

# 智慧製造大數據 歷年初賽案例介紹

---

Chunhui Chung (鍾俊輝)

Associate Professor

Department of Mechanical Engineering

National Cheng Kung University

E-mail: [chchung@mail.ncku.edu.tw](mailto:chchung@mail.ncku.edu.tw)



## 參考資料來源

---

- 智慧製造資料庫與大數據分析第六章(註冊成大育才網後可下載，網址：  
<https://ge.ncku.edu.tw/course/view.php?id=5605>)
- 張志宏，「107-111年大數據分析競賽初賽案例介紹與範例程式使用說明」  
講義，靜宜大學資訊傳播工程學系
- 洪哲倫，「大數據競賽介紹」講義，國立陽明交通大學
- 數據下載網址：<https://github.com/smartManufacturing>

# 2024全國智慧製造 大數據分析競賽

## 總獎金330萬

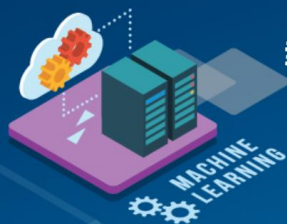
報名日期 **2024/06/12 - 2024/07/25**

歡迎國內數據分析高手一同競賽，參賽拿獎金！

線上報名

了解更多

說明會



# 回歸模型與分類模型評估指標

## 回歸模型評估:

- 均方根誤差(Root Mean Squared Error, RMSE)
- 計算預測值與實際值之間的誤差。
- 數值越小代表模型預測能力越好。

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (y'_t - y_t)^2}{n}}$$

## 分類模型評估:

- 準確率(Accuracy)
- 計算模型預測分類正確性的比率。
- 準確率越高代表模型表現越好。

$$Accuracy = \frac{\text{正確預測類別}}{\text{所有樣本數}}$$

# 加工件品質預測之應用案例

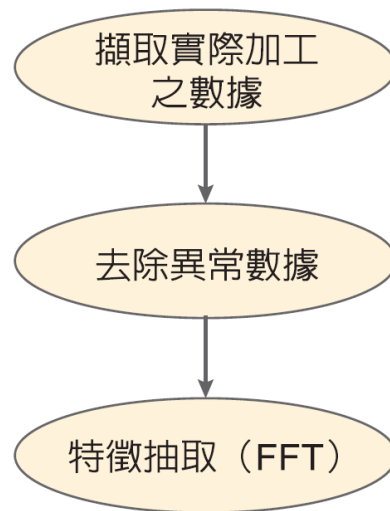
## Ch 6.3

<https://github.com/smartManufacturing/imbd2018>

# 加工件品質預測\_2018年數據競賽題目

- 在製程加工過程中，表面粗糙度數值為檢視加工品質的指標。
- 如何根據加工過程中的數據，檢視加工品質是一項重要的課題。
- 為了檢視加工件的品質好壞，此範例針對工具機台在加工過程所量測的數據，以預測表面粗糙度數值。
- 以一維卷積神經網路(1D-CNN)、多層神經網路(DNN)、循環神經網路(RNN)作為預測模型。
- 問題屬性:迴歸分析。

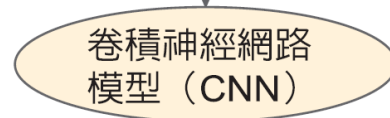
## 訊號預處理



## 模型設計



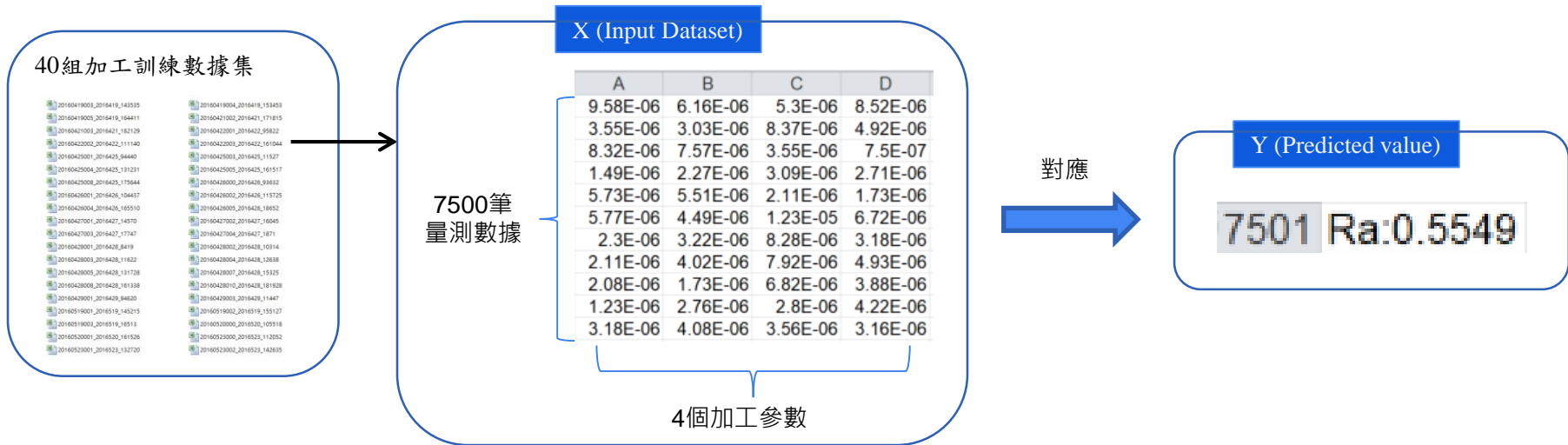
## 輸出結果



# 資料集描述

## Training Data

- 每組數據代表同一機台在一段時間內量測加工參數及相對應的粗糙度(Ra)數值(預測值)。
- 每組數據共有四個加工參數數據；共紀錄7500筆量測數據。
- 共有40組加工訓練數據，每組數據最後皆對應到一個粗糙度數值。



