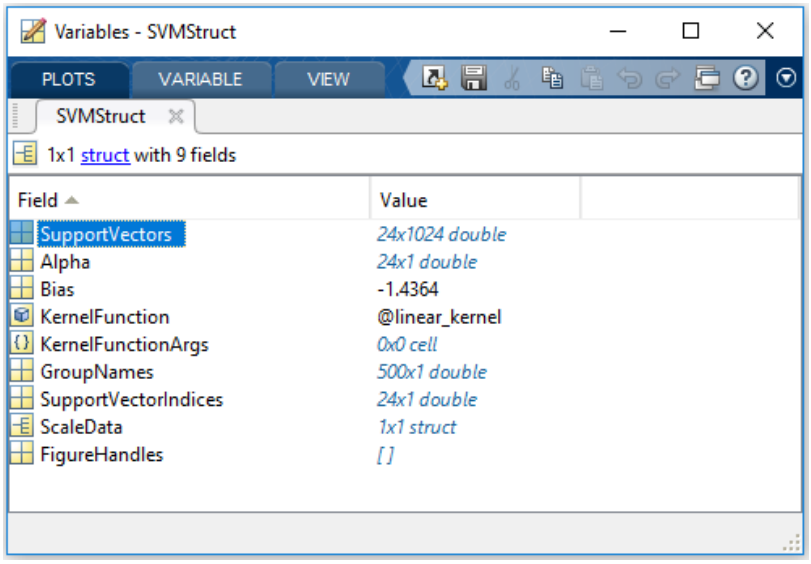


Figure 21 SVMStruct Variables from training SVM.

The 'SupportVectors' variable is the matrix of data points in which with each row corresponding to a support vector of trained SVM in the normalized data space. This matrix is a subset of the Training input data matrix and is weighted in vector of weight 'Alpha' which show positive sign when support vectors belong to Bad class and show negative sign when support vectors belong to Good class. 'Bias' is the intercept of the hyperplane that separates the two classes in the normalized data space.'KernelFunction' and 'KernelFunctionArgs' variable show the kernel parameters that we specify during training. 'GroupNames' is the class label which each row corresponds to the input data. 'SupportVectorIndices' variable specify the index position of support vector in the matrix of training input. The 'ScaleData' is the variable that show the normalized factor that calculate by SVM during training, 'ScaleData.shift' is the row of vector that each value is the negative of the mean across an observation in the training data and 'ScaleData.scaleFactor' is the row of vector that each value is 1 divided by the standard deviation of an observation in Training, the training data.

With the information from SVMStruct variables, the support vectors for Good and Bad class can be plot separately for better visualization and show in Figure 22.

