

東海大學應用數學系
碩士論文

指導教授：楊智烜

車床振動數據與車刀狀況之關聯性分析

Discrimination on the Status of Lathe Tools
from Vibration Data Analysis

研究生：洪廖憶

中華民國一零八年七月

東海大學

應用數學系

碩士學位口試委員審定書

本系碩士班 洪廖憶 君

所提論文 Discriminating the Status of Lathe Tool from Vibration Data
(車床振動數據與刀具狀況之關聯性分析)

合於碩士班資格水準，業經本委員會評審通過，特此證明。

口試委員：

黃 皇 男

張 友 友

指導教授：

楊 志 煜

系主任：

胡 碧 云

中華民國 一〇八 年 七 月 二十四 日

摘要

本研究目標為利用工具機的振動數據判別車刀的狀況。振動數據來源是安裝在車床主軸上的振動感測器，每秒量測 1,660 筆，單位是 $10^{-3}g$ 。本論文的數據有 A、B 兩組；分別來自不同工序。A 數據共有 14 筆：11 筆為正常車刀，2 筆為車刀磨損，1 筆為切斷刀磨損。B 數據紀錄的是某工序持續重複作 95 次---量測車刀從全新到磨損的連續過程---的振動數據。本論文利用移動標準差作數據前置處理，找出正常車刀與磨損車刀之間的明顯不同特徵，並與其他的常用的訊號分析方式做比較。最後嘗試利用深度學習的模型建立預測車刀狀況的鑑別系統。

Abstract

This article investigates the relationship between the vibration data and the status of lathe tools. The vibration data comes from the sensor mounted on the spindle of the lathe to measure the pressure in $10^{-3}g$ 1660 times per second. There are two sets of data, called data set A and B, respectively. The data set A consists of 14 time series, among which 11 data record the pressure of normal lathe tools and 3 data record the pressure of abnormal lathe tools. The data set B consists of 95 time series, which consecutively record the pressure for the same working process, as the lathe tool turns from normal status into abnormal status. We preprocess the time series by using moving standard deviation then find the characteristics which is capable of discriminating the status lathe tools. We also use other methods such as the short-time-Fourier-transformation, Hilbert-Huang transformation to preprocess the time series for comparison. Finally, we try to establish a LSTM (long short-term memory) deep learning model to predict the moving standard deviation of normal lathe tools according to the past vibration. Our future goal is to adopted a LSTM as a generator in a GAN (generating adversarial network) to automatically discriminating the status of lathe tools.

目錄

摘要.....	i
Abstract.....	iii
目錄.....	iv
圖目錄.....	v
謝誌.....	vi
一、研究背景.....	1
1.1 數據來源：.....	1
1.2 研究動機與目的：.....	3
二、研究方法.....	5
2.1 數據預處理：.....	5
2.2 分析工具：.....	13
2.2.1 Python.....	13
2.2.2 移動標準差.....	14
2.2.3 短時傅立葉轉換(STFT).....	15
2.2.5 深度學習之長短期記憶.....	16
三、研究成果.....	20
3.1 移動標準差.....	20
3.2 短時距傅立葉轉換(<i>Fourier transform</i>).....	24
3.3 深度學習之 LSTM.....	28
四、結論.....	33
4.1 結論：.....	33
4.2 未來研究方向：.....	34
參考文獻.....	35

圖目錄

圖(1)	振動感測器	1
圖(2)	車刀示意圖	1
圖(3)	工具機示意圖	2
圖(4)	工具機實圖	2
圖(5)	六筆 A 數據 x 軸之時間序列	5
圖(6)	六筆 A 數據 y 軸之時間序列	6
圖(7)	六筆 A 數據 z 軸之時間序列	6
圖(8)	五筆 A 數據 x 軸預處理後之時間序列	8
圖(9)	四筆 B 數據 x 軸之時間序列	9
圖(10)	四筆 B 數據 y 軸之時間序列	9
圖(11)	四筆 B 數據 z 軸之時間序列	10
圖(12)	B 數據 x 軸作移動標準差	11
圖(13)	預處理示意圖	11
圖(14)	五筆 B 數據 x 軸預處理後之時間序列	12
圖(15)	RNN 方塊圖	16
圖(16)	RNN 流程圖[6]	17
圖(17)	LSTM 流程圖[6]	18
圖(18)	LSTM 架構圖[4]	18
圖(19)	A 數據 x 軸作移動平均	20
圖(20)	A 數據 x 軸作移動標準差	20
圖(21)	B 數據 x 軸 B10 與 B01 比較圖	22
圖(22)	B 數據 x 軸 65 與 01 比較圖	22
圖(23)	B 數據 x 軸 66 與 01 比較圖	23
圖(24)	B 數據 x 軸 79 與 01 比較圖	23
圖(25)	B 數據 x 軸 80 與 01 比較圖	23
圖(26)	四筆 A 數據作 STFT	24
圖(27)	四筆 B 數據 x 軸作 STFT	26
圖(28)	神經網路架構圖表	29
圖(29)	每次訓練 Loss 值折線圖	30
圖(30)	每次訓練正確率折線圖	30
圖(31)	B 數據 24 測試結果	31
圖(32)	B 數據 92 測試結果	31

謝誌

感謝楊智烜老師在研究所這三年來的指導，不厭其煩的指點我。讓我吸收了許多知識，不拘束我接觸各個不同的領域，讓我慢慢摸索。在我遇到瓶頸時，也會指點迷津，幫助找到方向與解決問題的方法。實驗室裡的學弟妹，感謝他們時常幫我許多事物。感謝寶元數控的張友友經理提供的數據，讓我可以接觸到業界處理的問題，也教會了我 C++。還有系上的助教們，在一些小事上即時的幫助。最後感謝家人的支持，精神上的鼓勵。



一、研究背景

1.1 數據來源：

振動數據來源是安裝在車床主軸上的寶元數控的振動感測器 (SVI-1000)，每秒量測 1,660 次，單位是 $10^{-3}g$ 。本論文的數據有 A、B 兩組；分別來自不同工序。A 組數據共有 14 筆：11 筆為正常車刀，2 筆為車刀磨損，1 筆為切斷刀磨損，每筆數據是長度大約二十五萬的時間序列。B 組數據紀錄的是某工序持續重複作 95 次---量測車刀(組)從全新到磨損的連續過程---的振動數據，共 95 筆數據，每筆是長度大約 70,000 到 80,000 的時間序列（請注意 B 組數據的每個完整的工序都需要 3 把車刀）。



圖(1) 振動感測器



圖(2) 車刀示意圖



圖(3) 工具機示意圖



圖(4) 工具機實圖

1.2 研究動機與目的：

工具機的車刀屬於損耗品，如果車刀已磨損嚴重而未被察覺，繼續加工將會影響產品的精密度。如果能從機台的振動數據分析中，找出正常車刀與磨損車刀之間的明顯不同特徵，並且據此建立一個即時的判別系統，預測或檢測車刀的耗損狀況、甚至建議更換車刀的時機，如此必能提高產品的機密度，同時也能減少耗材成本。

本論文的數據屬於時間序列，傳統上，通常會利用傅立葉轉換將時間序列換成頻域分佈。但是這樣的作法，在頻域中看不到時間的變化。而現實的時間序列，頻域大多會隨著時間改變，只用傅立葉轉換看不出頻域隨時間變化的關係，因此有研究者在實務上改用短時傅立葉轉換（或稱作移動傅立葉轉換），將時間序列換成頻域隨時間變化的關係圖(參考[1-2])。也有研究者利用 Hilbert-Huang 轉換(參考[3])，將時間序列專換成時間與瞬間頻率之關係圖。在最近幾年，深度學習的領域有些類神經網的模型可以做時間序列的預測，例如遞迴神經網路（RNN, recurrent neural network）、長短期記憶模型（LSTM, long short-term memory）、或用 LSTM 合成生成對抗網路（GAN, generating adversarial network）作異常時間序

列的預測(參考[7])。利用深度學習的模型作異常檢測是近幾年新的嘗試趨勢，但前提是(對原始時間序列)要有合適的前置處理。本論文利用移動標準差對原始數據作前置處理，找出正常車刀與磨損車刀之間的明顯不同特徵，並與其他的常用的訊號分析方式做比較。最後嘗試利用深度學習的模型建立鑑別車刀狀況的鑑別系統。

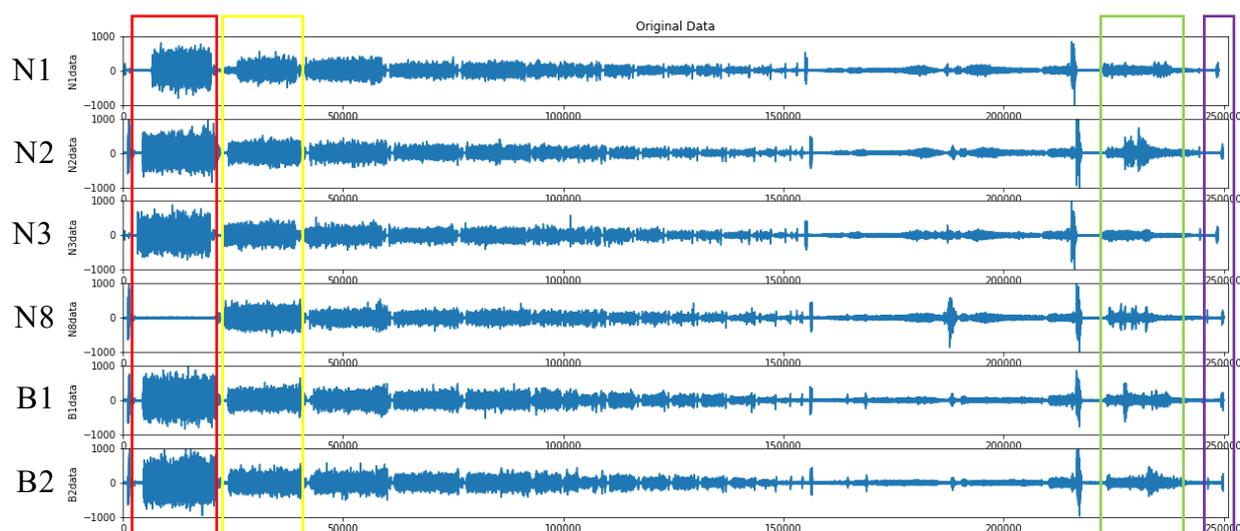
我們的研究結果發現：1. 利用移動標準差，可以明顯區別 A 組數據的正常車刀與異常車刀。2. B 組數據是新刀持續加工至車刀耗損，利用移動標準差，可以觀察出車刀持續加工時，振動數據的移動標準差之變化趨勢。