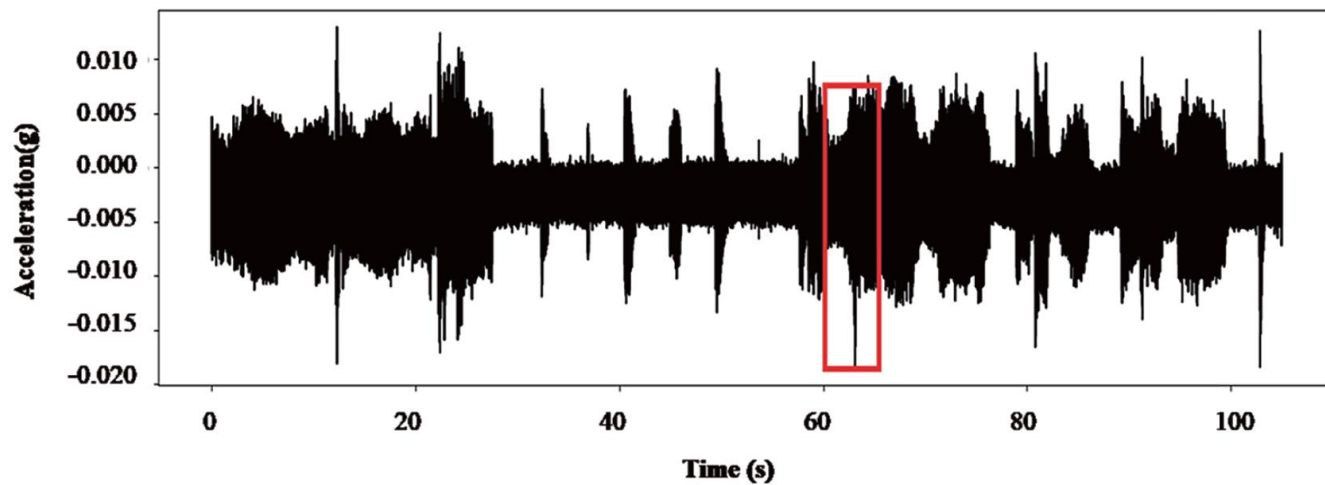
## Testing Data

- 共有10組加工測試數據集，測試數據格式與訓練數據相同。
- 唯獨不同為無提供粗糙度數值。

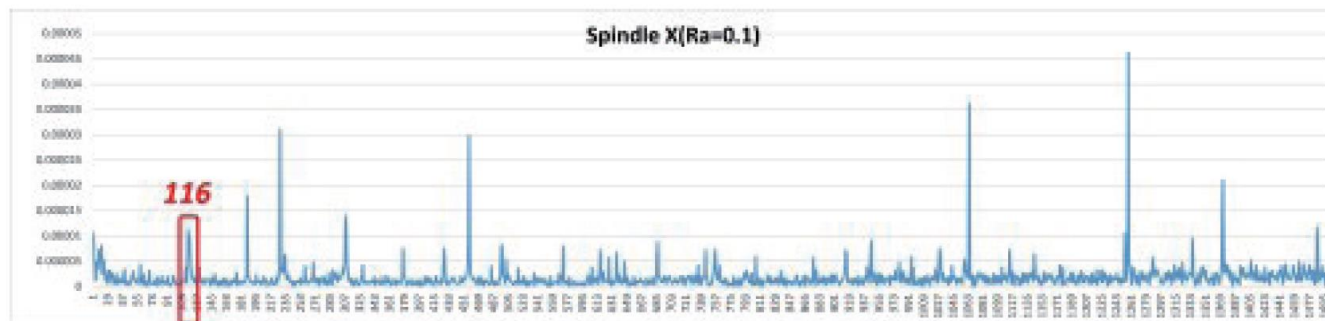


# 數據選取

- 在本範例中，數據之特徵抽取分為兩部分，第一部分利用傅立葉轉換抽取原始數據，第二部分以1-D CNN作為自動抽取原始震動訊號之特徵。
- 首先原始震動訊號餵進模型前為10000 Hz，而本範例主要針對最後5秒的精加工時間，故共有50000筆之震動訊號數據。
- 原始資料透過傅立葉轉換後，每秒訊號特徵頻譜資料減為5000 Hz作為輸入層之特徵數據。
- 此範例的銑削加工之主軸速度為116.6 Hz（7000rpm），了方便計算及解釋銑削過程的震動頻譜，在此將頻譜擴展為10倍頻之1500 Hz的頻譜，故每秒傅立葉轉換後之頻譜取前1500筆做為特徵資料，最後五秒的精加工共有7500筆。
- 所提供資料為最後5秒傅立葉轉換後之資料。
- 此外，為了證實1-D CNN可作為特徵抽取器，本範例將10000 Hz的原始震動訊號餵進1-D CNN模型以進一步作模型訓練。