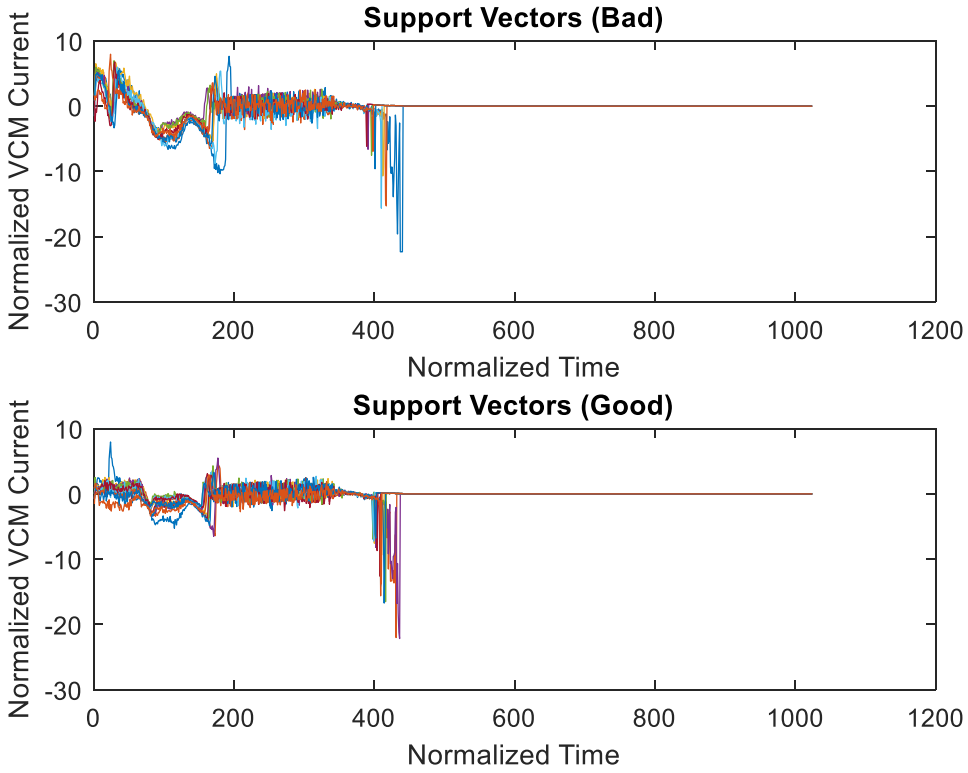


Figure 22 Support Vector of VCM current separate between Good and Bad Class.

The testing data set can be classified with Support Vector Machine using *svmclassify* Matlab® command.

```
Group = svmclassify(SVMStruct,Sample,'Showplot',true)
```

Where the 'SVMStruct' is the output struct from *svmtrain* and 'Sample' is our testing or validation matrix of data. The name-value pair argument of 'Showplot'=>' true' specify the command to plot the figure of Sample matrix data but only data in two-dimensional that can be plot with this option.

Code fragment below show our implementation and performance measurement of SVM. 'XTrainMat' is our VCM current data matrix and 'YTrainMat' is the matrix that each row corresponding to VCM training data. These two matrixes are use as input of svmtrain to train SVM with our training data. After the training, the output SVM Struct is generated and then use as input for svmclassify function for classifying testing data set. After classification, the actual class of testing output is compare with predicted output and then calculate the performance measure.

```
SVMStruct = svmtrain(XTrainMat,YTrainMat,'kernel_function','linear');

Group = svmclassify(SVMStruct,XTestMat);

YPred = categorical(Group);
YTest = categorical(YTestMat);
```

```
TP = 0; % True Possitive
TN = 0; % True Negative
FP = 0; % False Possitive
FN = 0; % Fasle Negative

for i = 1:size(YPred,1)
    if (YPred(i) == categorical(0) && YPred(i) == YTest(i))
        TP = TP + 1;
    end

    if (YPred(i) == categorical(1) && YPred(i) == YTest(i))
    TN = TN + 1;
    end

    if (YPred(i) == categorical(0) && YPred(i) ~= YTest(i))
        FP = FP + 1;
    end

    if (YPred(i) == categorical(1) && YPred(i) ~= YTest(i))
        FN = FN + 1;
    end
end

% Accuracy
Accuracy = (TP + TN)/(TP+TN+FP+FN);

% Precision
Precision = TP/(TP+FP);

% Recall ,Sensitivity
Recall = TP/(TP+FN);

% F-Measure
F-Measure = 2*TP/(2*TP+FP+FN);
```

Chapter 4

Research Method

4.1 Data Collection Process

In the process of collecting data of voice coil motor current to use for training the models. The data is stored in pair format between voice coil motor current data and the class of assembly process. This class indicates that which hard disk drives are assembly correctly. The Good class represents the hard disk drive that assembly correctly and the Bad class represents the hard disk drive that have abnormality in hard disk drive assembly process. We excluded the drives that are completely and properly installed but fail during servo writing process from this study. The voice coil motor current data is sampling every time unit starting when load command is invoked for 1024-time units. This time interval is long enough to cover event from loading head until head loading onto media and found the crash stop, on average this process takes around 300 to 600-time unit as depicted in Figure 4(B).

The voice coil motor current data from many drives are collected in form of Figure 23 in which Figure 23(A) is array that each row collect the data of voice coil motor from 1-time unit to 1024 time unit in Figure 23(B) in each row collect the class of hard disk drive.

	1	2	3	.	.	.	1024
HDD#1 =>							
HDD#2 =>							
HDD#3 =>							

(A)

	Class
HDD#1 =>	Bad
HDD#2 =>	Bad
HDD#3 =>	Good

(B)

Figure 23 (A) array of VCM current. (B) class of hard disk drive.