二、研究方法

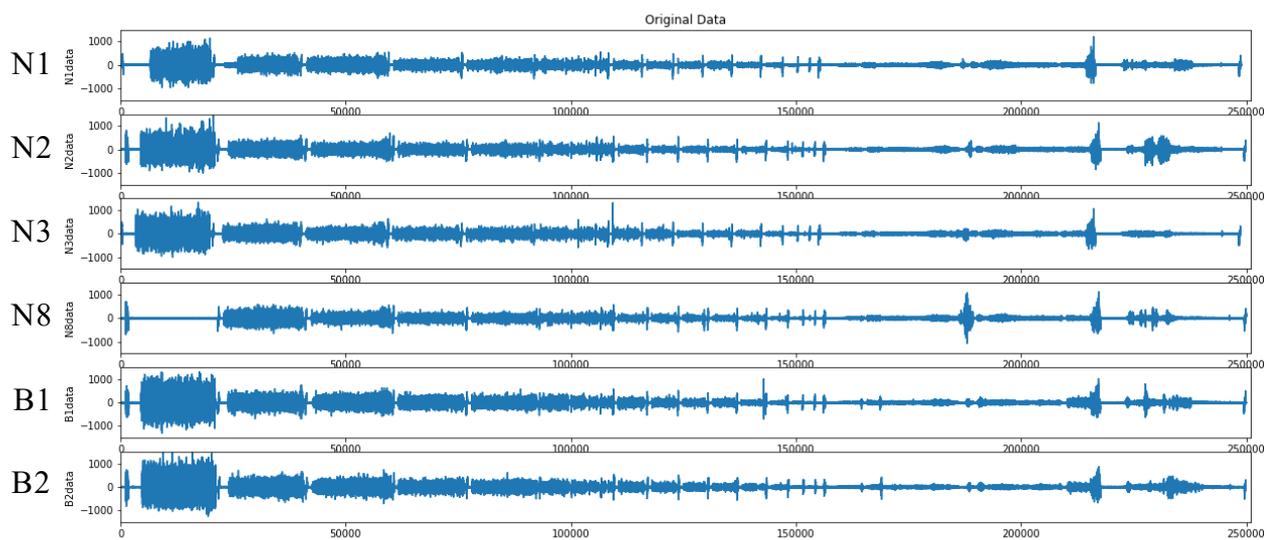
2.1 數據預處理：

A 數據：

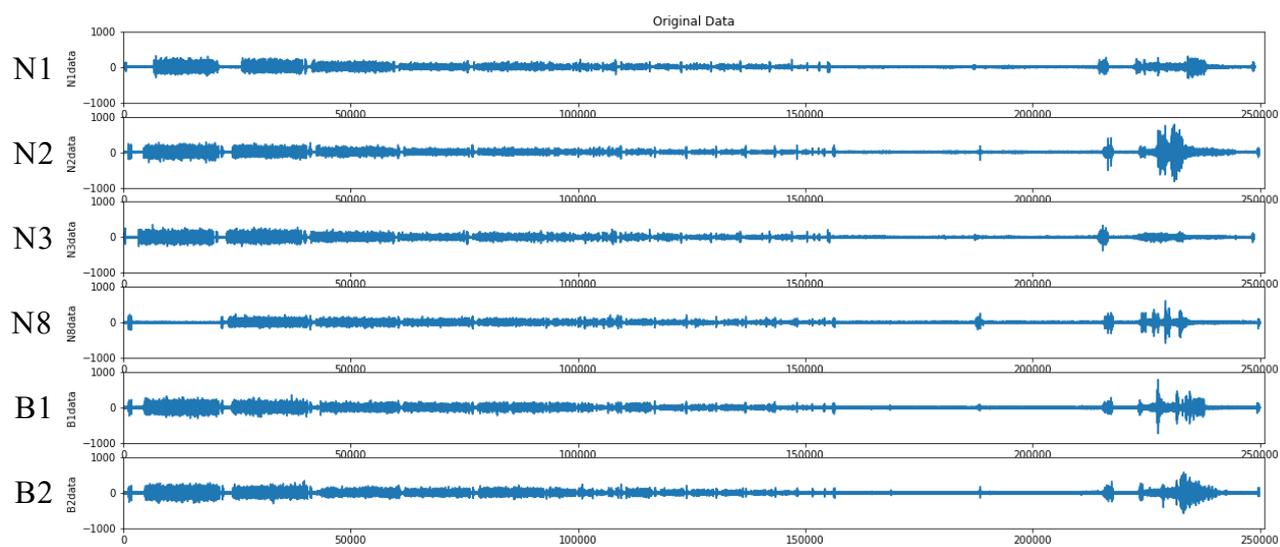
A 組數據共有 13 筆，每筆約為長度 250,000 的時間序列，正常數據有 11 筆，分別以編號 N1~N11 依序表示；車刀磨損的數據有 2 筆，分別以 B1、B2 表示。為了方便說明，以數據 N1、N2、N3、N8、B1、B2 演示，如圖(4)所示，取 x 軸的 6 筆數據觀察，由上而下，分別為數據 N1、N2、N3、N8、B1、B2。紅框為車刀歸位的過程；綠框為切斷刀切銷的部分；紫框為數據結束狀態，會有小幅的振盪。



圖(5) 六筆 A 數據 x 軸之時間序列



圖(6) 六筆 A 數據 y 軸之時間序列



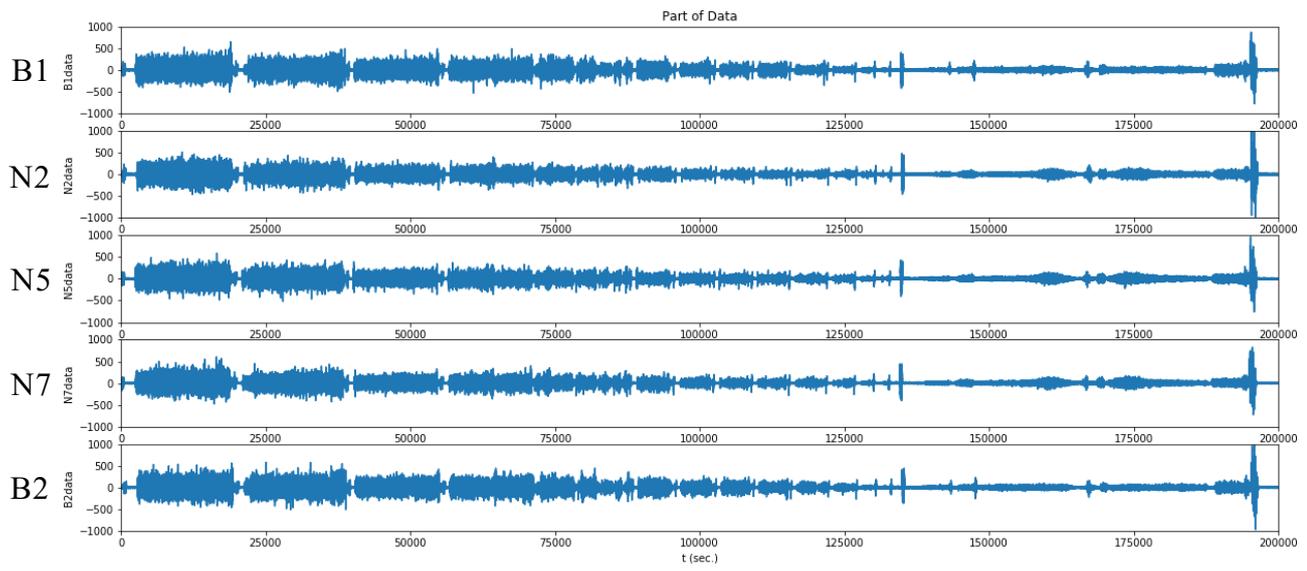
圖(7) 六筆 A 數據 z 軸之時間序列

a.數據觀察：

觀察後發現只有 x 軸較其他軸的數據有明顯變化，因此以 x 軸作為分析目標。

b.數據預處理：

1. 觀察每筆數據的圖形，發現整段工序結束會有一個小幅度的振動(圖(5)中的紫框)。以此作依據找出所有數據中時間序列最短的長度。
2. 以最短長度從後面往前取等長序列，使得結束在同一時間點。
3. 經觀察，如圖(5)所示，部分數據的紅框部分不完整，所以我們取黃框為分析起始點。
4. 接續第 3 步驟，從第 2,000 取至第 220,000 的時間點作為分析數據，目的是為了剔除紅框部分及切斷刀的部分，不讓切斷刀影響車刀的分析狀況。處理後的數據，如圖(8)所示。