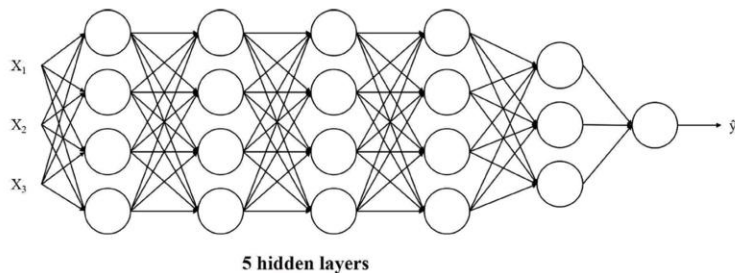


▲ 圖6-8 原始振動訊號之時域圖



▲ 圖6-9 傅立葉轉換後之訊號頻域圖

## 多層神經網路模型



- 部分所使用的模型為**DNN Model**。
- 因此透過資料預處理後得到**7500**點資料，因此將**7500**點資料壓成一維陣列，最後資料為**7500**點的一維陣列對應至一個表面粗糙度的數值。
- 在本範例中共有**50**筆資料，在模型訓練之前，需要將資料分為訓練集與測試集。
- 訓練集主要用於**DNN**模型之訓練學習的資料集；當模型訓練完成後，將測試集資料作為評估模型的依據。在此將原訓練以及測試資料共**50**筆分為**9：1**之比例區分訓練集與測試集資料。
- 然而實際的區分比例可以依據不同狀況而定，一般資料筆數較多的情況下，會使用**8：2**區分資料。



▲ 圖6-21 DNN模型之各層參數值

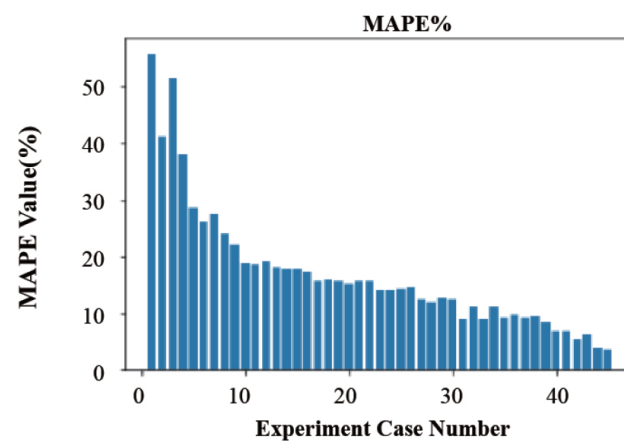
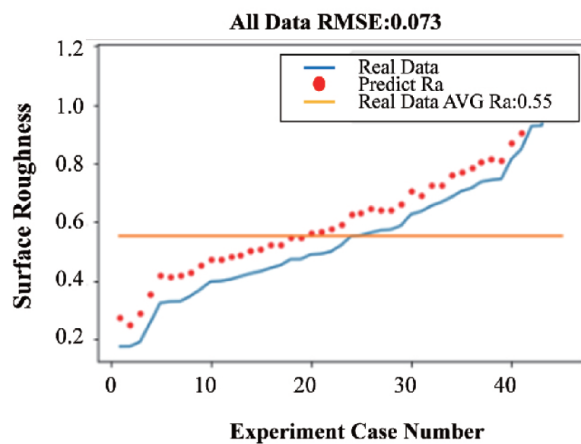
▼ 表6-1 DNN基本參數與函數名稱

參數名稱	說明	函數名稱
損失函數	神經網路在計算輸出結果時的公式	MSELoss
最佳化器	在反向傳遞時，使神經網路最佳化	Adam
學習函數	神經網路的學習數度	LR
訓練次數	神經網路反覆學習的次數	EPOCH
批量大小	一次神經網路學習幾筆資料量	BATCH_SIZE

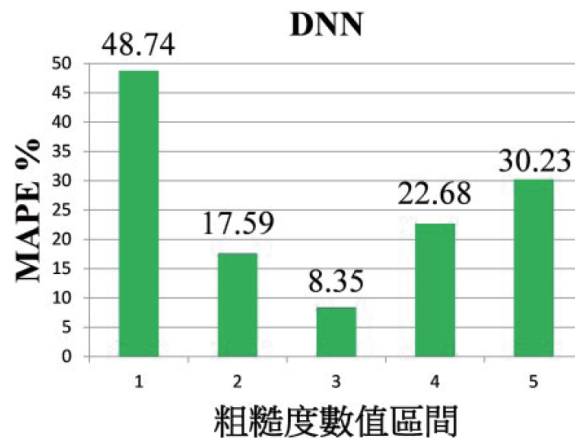
0.00035

2501

8



▲ 圖6-24 DNN模型訓練結果及個別誤差率



▲ 圖6-25 DNN模型各區間之個別誤差率

## 長短期神經網路模型

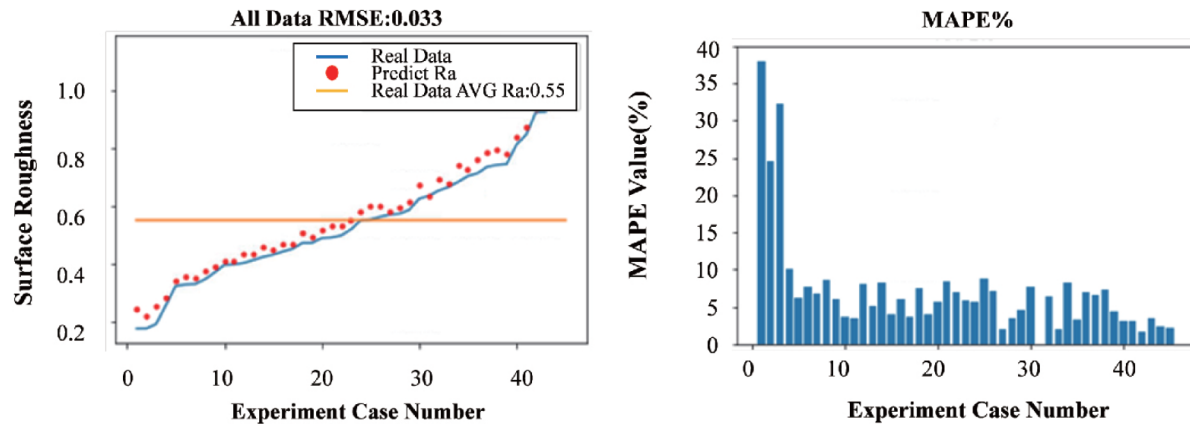
- 將資料重新排成LSTM模型要求的形式 (  $10 \times 7500 \rightarrow 10 \times 5 \times 1500$  , 分別代表為 : 資料筆數、時間單位、每個時間單中的資料點數 ) 。

```
RNN(  
    (rnn): LSTM(1500, 32, batch_first=True)  
    (out): Linear(in_features=32, out_features=1, bias=True)  
)
```