4.2 Experiment Groups

In this study, the voice coil motor current data have been collected from 7539 drives. In which only 20 abnormal drives (Bad class) is collected, the remaining data set, 7519 is collected from good drive (Good Class). The small ratio of bad class to good class samples (0.27%) introduce imbalance problem during training process. So, we divide the collected data set into three experiment groups and in each group is separate for training and testing data set. Table 2 shows the data set for each experiment group. For each experiment group it has different number of training data set, testing data set and ration of bad drive in training data set. The ratio of bad drive is calculated by the number of bad sample in training data set divide by the total number of training data set. The experiment also plays with varying number of training data set because in practical it is needed to consider how many training data

set should be collected from assembly line to adequately satisfy the training requirement of each machine learning model. Larger data set need more time and resources for collecting while fewer data set may introduce overfitting to the machine learning model. The number training data set for experiment group #1, #2 and #2 is 500, 100 and 50 respectively and the ratio of bad drive in each experiment group is 3%, 10% and 10% respectively.

Table 2 *Data set separated by group of experiment.*

Experiment Group #	Training Data Set		Testing Data Set		Total	Ratio of Bad Drive
	Good	Bad	Good	Bad		
#1	485	15	7034	5	500	3%
#2	90	10	7429	10	100	10%
#3	45	5	7474	15	50	10%

4.3 Training and Testing

All the techniques that we have proposed as discuss in section 3, which are Bi-LSTM, wavelet transform with Convolutional Neural Network and SVM are supervised learning technique which is the learning technique that map input to an output based on example of input and output pairs. So, it is required to separate VCM current data set into two data set, one is training data set which use to train the network model and second is testing data set. The network model that that we get from training data set will be use as model to evaluate testing data set. The classified result from testing data set with the models are compare with real HDD assembly class. The performance metrics are evaluated. The process flow is shown in Fig. 24.

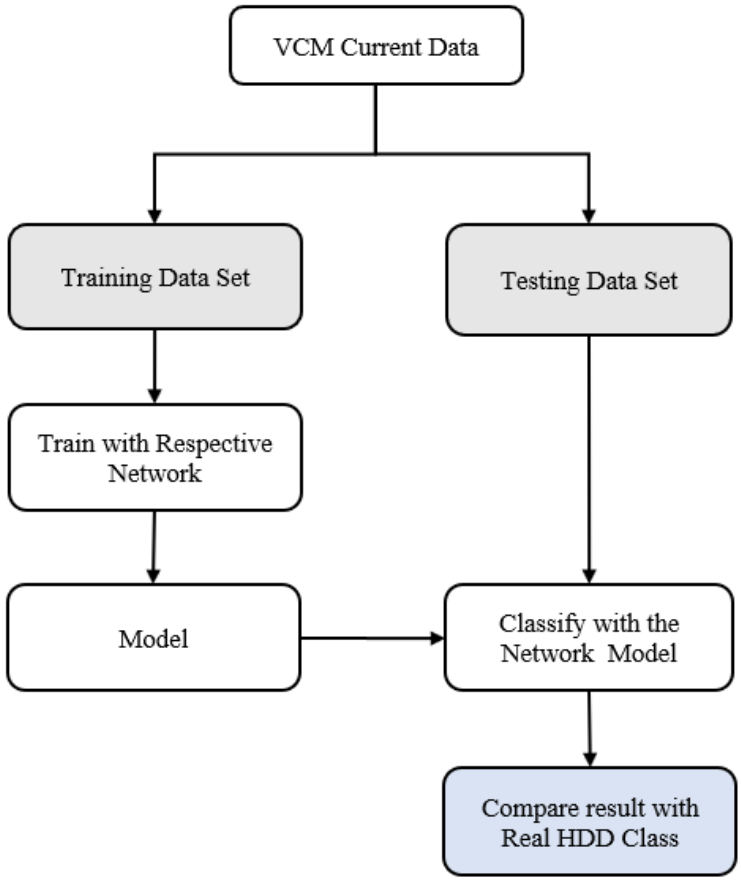


Figure 24 Training and Testing work flow.