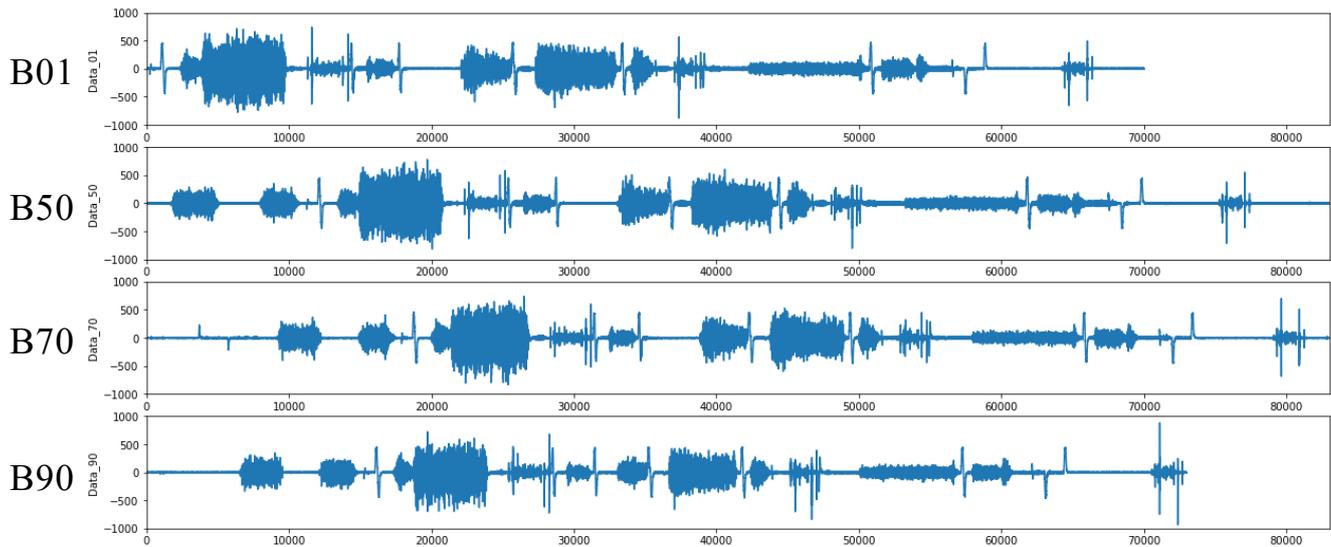


圖(8) 五筆 A 數據 x 軸預處理後之時間序列

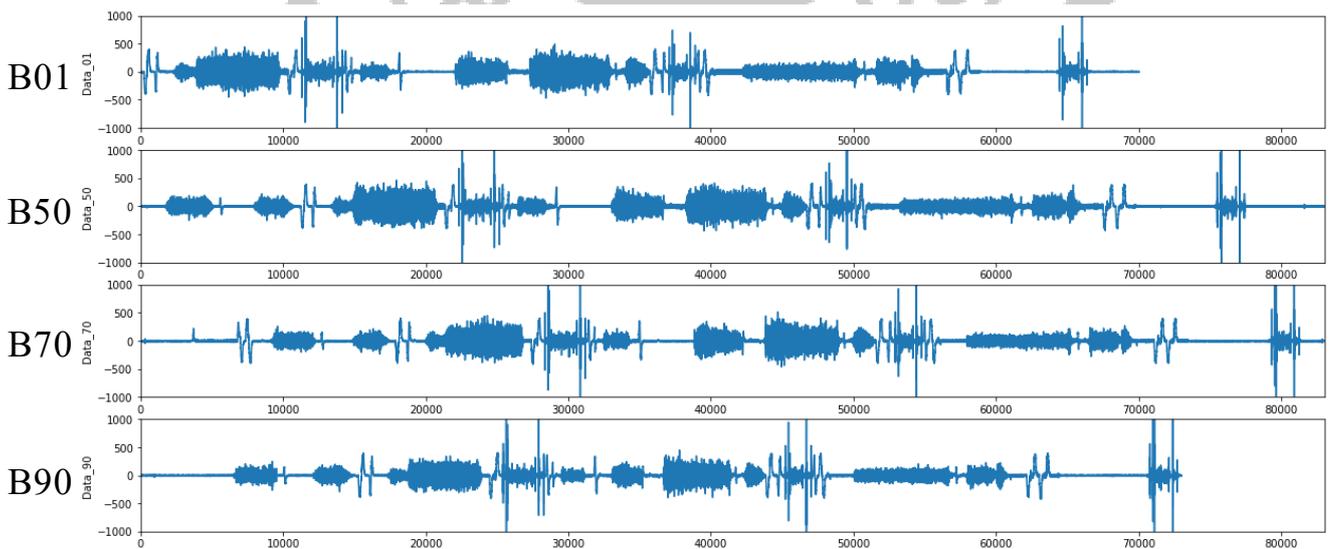


B 數據：

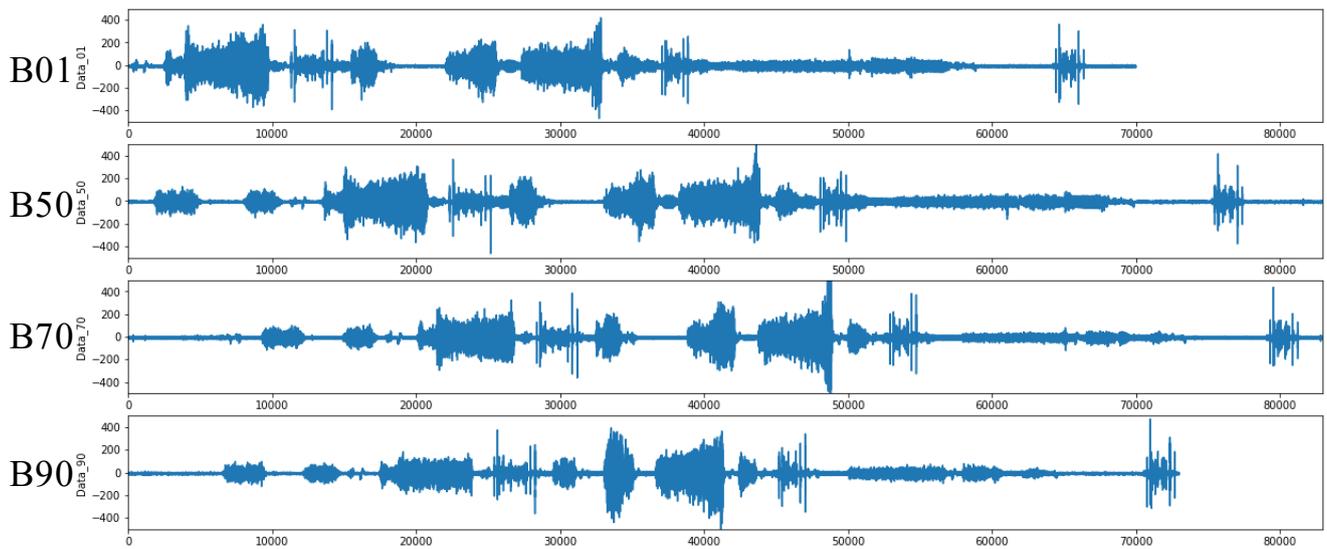
B 組數據共有 95 筆，紀錄以相同工序連續加工 95 次的振動。
但每次紀錄的長度不盡相同，數據長度最短長 69,000、最長長 83,000，數據編碼則以 B01~B95 依序表示。