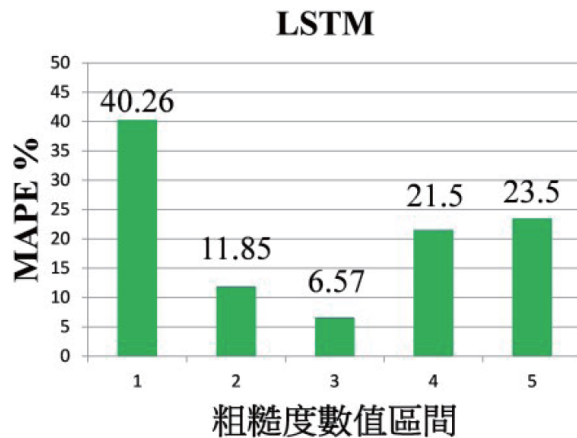
```
EPOCH = 501
```

```
BATCH_SIZE = 10
```

```
LR = 0.035
```



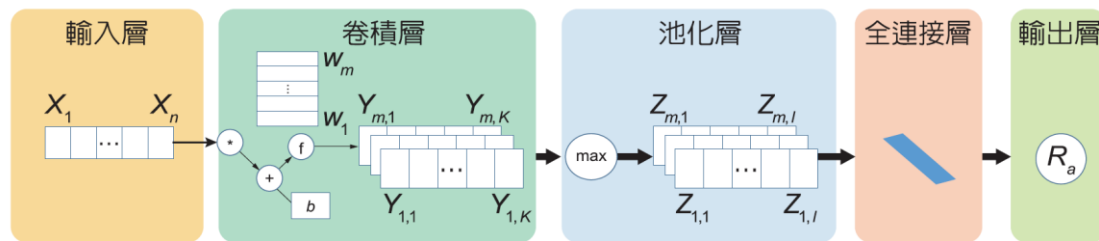
▲ 圖6-31 LSTM模型訓練結果及個別誤差率



▲ 圖6-32 LSTM模型各區間的個別誤差率

# 卷積神經網路模型

- 目前已有許多文獻提出一維卷積神經網路模型可有效的將原始訊號進行特徵抽取。利用**1-D CNN**最大的優點在於不需要任何的特徵抽取器，此方法可以直接處理原始數據。
- 本範例屬於單一參數多點資料的特性，此資料共有**50**筆，每筆資料包含**10000**點資料的振動訊號，而在最後精切削為**5**秒，因此本範例將原始數據型態**10000**點資料的振動訊號搭配一維卷積神經網路（**1D-CNN**）模型，因此先將資料壓成一維陣列，這時整體資料狀態為**50筆\*50000**點。



▲ 圖6-40 1-D CNN模型架構



# 卷積神經網路模型設定

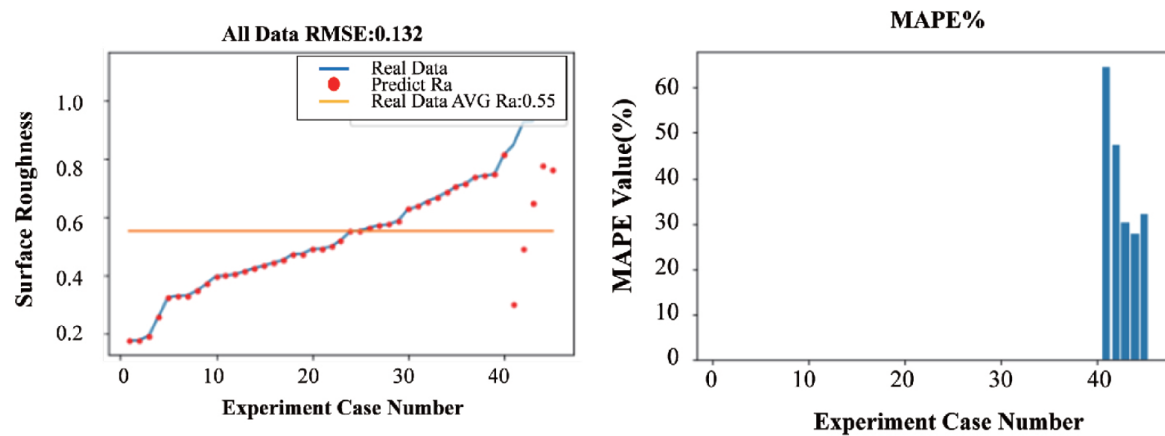
```
CNN(  
  (conv1): Sequential(  
    (0): Conv1d(1, 16, kernel_size=(500,), stride=(300,), padding=(250,))  
    (1): ReLU()  
    (2): MaxPool1d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)  
  )  
  (out): Linear(in_features=1328, out_features=1, bias=True)  
)
```