4.4 Performance Measurement

The Precision is defined as in Equation (8), where TP (true positives) referred to the number of correctly predicted positive sample (bad drive). FP (false positive) is referred to the number of negative samples (good drives) that are incorrectly predicted as positive sample (bad drive). The Recall is defined as Equation (9), where FN (false negative) is the number of positive samples that are incorrectly predicted as negative. The accuracy is computed as in Equation (10).

In this study, due to the number of negative sample is far more than the number of positive sample which introduce imbalance issue to training process, so the F-measure is evaluated. The F-measure is the harmonic average of precision and recall which be written as Equation (11).

$$\text{Precision} = TP/(TP+FP) \quad (8)$$

$$\text{Recall} = TP/(TP+FN) \quad (9)$$

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN) \quad (10)$$

$$\text{F-Measure} = 2TP/(2TP+FP+FN) \quad (11)$$

Chapter 5

Experiment Result and Discussion

5.1 Experiment Result

This study was evaluated with testing data set as described in section 4. An Accuracy, precision, recall and F-measure were calculated as performance measurement.

The results were evaluated at the end of training epoch. In this study we set the maximum epoch to 100 for Bi-LSTM and Wavelet-CNN network model. However, we observed that training accuracy and training loss were converged earlier than that, around 30-40 epochs. The testing results are shown in Table 3 and its performance measurement are shown in Table 4.

Table 3 Classification result from testing data set.

ML Techniques	Groups	Good	Bad	TP	FP	TN	FN
Bi-LSTM	#1	7034	5	5	0	7034	0
	#2	7429	10	10	0	7424	0
	#3	7474	15	13	0	7474	2
Wavelet-CNN	#1	7034	5	5	4	7030	0
	#2	7429	10	10	15	7414	0
	#3	7474	15	13	15	7459	2
SVM	#1	7034	5	5	0	7034	0
	#2	7429	10	10	5	7424	0
	#3	7474	15	15	3	7471	0

Table 4 Performance measurement matrices.

ML Techniques	Groups	Accuracy	Precision	Recall	F-measure
Bi-LSTM	#1	100.00%	100%	100%	100%
	#2	100.00%	100%	100%	100%
	#3	99.97%	100%	87.0%	93.0%
Wavelet-CNN	#1	99.94%	55.6%	100%	71.4%
	#2	99.80%	40.0%	100%	57.1%
	#3	99.77%	46.4%	86.7%	60.5%
SVM	#1	100.00%	100%	100%	100%
	#2	99.93%	66.7%	100%	80.0%
	#3	99.96%	83.3%	100%	90.9%

5.2 Experiment Discussion

For experiment group #1, which have 7034 negative samples (Good drive) and 5 positive samples (Bad drive). Bi-LSTM based prediction and SVM based prediction show perfect results in matric measurement which resulted in 100% accuracy, 100% precision, 100% recall and 100% F-Measure while Wavelet-CNN based prediction resulted in 99.94% accuracy, 55.6% precision, 100% recall and 71.4% F-measure.

For experiment group#2, which have 7429 negative sample and 10 positive samples. Bi-LSTM based prediction still shown perfect matric measurement performance, 100% accuracy, 100% precision, 100% recall and 100% F-measure while SVM based prediction show a lightly dropped in accuracy, dramatically dropped in precision and F-measure which resulted in 99.93% accuracy, 66.7% precision, 100 % recall and 80% F-measure. Wavelet-CNN shown worsen result than its experiment's result in group#1 which resulted in 99.80% accuracy, 40% precision, 100% recall and 57.1% F-measure.

For experiment group#3, which use totally 50 samples as training data set as described in section 4B. In this experiment group, 7474 negative samples and 15 positive sample were evaluated. None model can archive perfect matric measurement performance. Bi-LSTM based prediction resulted in 99.97% accuracy, 100% precision, 86.7% recall and 93.0% F-measure. Wavelet-CNN resulted in 99.97% accuracy, 46.4% precision, 86.7% recall and 60.5% F-measure. SVM based prediction resulted in 99.96% accuracy, 46.4% precision, 86.7% recall and 60.4% F-measure.