圖(9) 四筆 B 數據 x 軸之時間序列



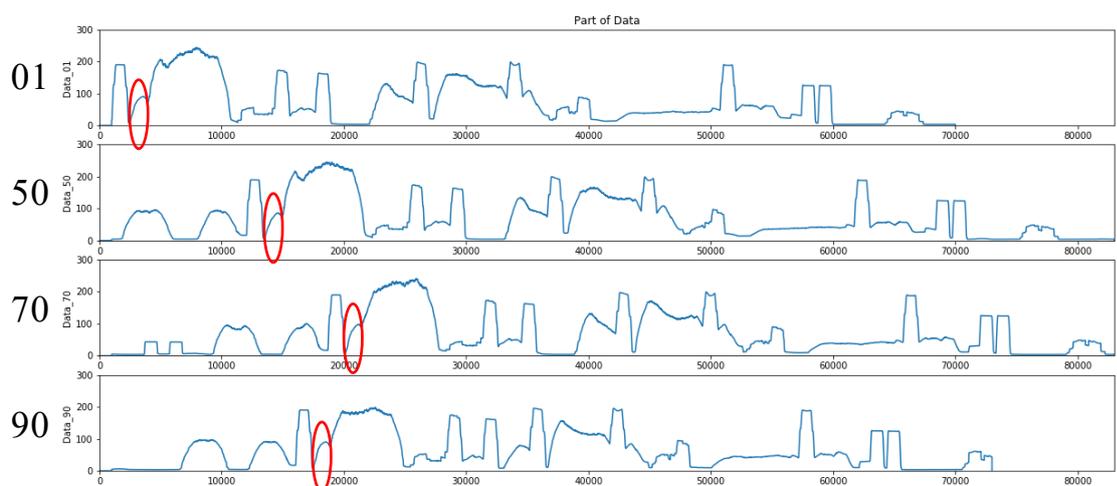
圖(10) 四筆 B 數據 y 軸之時間序列



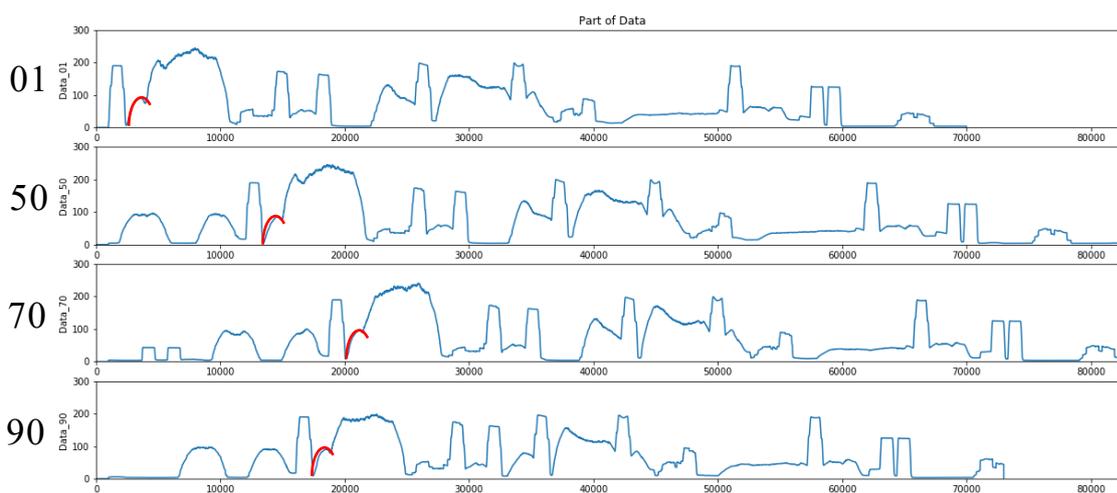
圖(11) 四筆 B 數據 z 軸之時間序列

數據預處理：

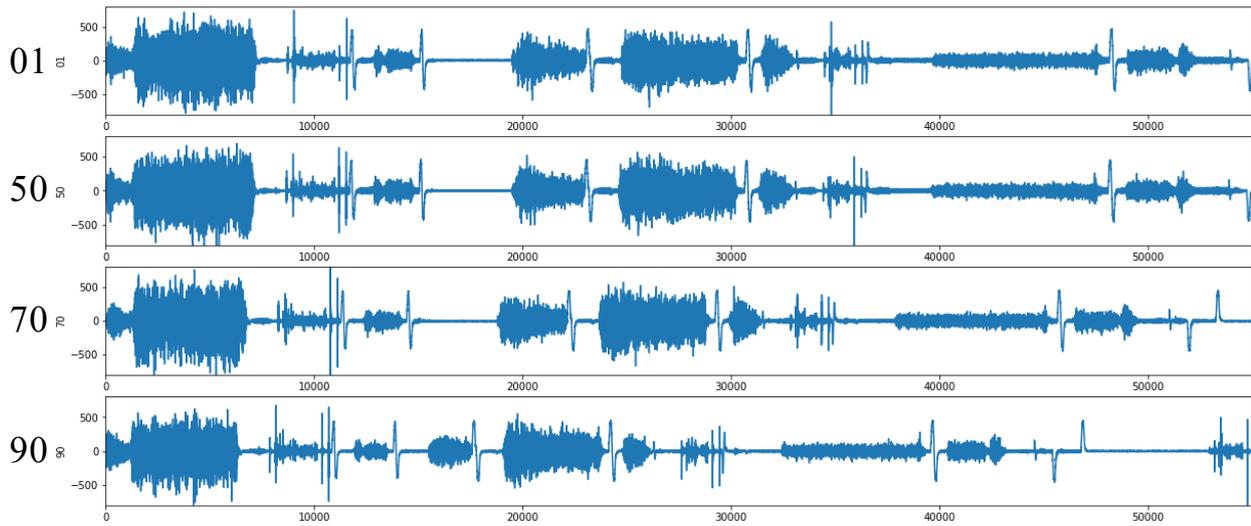
對每筆的 x 軸的數據作移動標準差，取移動標準差的窗口長度是 1,000，參考圖(9)。觀察後發現每筆數據剛開始都有相似的一小段，如圖(12)紅圈位置。因此就以 B 數據 B01 的 2,600~4,600 的這段時間的數據作為開始的標準窗口，將其它數據平移與標準窗口作均方誤差(mean-square error, MSE)，閥值為 1.5，當均方誤差小於 1.5，立即停止並提取此時間點作為開始時間點，並且至往後取總長 55,000 作為數據處理的範圍，如圖(13)中的紅色曲線之後總長 55,000 的數據。



圖(12) B 數據 x 軸作移動標準差



圖(13) 預處理示意圖



圖(14) 五筆 B 數據 x 軸預處理後之時間序列

處理後的數據，如圖(14)所示，由上至下，分別為數據 B01、B50、B70、B90，對齊時間線之後的數據。

2.2 分析工具:

2.2.1 Python

Python 語言是一種現今在數據科學中被廣泛使用的直譯式的程式語言。強調程式碼的可讀性和簡潔的語法，讓新手在分析數據時，容易閱讀和撰寫，Python 是動態語言，不像其他的靜態語言如 C 那樣需要書寫聲明語句。Python 是完全物件導向的語言，函式、模組、數字、字串都是物件。現今有許多開源的函示庫在數據分析上可以方便使用，如:NumPy、SciPy、matplotlib、Keras...等。NumPy 的核心功能在於能夠方便處理多維度的陣列運算；SciPy 是一款開源的程式庫、數學工具包。Scipy 包含的模組有最佳化、訊號處等其他科學與工程常用的計算工具；Matplotlib 是 NumPy 可視化操作介面，非常類似 MATLAB，但沒像 MATLAB 那麼人性化；Keras 是一個用 Python 撰寫的神經網路庫。

2.2.2 移動標準差

標準差(Standard deviation)，在統計上常用來測量一組數值與其平均值的(平均)分散程度。低標準差表示數據傾向於接近平均值。假設 $x(i)$ 為時間序列，令 w 是窗口長度為， μ_i 是第 i 個窗口的平均值，即 μ_i 是 $x(i-w+1) \sim x(i)$ 的平均值，

$$\mu_i = \frac{1}{w} \sum_{j=1}^w x(i-j+1)$$

則移動標準差 $S_x(i)$ 定義為

$$S_x(i) = \sqrt{\frac{\sum_{j=1}^w (x(i-j+1) - \mu_i)^2}{w}}$$

2.2.3 短時傅立葉轉換(STFT)

由法國學者傅立葉(Fourier)提出，所以以其名來命名。傅立葉轉換(*Fourier transform*)是一種線性積分變換，用於信號在時域和頻域之間的變換，在物理學和工程學中有許多應用。若 $f(t)$ 為時域原始訊號，則其傅立葉轉換為

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

其中， t 為時間， ω 為頻率。

短時距傅立葉轉換(STFT)，可以視為在一段時域信號上用一移動的窗口取出片段進行傅立葉轉換，以 $x(t)$ 為中心，窗口長度為 $2B$ 的短時距傅立葉轉換定義為

$$X(t, f) = \int_{t-B}^{t+B} x(\tau)e^{-i2\pi f\tau} d\tau$$

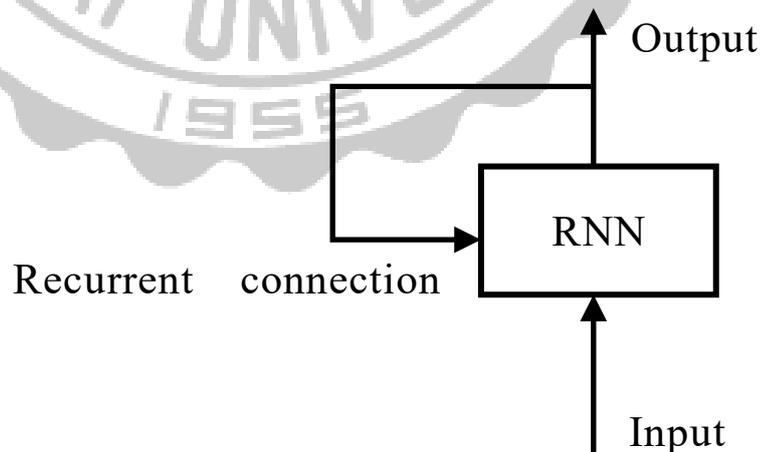
2.2.5 深度學習之長短期記憶

本論文深度學習的深度學習網路是在 Keras 上建模的。

Keras 運行在 Tensorflow、Microsoft Cognitive Toolkit、Theano 或 PlaidML 之上。Tensorflow、Theano 等是與 Keras 相較低階的程式庫，雖然可以控制深度學習模型的細節，但需要更多程式碼及花費更多時間開發，才能達成。Keras 程式庫擁有可快速建立模型、簡潔、可讀性高，這幾種特色。

長短期記憶(Long Short – Term Memory，LSTM)是一種時間遞歸神經網路(RNN，Recurrent Neural Network)。適合學習與預測時間序列。

遞歸神經網路(RNN) [6]，處理序列的方式是，遍歷所有序列元素，並保存內容的相關信息，它是具有內部環的神經網路，方塊圖如圖(14)所示。



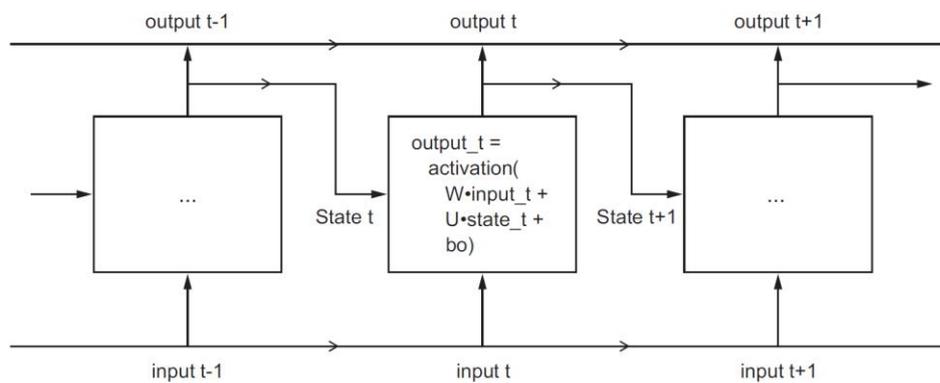
圖(15) RNN 方塊圖

RNN 是一個 *for* 迴圈，使用前一次的迭代的輸出結果 $Output(t - 1)$ 視為此次迭代的當前狀態 $state(t)$ 與當前的輸入值一起計算輸出(如圖(8))。