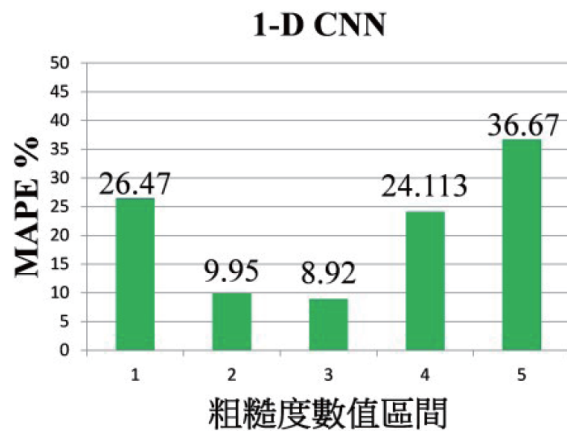
▲ 圖6-43 1-D CNN模型之架構

EPOCH = 2501; BATCH\_SIZE = 8; LR = 0.00035

Loss: M S E



▲ 圖6-47 1-D CNN模型訓練結果及個別誤差率



▲ 圖6-48 1-D CNN模型各區間的個別誤差率

# 溫度數據分類之應用案例

## Ch 6.7

<https://github.com/smartManufacturing/imbd2019>

# 溫度數據分類\_2019年數據競賽題目

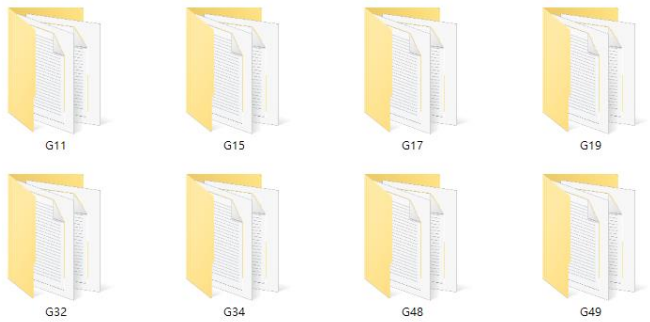
- 複合材料加工是一項複雜的加工製程。複合材料的成型需同時藉由加熱及加壓之熱壓爐成型加工。在加工過程中，若熱壓爐因過於高溫導致洩氣，將產出不良的複合材料產品品質。因此，如何在熱壓爐成化階段，即時預測溫度變化所產出的類別是一項重要的需求。
- 本範例主要根據不同的加工溫度，以預測不同溫度變化之類別。在此將加工溫度對產品品質的影響分為八個類別。
- 以深度學習技術中的卷積神經網路（CNN）作為模型，預測產品品質類別。
- 問題屬性:預測分類

# 資料集描述(1)

## Training Data

- 共分為8類別(G11、G15、G17、G19、G32、G34、G48、G49)
- 每個類別有數個爐數，共230個爐數，其中每個爐數內有5~8個PTC溫度數據欄位

8個類別



類別	爐數
G11	23
G15	31
G17	16
G19	30
G32	32
G34	33
G48	31
G49	34
Total	230



爐在一段連續時間內量測的數據

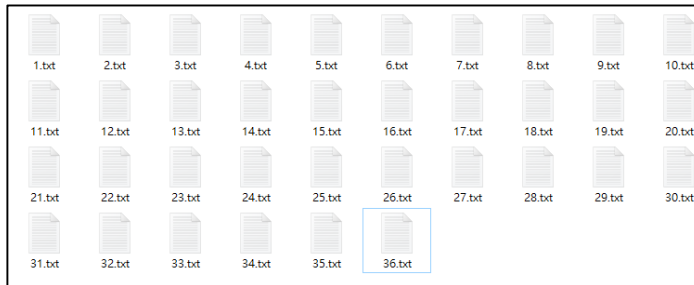
G11-PTC13 Deg.F	G11-PTC14 Deg.F	G11-PTC15 Deg.F	G11-PTC16 Deg.F	G11-PTC17 Deg.F
65.9	65.8	64.2	64.9	66.0
65.9	65.8	64.2	64.9	66.0
65.9	65.8	64.2	64.9	66.0
65.9	65.8	64.2	64.9	66.0
66.6	67.2	65.7	65.6	67.3
68.0	68.9	67.5	66.4	68.6