The results for each proposed method are compared across all experiments groups. Fig. 25 show the accuracy evaluation matric compare the across the proposed machine learning methods and experiment groups. The comparison of precision, recall and F-measure are shown in Fig. 26, 27 and 28 respectively.

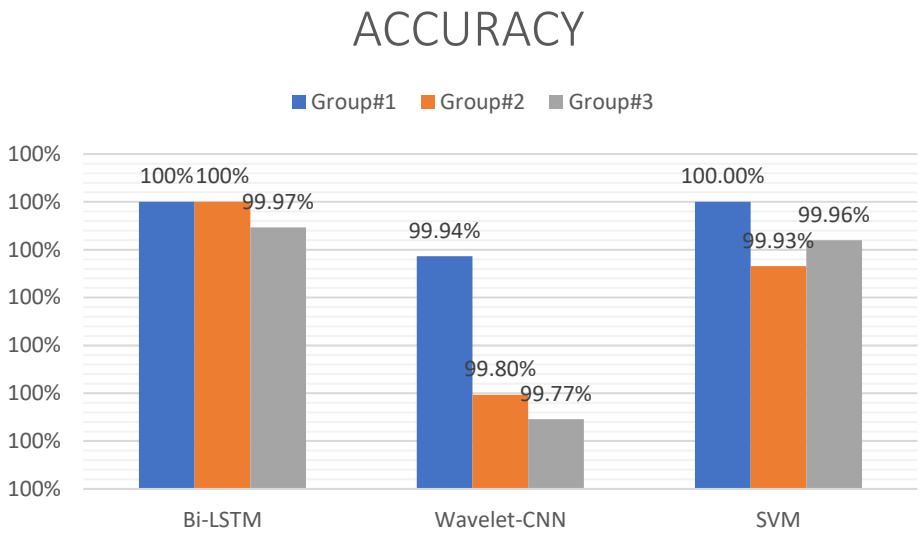


Figure 25 Accuracy evaluation matric using testing data set.

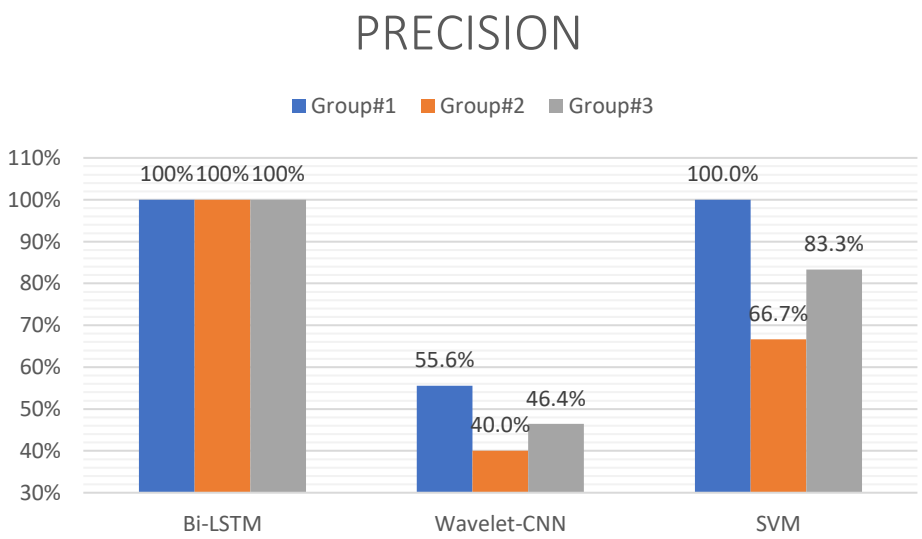


Figure 26 Precision evaluation matric using testing data set.