$$State(t) = Output(t - 1)$$

$$Output(t) = \sigma(U \cdot Input(t) + W \cdot State(t) + b)$$

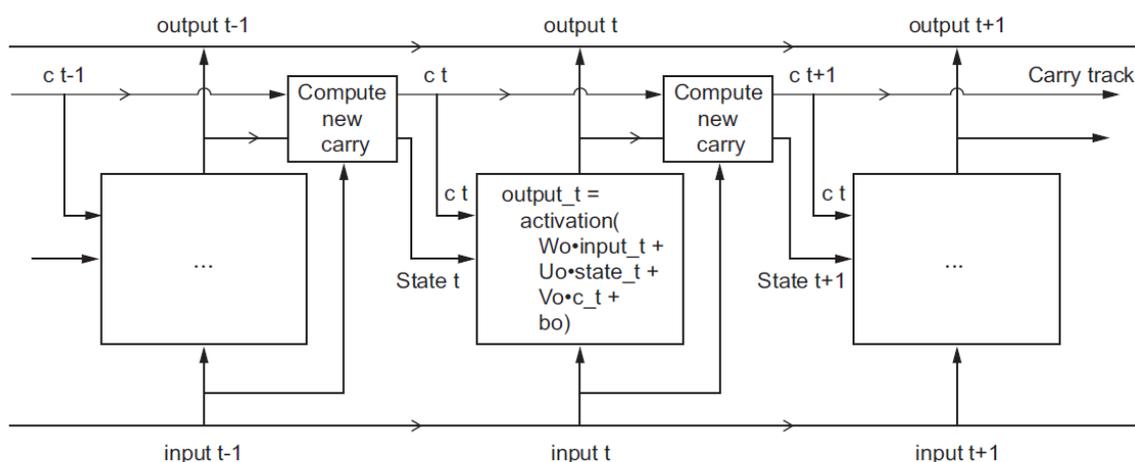
其中 U ， W 是兩個包含權重的矩陣， b 是偏移量。



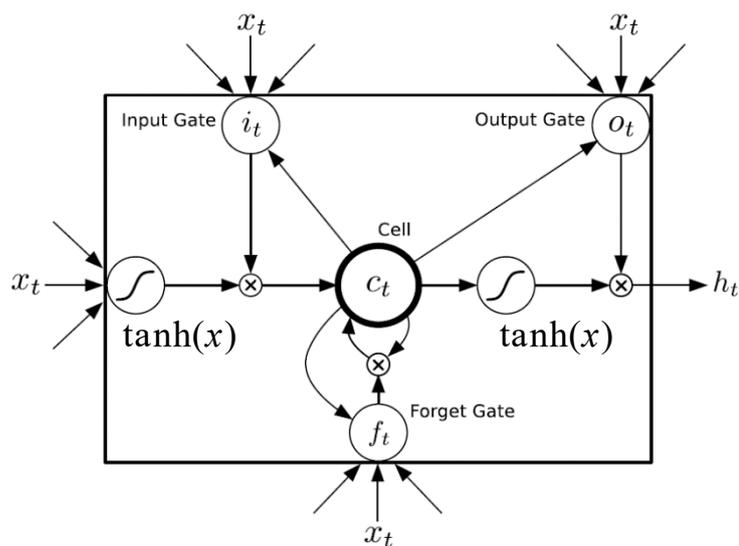
圖(16) RNN 流程圖[6]

RNN，在理論上，能夠記住許多時間的信息，但實際在訓練時，會出現梯度消失(*vanishing gradient*)的問題。為了改善梯度消失的問題，1997年由 Hochreiter 和 Schmidhuber 開發 LSTM，此演算法在，是兩人研究梯度消失問題的重要成果。

LSTM 架構可參考[4-6]，如圖(16)及圖(17)所示，增加了一種特製的值 $c(t)$ ，這個值是在圖(16)中”compute new carry”的輸出，帶著跨越多時間步的信息，使得它能夠學習數據在時間上的相關性。



圖(17) LSTM 流程圖[6]



圖(18) LSTM 架構圖[4]

$c(t)$ 視為攜帶先前訊息的值， $h(t)$ 則表示當前狀態的值，如圖

(16)所示。以下為神經元架構以函數表示：

$$i(t) = \sigma(U_i \cdot x(t) + W_i \cdot h(t-1) + V_i \cdot c(t-1) + b_i)$$

$$f(t) = \sigma(U_f \cdot x(t) + W_f \cdot h(t-1) + V_f \cdot c(t-1) + b_f)$$

$$c(t) = f(t) * c(t-1) + i(t) * \tanh(U_c \cdot x(t) + W_c \cdot h(t-1) + b_c)$$

$$o(t) = \sigma(U_o \cdot x(t) + W_o \cdot h(t-1) + V_o \cdot c(t) + b_o)$$

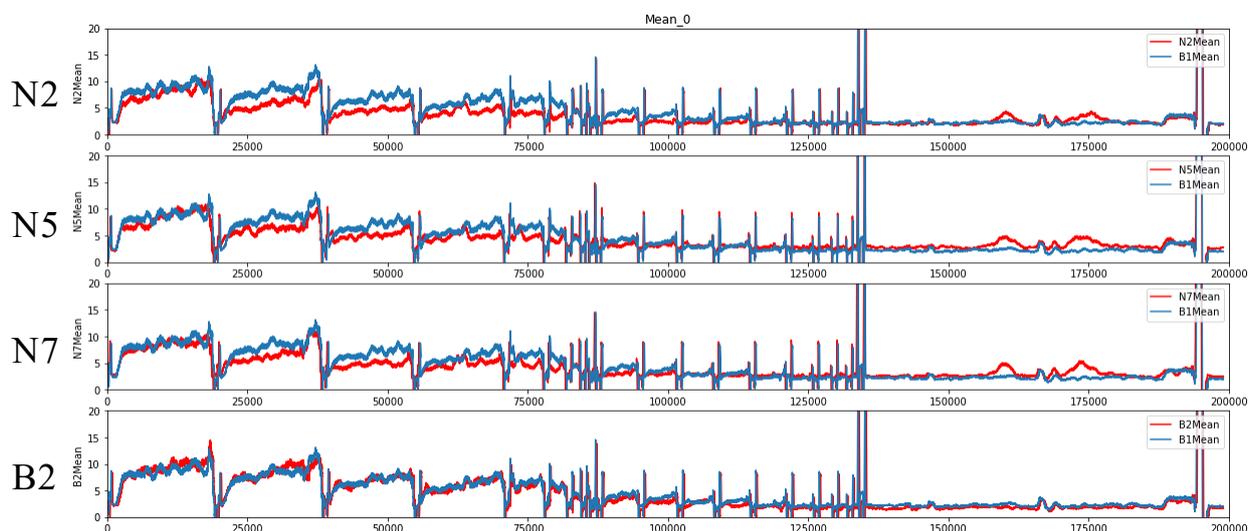
$$h(t) = o(t) * \tanh(c(t))$$

σ 為啟動函數， i 、 f 、 o 、 c 別為 *input gate*, *forget gate*, *output gate*, *cell*(在圖(16)中的”Compute new carry”)； U , W , V 分別為 $x(t)$ 、 $h(t)$ 、 $c(t)$ 的權重矩陣，若在 $i(t)$ 裡，則分別以 U_i , W_i , V_i 表示； b 為偏移量，若在 $i(t)$ 裡，則分別以 b_i 表示。