一段時間內  
量測數據

5~8個PTC溫度數據欄位

## Testing Data

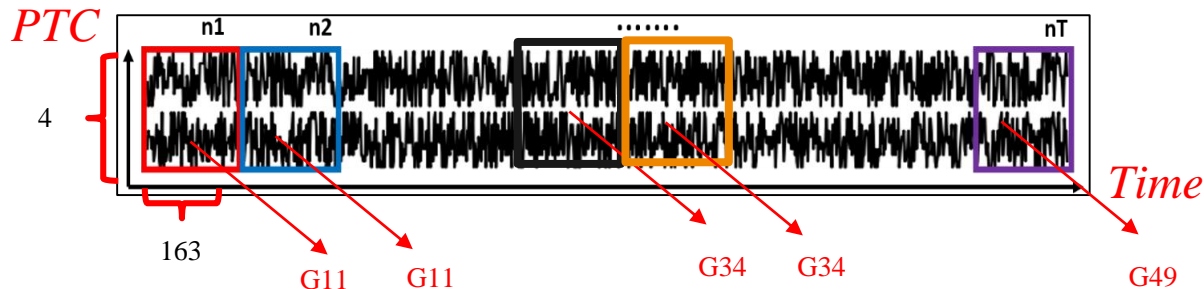
- 共36爐數，判斷各爐屬於哪個類別



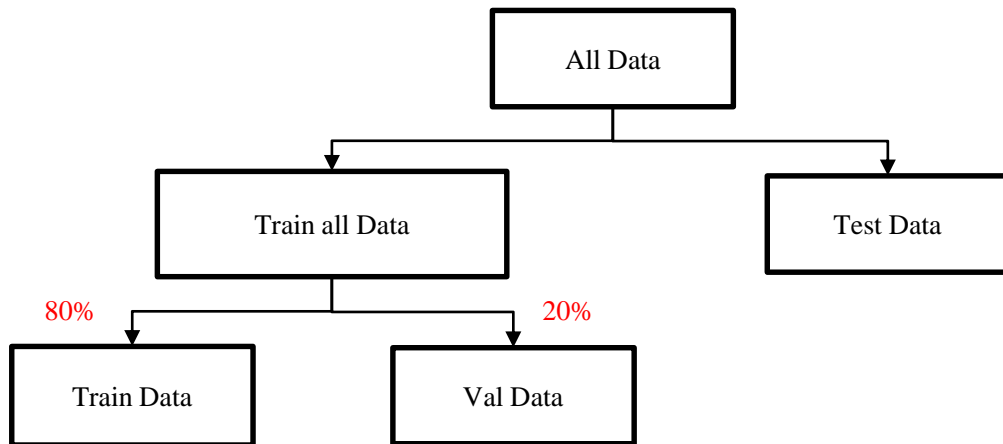
# 資料集描述(2)

## □ Data Define

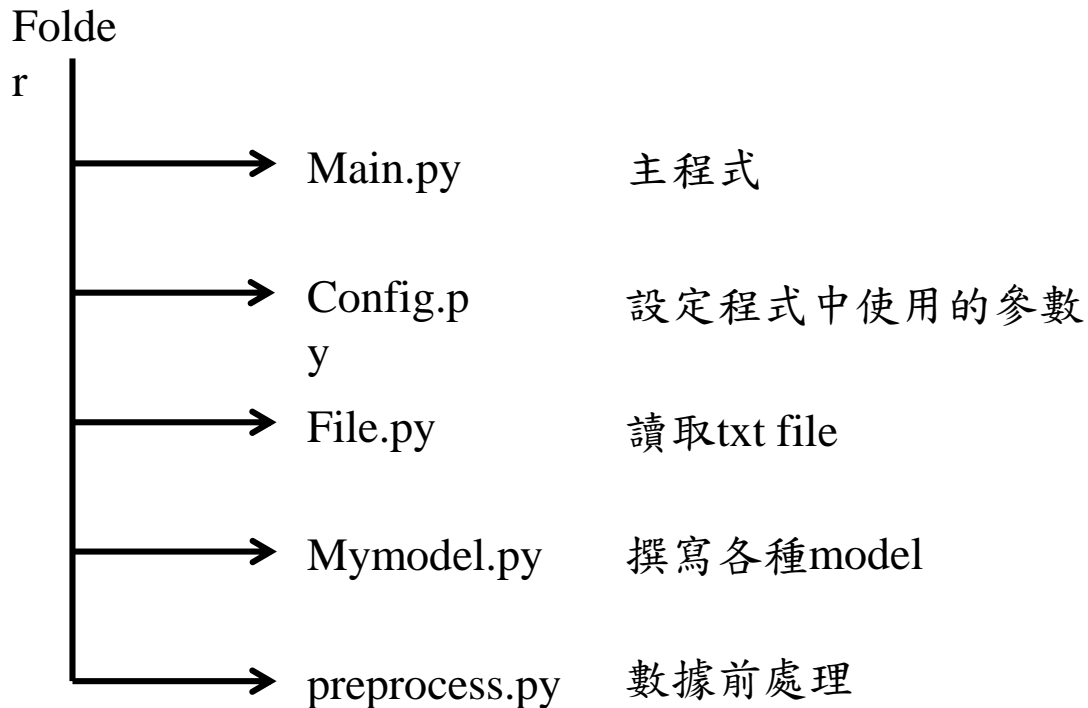
- 每一爐統取4筆PTC資料，時間取163個時間點→視為單張圖片為163x4 Image(如圖)



## □ Data Architecture



# 程式架構定義



# CNN模型架構

```
16         model = Sequential()
17         model.add(Conv2D(32, kernel_size=(3, 3),
                           activation='relu',
                           input_shape=(img_rows, img_
18 cols,1)))
        model.add(Flatten())
        model.add(Dense(128, activation='relu'))
        model.add(Dense(256, activation='relu'))
        model.add(Dense(512, activation='relu'))
        model.add(Dense(512, activation='relu'))

        model.add(Dense(1024, activation='relu'))
        model.add(Dense(1024, activation='relu'))
        model.add(Dense(2048, activation='relu'))
        model.add(Dense(2048, activation='relu'))
        model.add(Dense(2048, activation='relu'))
        model.add(Dense(1024, activation='relu'))
        model.add(Dropout(0.5))
        model.add(Dense(Set.class_num,
activation='softmax'))
```



# CNN模型訓練結果

- 訓練過程：

```
Epoch 64/70
184/184 [=====] - 1s 3ms/step - loss: 0.2093 - acc: 0.8478
: 0.2609
Epoch 65/70
184/184 [=====] - 1s 3ms/step - loss: 0.2174 - acc: 0.8750
: 0.2609
Epoch 66/70
184/184 [=====] - 1s 3ms/step - loss: 0.2061 - acc: 0.8641
: 0.2609
Epoch 67/70
184/184 [=====] - 1s 3ms/step - loss: 0.2010 - acc: 0.8641
: 0.2609
Epoch 68/70
184/184 [=====] - 1s 3ms/step - loss: 0.2041 - acc: 0.8804
: 0.2609
Epoch 69/70
184/184 [=====] - 1s 3ms/step - loss: 0.1926 - acc: 0.8804
: 0.2609
Epoch 70/70
184/184 [=====] - 1s 3ms/step - loss: 0.2162 - acc: 0.8587
```