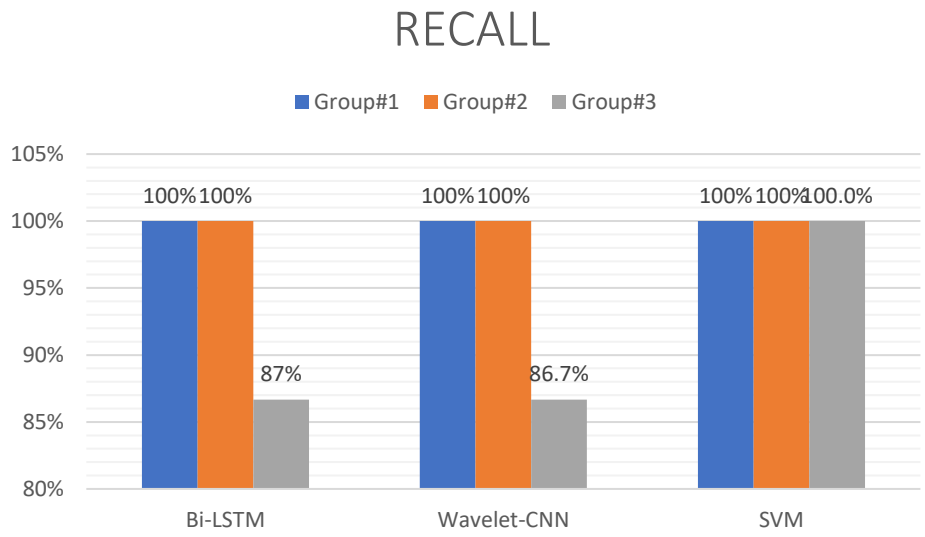


Figure 27 Recall evaluation matric using testing data set.

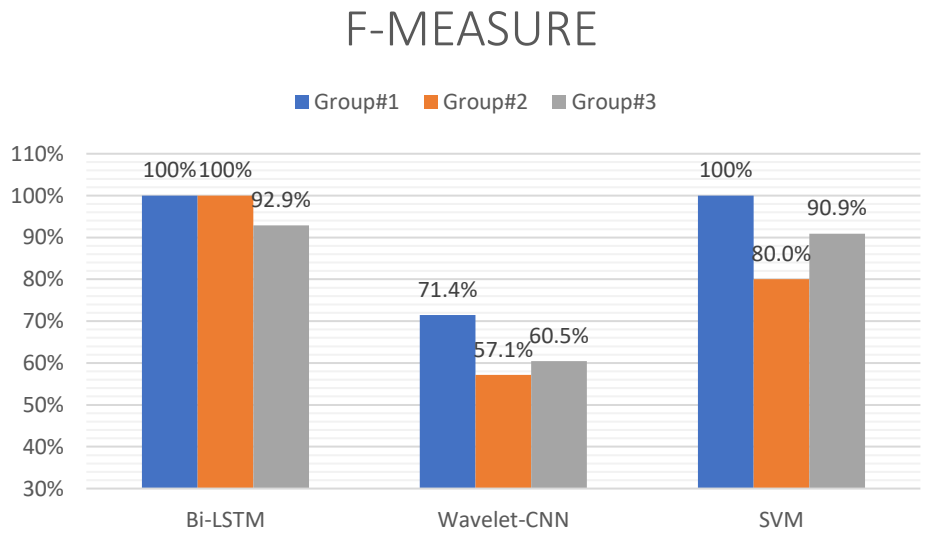


Figure 28 F-Measure evaluation matric using testing data set.

Chapter 6

Conclusion

We presented three methods to detect abnormality in hard disk drive assembly process which is Bi-LSTM, wavelet-CNN and SVM. From the experiments results, we found that Bi-LSTM and SVM are a powerful approach for detect this abnormality since it's give perfect performance measurement matric (100% accuracy, 100% precision, 100% recall and 100% F-measure) on experiment group #1 and #2.

Even though, there is no any methods that can archive perfect performance measurements on experiment group # 3, Bi-LSTM show highest accuracy (99.97%) over SVM (99.96%) and Wavelet-CNN (99.77%) and Bi-LSTM still resulted perfect score on precision in experiment group #2 while SVM and Wavelet-CNN resulted only 83.3% and 46.4% respectively. However, SVM show better recall result than Bi-LSTM on this experiment group, 100% over 87%.

In practical application of this detection in hard disk drive manufacturing process. If the number of training samples are difficult to collected as in experiment group # 3, one need to trade-off between two consequences. One is overkilled the good drive (fail FP drives) and second is letting the bad drives pass through the customer (not fail FN drive).