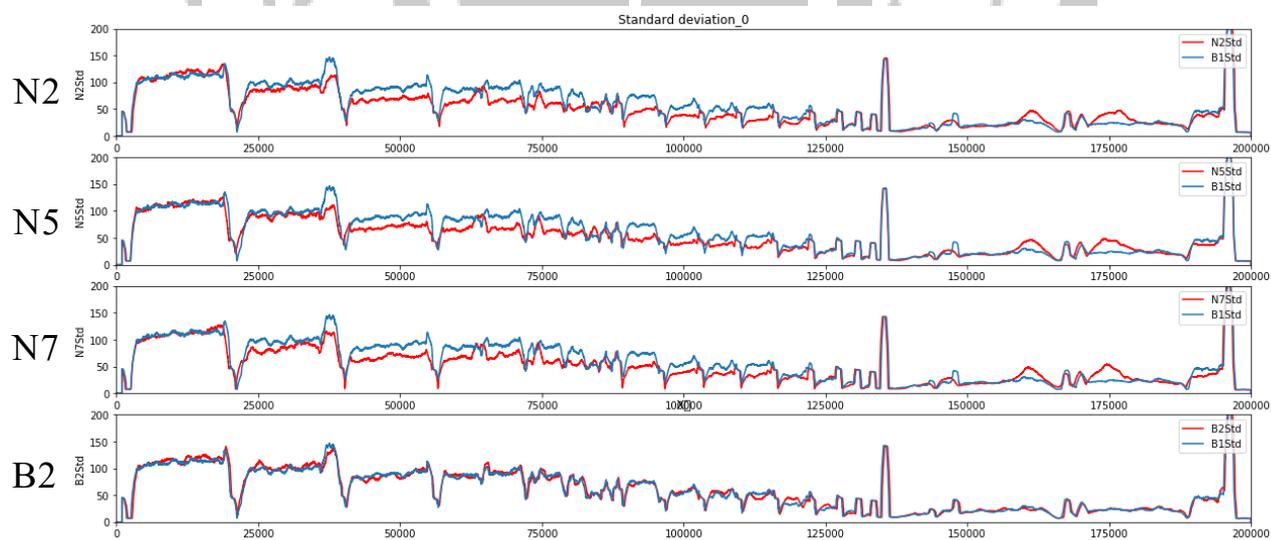
三、研究成果

3.1 移動標準差

A 數據：



圖(19) A 數據 x 軸作移動平均

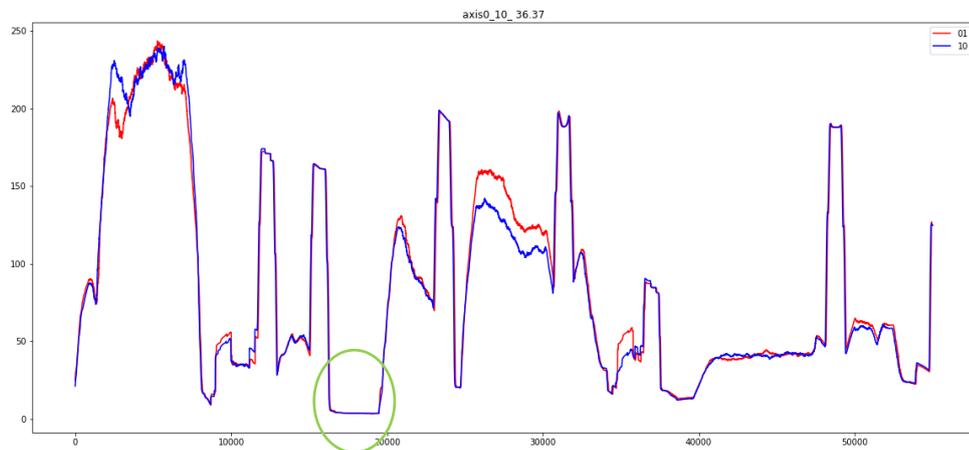


圖(20) A 數據 x 軸作移動標準差

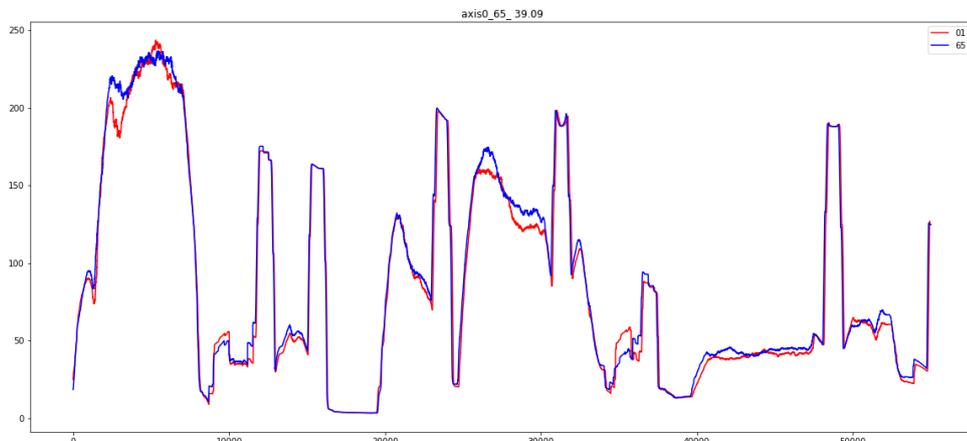
利用移動標準差及移動平均預處理後（移動標準差與移動平均皆以每個窗口以長 1000 為一個窗口寬度），發現正常車刀與磨損車刀的移動平均差距不大，如圖(18)所示。但移動標準差有明顯的差異：為了方便說明，從 x 軸的 N2、N5、N7、B2 與 B1 比較，如圖(19)所示。圖(18)-(19)中的藍線皆為 B1 數據，紅線由上至下，分別為 N2、N5、N7、B2。磨損車刀的移動標準差明顯大於正常車刀的移動標準差，其可能的解釋為：車刀磨損時，會震盪的比較劇烈，如圖(19)所示，N2 與 B1 比較，磨損時，移動標準差相對比較大。

B 數據：

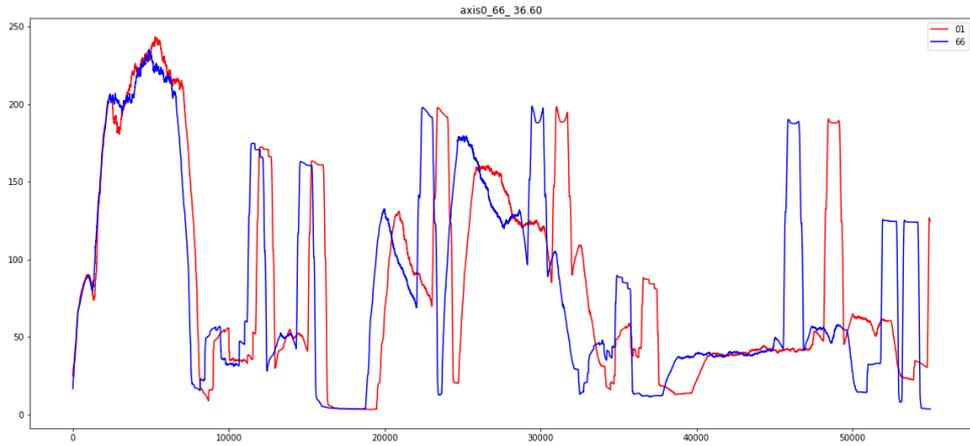
以 B10、B65、B66、B79、B80 這五筆數據 x 軸的移動標準差與與 B01 數據的 x 軸的移動標準差作比較來說明。由觀察發現 B66-B95 的數據第一次子工序加工時間會明顯縮短，而在 B80 以後 15,000~2,0000 之間的窪地 (圖(21)綠框) 會消失,窪地消失結果，如圖(25)的紫框部分。