- 預測結果：

```
predict: [4 2 3 3 4 4 4 4 4 4 5 4 5 5 5 5 5 6 6 6 6 1 6 5 5 3 5 5 1 1 1 1 2]
Real: [0 2 3 3 4 4 4 4 4 4 5 0 5 5 5 5 6 6 6 6 1 6 7 7 7 7 7 1 1 1 1 1 2]
Accuracy(%): 77.77777777777779
```

# 加工機台參數預測之應用案例

## Ch 6.8

<https://github.com/smartManufacturing/imbd2020>

# 加工機台參數預測\_2020年數據競賽題目

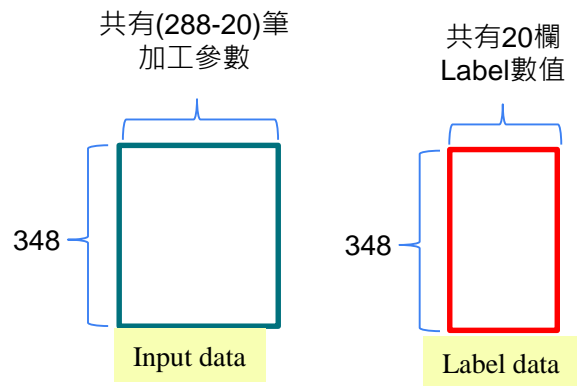
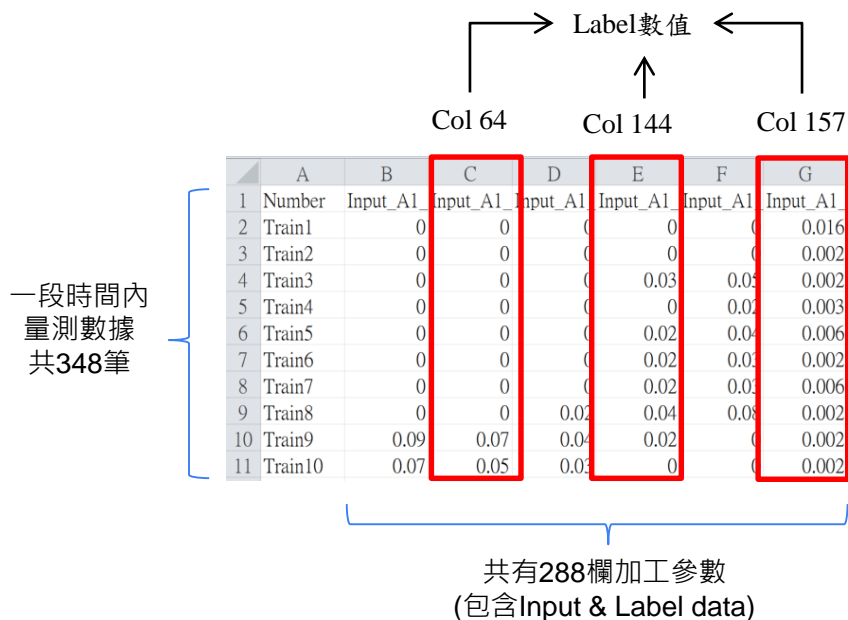
- 資料來源為鑽孔機加工品質的輸出
- 在加工製程中，不同加工參數大幅影響加工品質。
- 可藉由模型來評估此加工參數對於加工品質的好壞。
- 本範例主要根據不同的加工參數，以預測不同的加工品質之數值。
- 以Extra tree regressor作為預測模型。
- 問題屬性:迴歸分析

# 目標

- 以加工機台完整的「加工參數」和「加工品質」作為訓練資料，於測試階段預測 20項重點參數，20項重點參數分別為：
- Input\_A6\_024、Input\_A3\_016、Input\_C\_013、Input\_A2\_016、  
Input\_A3\_017、Input\_C\_050、Input\_A6\_001、Input\_C\_096、Input\_A3\_018、  
Input\_A6\_019、Input\_A1\_020、Input\_A6\_011、Input\_A3\_015、  
Input\_C\_046、Input\_C\_049、Input\_A2\_024、Input\_C\_058、Input\_C\_057、  
Input\_A3\_013、Input\_A2\_017。

# 資料集描述

內含總共**348**筆資料。每筆資料包含**281**項加工機台參數設定與**6**個鑽孔機加工品質的輸出結果，其中每筆輸入資料包含**137**項**6**個鑽孔機（A1~A6）的共同參數以及**144**項單一鑽孔機的各別參數。



Model 1: 對於預測Col 64欄位的數值  
Input = 348\*268  
Label = 348\*1

Model 2: 對於預測Col 144欄位的數值  
Input = 348\*268  
Label = 348\*1

以下類推(分別建立20個模型來預測數值)

# 資料集描述 (II)

▼ 表6-6 參數說明

Input_C_001~137	與Output_A1~A6相關的共同參數
Input_A1_001~024	與Output_A1 相關的參數
Input_A2_001~024	與Output_A2 相關的參數
Input_A3_001~024	與Output_A3相關的參數
Input_A4_001~024	與Output_A4相關的參數
Input_A5_001~024	與Output_A5相關的參數
Input_A6_001~024	與Output_A6相關的參數
Output_A1~A6	六個鑽頭加工品質的輸出結果