From the experiment group #3, the FP rate that classify by SVM is 3/7489 which is around 400 drives in a million of manufacturing drives. This mean that 400 drives would be over-killed if we employ SVM in experiment group #3 (This value is 0 when classify with Bi-LSTM). On the other hand, if we use Bi-LSTM model to classify, the recall with this model is 86.7% which mean if there is 100 of abnormality drives in a million of manufacturing drives. There will be 14 drives pass the abnormality detection (This value is 0 when classify with Bi-SVM).

From our discussion above, we encourage to use training sample size at least 100 samples with 10% of positive drive in it to perfectly classify the good and bad drive with Bi-LSTM and SVM model.

The advantage that SVM have over Bi-LSTM is the time to train the model. It took around 3 hours to train Bi-LSTM model with 500 samples size, but it took less than 5 minutes to train the same sample with SVM model.

The result from Wavelet-CNN show worsen than other model in performance measurement matrices, this maybe because we not directly training the model from scratch, but we leverage the existing GoogLeNet as our pretrained model for CNN. GoogLeNet is good in classify general images like animals, flower etc. because it is trained based on these images but not well-suited to classify the images in our study.

REFERENCES

1. Cortes, C. and V.J.M.L. Vapnik, *Support-Vector Networks.* 1995. **20**(3): p. 273-297.
2. Kumar, A. and R. Kumar, *Time-frequency analysis and support vector machine in automatic detection of defect from vibration signal of centrifugal pump.* Measurement, 2017. **108**: p. 119-133.
3. Ebrahimi, M.A., et al., *Vision-based pest detection based on SVM classification method.* Computers and Electronics in Agriculture, 2017. **137**: p. 52-58.
4. Murray, J.F., G.F. Hughes, and K. Kreutz-Delgado, *Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application %J J. Mach. Learn. Res.* 2005. **6**: p. 783-816.
5. Hochreiter, S., #252, and r. Schmidhuber, *Long Short-Term Memory %J Neural Comput.* 1997. **9**(8): p. 1735-1780.
6. Li, J. and Y. Shen. *Image describing based on bidirectional LSTM and improved sequence sampling*. in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*2017.
7. Ertugrul, A.M. and P. Karagoz. *Movie Genre Classification from Plot Summaries Using Bidirectional LSTM*. in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. 2018.
8. Feng, C., T. Li, and D. Chana. *Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks*. in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2017.
9. Wang, Y., et al. *Water quality prediction method based on LSTM neural network.* in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. 2017.
10. Heryadi, Y. and H.L.H.S. Warnars. *Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM*. in *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*. 2017.
11. Cun, Y.L., et al., *Handwritten digit recognition with a back-propagation network*, in *Advances in neural information processing systems 2*, S.T. David, Editor. 1990, Morgan Kaufmann Publishers Inc. p. 396-404.
12. Szegedy, C., et al. *Going deeper with convolutions*. in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
13. İ, H. and H. Oğul. *DeepMBS: Prediction of Protein Metal Binding-Site Using Deep Learning Networks*. in *2017 Fourth International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*. 2017.
14. Qibin, Z. and Z. Liqing. *ECG Feature Extraction and Classification Using Wavelet Transform and Support Vector Machines*. in *2005 International Conference on Neural Networks and Brain*. 2005.
15. Leonarduzzi, R., G. Schlotthauer, and M.E. Torres, *Wavelet Leader Based Multifractal Analysis of Heart Rate Variability during Myocardial Ischaemia*. Vol. 2010. 2010. 110-3.
16. Li, T. and M. Zhou, *ECG Classification Using Wavelet Packet Entropy and Random Forests.* 2016. **18**(8): p. 285.

17. The MathWorks Inc, *Deep Learning Toolbox User Guide.* 2018.

VITA

NAME	Masayuti Simongyi
DATE OF BIRTH	20 November 1983
INSTITUTIONS ATTENDED	Bachelor of Engineering in Electrical Engineering, Kasetsart Universtiy
HOME ADDRESS	46/275 M.16 T.Khlong Nueng, Khlong Luang District,Pathum Thani Province, 12120, Thailand