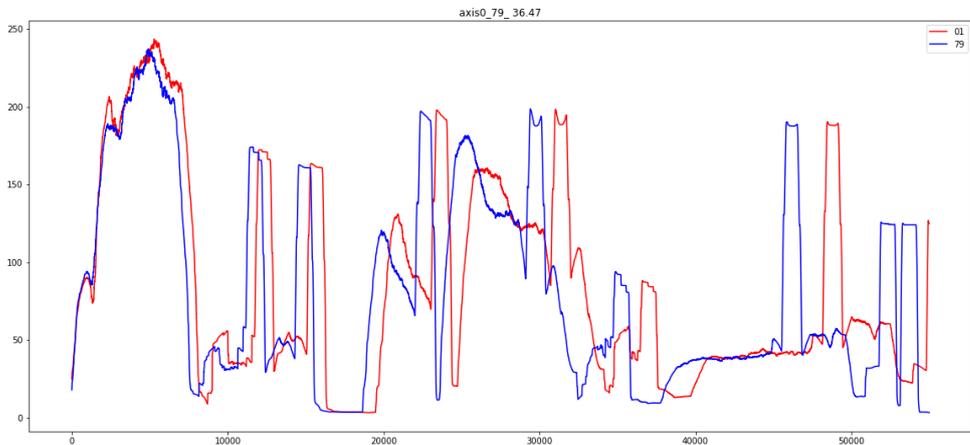
圖(21) B 數據 x 軸 B10 與 B01 比較圖



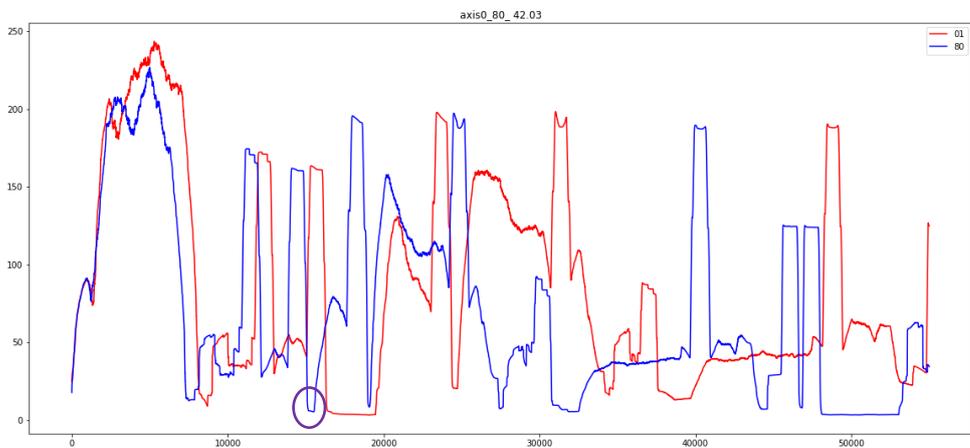
圖(22) B 數據 x 軸 65 與 01 比較圖



圖(23) B 數據 x 軸 66 與 01 比較圖



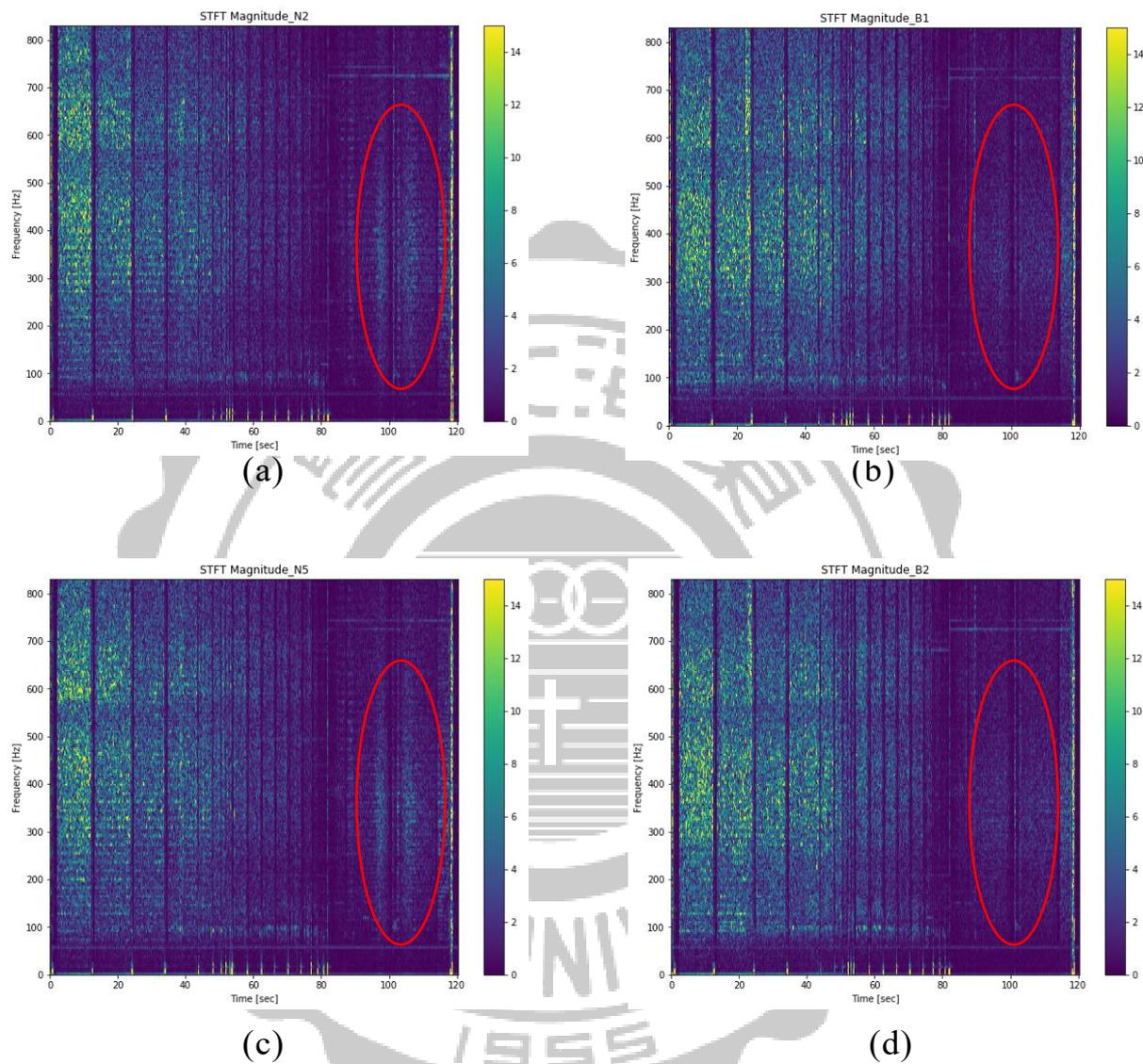
圖(24) B 數據 x 軸 79 與 01 比較圖



圖(25) B 數據 x 軸 80 與 01 比較圖

3.2 短時距傅立葉轉換(*Fourier transform*)

A 數據：



圖(26) 四筆 A 數據作 STFT

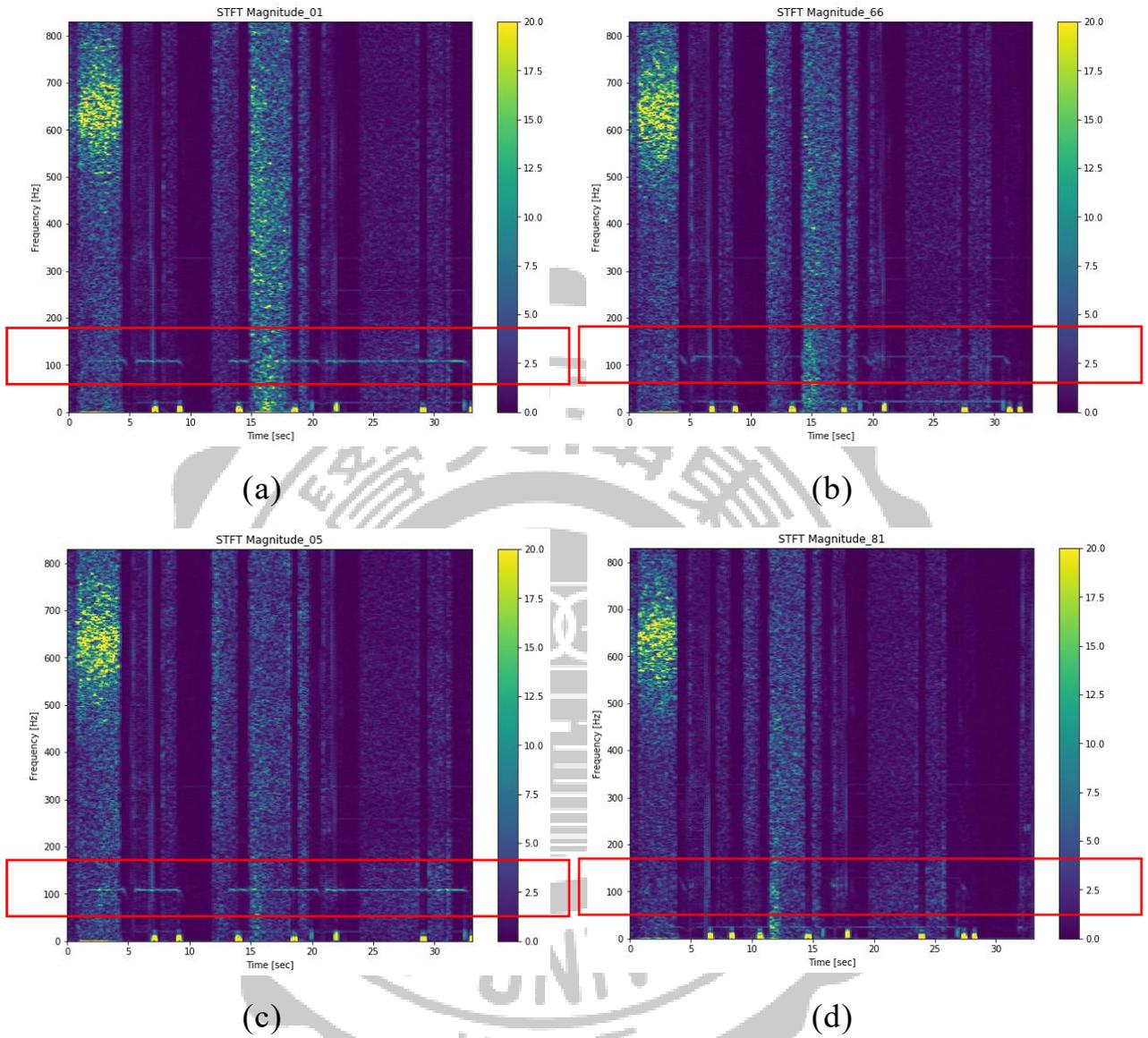
圖(25)：(a)為 N2 的 STFT 頻譜圖；(b)為 B1 的 STFT 頻譜圖；

(c)為 N5 的 STFT 頻譜圖；(d)為 B2 的 STFT 頻譜圖。

如圖(25)所示，在時間 100~120 之間(紅框)，微小的差異，正常的振幅比較大，但是很不明顯。



B 數據：



圖(27) 四筆 B 數據 x 軸作 STFT

圖(26)：(a)為 B01 的 STFT 頻譜圖；(b)為 B66 的 STFT 頻譜圖；
(c)為 B05 的 STFT 頻譜圖；(d)為 B81 的 STFT 頻譜圖。

由圖(27)觀察，有明顯看到有一些粗加工及細加工，在 B66 以後的數據第一次子工序加工時間會明顯縮短，且在頻率 100 時，振幅會變得比較小(紅框)，到了 B81 就消失了。



3.3 深度學習之 LSTM

以 B 數據作測試，用 Keras 的 Model (Functional API)來架構神經網路，以前 5,000 個時間的移動標準差點預測後 3,000 個時間點的移動標準差。

以下為神經網路架構，如圖(27)所示：

輸入數據為(5,000，1)的格式，共三層：

第一層為 4 個 LSTM 架構的神經元，以'sigmoid'為啟動函數；

第二層為 1 個一維卷積(conv1D)的神經元，以'linear'為啟動函；

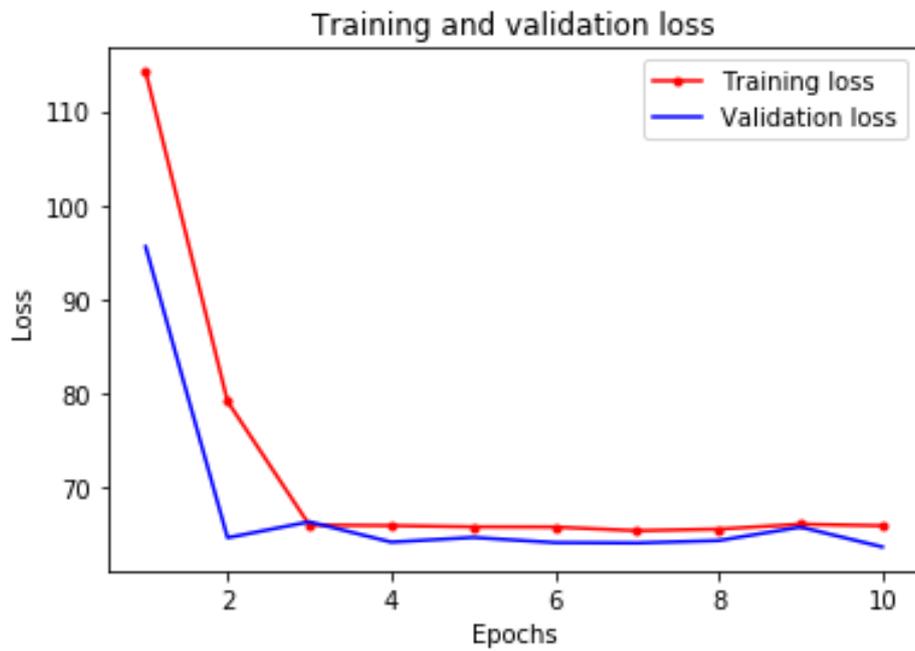
第三層為 3,000 個全連接架構(Dense)的神經元，以'relu'為啟動函數，輸出格式為(3,000，1)

Layer (type)	Output Shape	Param #
data1 (InputLayer)	(None, 5000, 1)	0
lstm_11 (LSTM)	(None, 5000, 10)	480
conv1d_11 (Conv1D)	(None, 4991, 1)	101
reshape_10 (Reshape)	(None, 4991)	0
dense_9 (Dense)	(None, 3000)	14976000
reshape_11 (Reshape)	(None, 3000, 1)	0
Total params: 14,976,581		
Trainable params: 14,976,581		
Non-trainable params: 0		

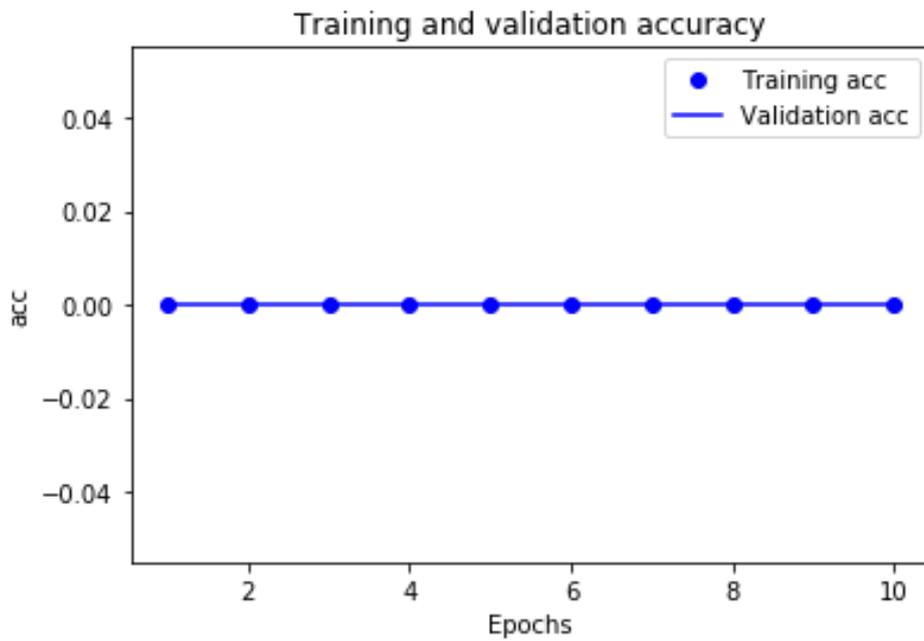
圖(28) 神經網路架構圖表

訓練過程：

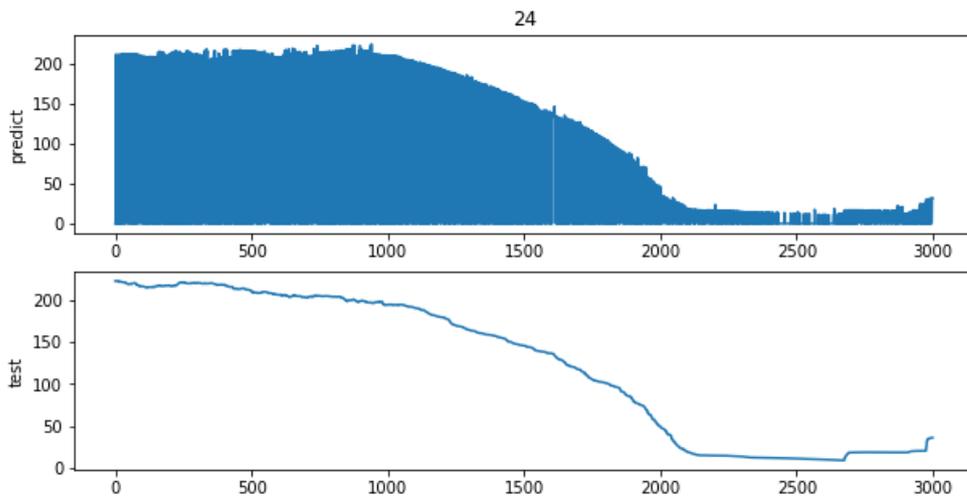
因為希望經過學習後，能根據一段時間的移動標準差預測正常車刀的未來一段時間的移動標準差。因此取數據 B01, B02, ..., B65 數據的前 5,000 個時間作為訓練資料，5,001~8,000 為訓練標籤（預測目標）。先將數據 B01, B02, ..., B65 作隨機排列，前 10 筆做測試資料（test data：不加入模型訓練），接著取後 55 筆作為訓練資料。訓練時，以”logcosh”做為 loss function，RMSprop 為優化器，再從剩下的 55 筆訓練資料取 20% = 11/55 的數據作驗證資料（validation data：為了在訓練過程中監控模型當前的精度），總共作 10 次訓練。



圖(29) 每次訓練 Loss 值折線圖

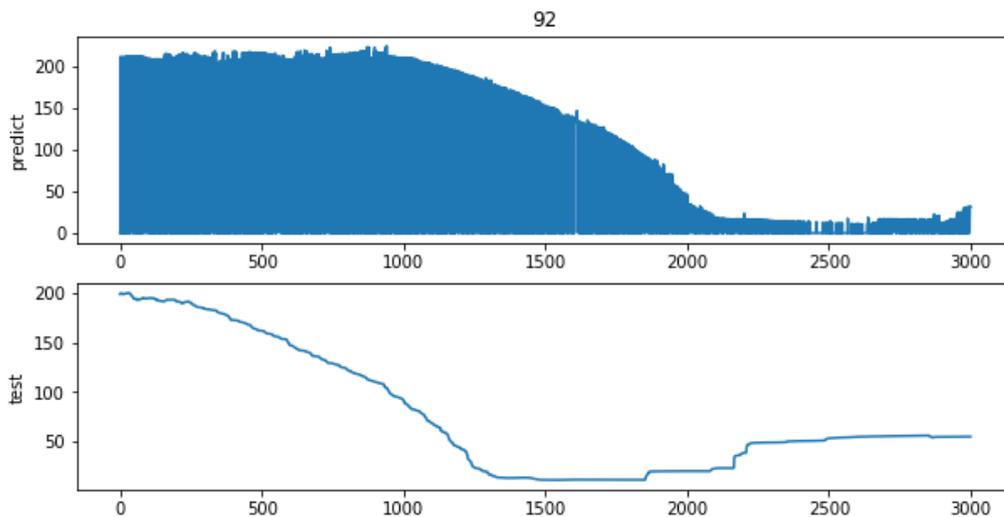


圖(30) 每次訓練正確率折線圖



圖(31) B 數據 24 測試結果

圖(31)：上圖為深度學習模型預測數據 24 的移動標準差，下圖為數據 24 的真實移動標準差。



圖(32) B 數據 92 測試結果

圖(32)：上圖為深度學習模型預測數據 92 的移動標準差數據 92 預測後的結果，下圖為數據 92 的真實移動標準差。

圖(29)-(30)顯示的是某次訓練結果的 Loss 值與訓練正確，可看出 Loss 值在訓練第 4 次後，就不再下降，所以我們取第 5 次的參數作為神經網路的預測模型的參數，並將 10 筆測試資料，放入學習好的模型，結果 Loss 值為 63.66284；正確率為 0。

(訓練或預測)正確率是 0 的可能原因可參考圖(31)-(32)---上圖分別是數據 B24、B92 (該次訓練，B24、B92 被隨機排入測試資料群)放入學習好的模型預測後 3,000 個---有不少預測值是 0，與實際值(下圖)差距過大，導致正確率為零，但 Loss 值還是相對較低。從藍色包絡可看出神經網路有學習到正常車刀的移動標準差的趨勢(數據 B24 是正常車刀，而數據 B92 是異常車刀)。

四、結論

4.1 結論：

分別對數據作移動標準差、短時傅立葉分析、Hilbert-Huang 轉換作預處理（Hilbert-Huang 轉換效果不佳，所以論文中未列出預處理結果）。發現移動標準差的方法比較能明確區別正常車刀具與異常車刀的差異。在 A 數據作移動標準差及移動平均後，發現移動平均值大致相同，但移動標準差中，磨損數據大部分會明顯大於正常數據，表示磨損數據振動比正常數據更加劇烈；在 B 數據中，雖然每筆移動標準差大小看不出有所變化，但從時間軸上觀察，峰值與低點的時間點，在數據 B66 及數據 B81 之後產生了變化。數據 B66 的變化在於第一次子工序加工時間明顯變短；數據 B80 則是在時間 20,000 前有一個窪地，它幾乎消失，使得之後的子工序更加往前平移。

4.2 未來研究方向：

在數據預處理的部分，雖然蒐集數據時，會有時間飄移的問題，我們可以先取一小段。以這一小段作為標準窗口，與其它數據平地平均誤差，當誤差落在可接受範圍，立即提取此時間點作為異常檢測的起始點。檢測方式可以用移動標準差作為檢測工具，檢測目標可為即時檢測或者是對一個工序作檢測。將預處理方式與檢測方式整合在一起，進而達到即時檢驗的效果。

數據預處理後，我們嘗試利用 LSTM 的深度學習神經網路，希望經過學習後，能根據一段時間的移動標準差預測正常車刀的未來一段時間的移動標準差。學習後以部分 B 數據做作測試，乍看似乎能大約預測出趨勢（也就是圖(29)、圖(30)的藍色包絡），但仔細觀察測試結果，有太多預測值是 0，導致圖形上看到的是整片藍色的區域，未來勢必要修正神經網路的架構使之能“完整”地正確預測正常車刀的未來一段時間的移動標準差。我們的未來目標是：1. 建立能“完整”地正確預測正常車刀的未來一段時間的移動標準差的 LSTM 及建立一個能準確計算正確率的函式。2. 將 LSTM 嵌入生成對抗網路的生成器，作為預測模型，而生成對抗網路的辨識器則可作為異常車刀異檢測器。

參考文獻

- [1] 陳瑞卿(2016)，可變視窗時頻訊號處理與輪機振動訊號應用，
國立高雄海洋科技大學輪機工程研究所碩士論文
- [2] 鄭允睿(2015)，應用時頻轉換振動訊號分析於軸承異常監視之
研究，國立中興大學機械工程學系研究所碩士學位論文
- [3] Huang, N. E., Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng,
N.-C. Yen, C. C. Tung, and H. H. Liu, The empirical mode
decomposition and the Hilbert spectrum for nonlinear and non-
stationary time series analysis. Proc. R. Soc. London, Ser. A, 454,
903–995, 1998
- [4] Alex Graves, Generating sequences with recurrent neural networks.
arXiv: 1308.0850v5 [cs.NE] 5 Jun 2014
- [5] S. Hochreiter and J. Schmidhuber, Long short-term memory.
Neural Computation, 9(8): 1735-1780, 1997
- [6] Francois Chollet, Deep Learning with Python. Manning
Publications Co., 2017
- [7] Dan Li, Dacheng Chen, Jonathan Goh, and See-Kiong Ng,
Anomaly detection with generative adversarial networks for
multivariate time series. arXiv: 1809.04758v2 [cs.LG] 9 Oct 2018