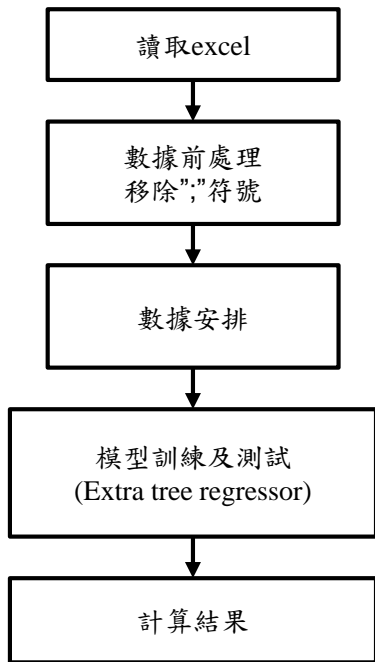
## 資料集描述 (III)

在Input\_C\_015~038與Input\_C\_063~082包含測量偏移量的文字參數，其說明如下：

▼ 表6-7 \_C測量偏移量的文字參數說明

N; 0	無偏移
R; 1	向 $x^+$ 方向偏移1個單位
L; 1	向 $x^-$ 方向偏移1個單位
U; 1	向 $y^+$ 方向偏移1個單位
D; 1	向 $y^-$ 方向偏移1個單位

# 方法流程



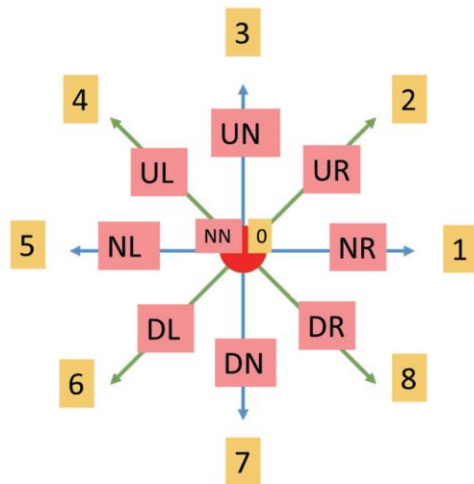
## ■ *Extra Tree Regressor*

- 基於決策樹所建立的模型。
- Extra tree主要針對劃分節點及閾值，採用隨機性方式挑選。
- 可避免模型訓練的過擬合。



# 資料預處理

- 缺失值處理對於缺失值補上平均，刪除缺失值過多的欄位，對於非數值之資料（偏移量）的前處理，則採取將偏移量的資料類比為向量的資料，將其方向給予對應的數字，偏移量編碼如圖6-64所示。



▲ 圖6-64 偏移量編碼

# 分析結果

- 最後WRMSE的結果可能因為python版本而有些許不同，以下列出部分版本分析結果。
- python version 3.7.3: WRMSE:0.06015
- python version 3.7.4: WRMSE:0.05502
- python version 3.7.5: WRMSE:0.05863
- python version 3.7.6: WRMSE:0.06278

# 設備感測資料於物理特性值預測之應用案例

## Ch 6.9

<https://github.com/smartManufacturing/imbd2021>

## 設備感測資料於物理特性值預測\_2021年數據競賽題目

- 工具機的加工精度除本身系統控制因素，也受外部環境影響。而在環境影響因素中，溫度變化將使工具機主軸依照熱漲冷縮原理產生細微差異量，此差異量稱為伸長量。又因工具機主軸可移動方向分為X、Y、Z三軸，進而也需分別計算出三軸各自的伸長量進行補正。因此在加工過程中，如何依工具機整體溫度變化預測出各軸伸長量進行參數補償是一改善加工精度的重大關鍵。
- 透過溫度感測器收集的溫度變化數據做為特徵，並以Z軸伸長量作為預測目標。問題屬性:迴歸分析

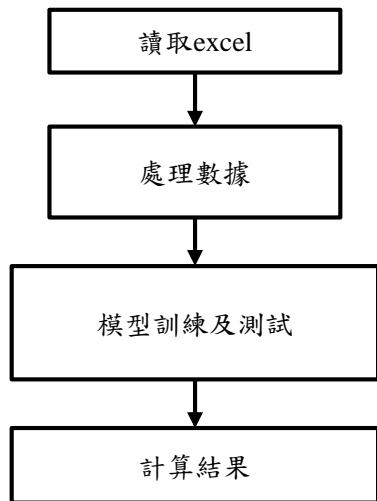
## 資料集描述

- 此範例共有2份csv，分別為train.csv、test.csv。
- train.csv為訓練數據，此數據是透過12個佈署在工具機各個位置的溫度感測器收集，並表示為F1至F12的特徵欄位，O欄位為Z軸伸長量。
- test.csv為測試數據，欄位表示與訓練數據相同，但為不同次加工收集，此份用於驗證模型預測的正確性。

▼ 表6-8 設備感測資料

欄位編號	欄位名稱	欄位說明
欄位1	SeqNo	筆數編號
欄位2~14	F_1 ~ F_13	特徵欄位
欄位15	O	目標值

# 方法流程與結果



## □ 長短期記憶模型(Long short-term memory)

- 透過使用Input Gate、Output Gate、Memory Cell以及Forget Gate四個部分為模型架構，藉以解決訓練深度網路常遇到的梯度消失與梯度爆炸的問題
- 特別適合用於處理時間序列類型的資料與自然語言處理相關應用的問題。
- Time Step 設為30。
- MAE約為8.607。

## □ LightGBM

- LightGBM是微軟團隊所提出基於梯度提升(Gradient boosting)的學習方法。屬於決策樹模型的一種。
- 有快速訓練與低記憶體需求的特點，並且可適用於分類以及迴歸任務。
- MAE約為3.528。

# 表面膜厚預測之應用案例

## Ch 6.10

<https://github.com/smartManufacturing/imbd2022>

## 表面膜厚預測\_2022年數據競賽題目

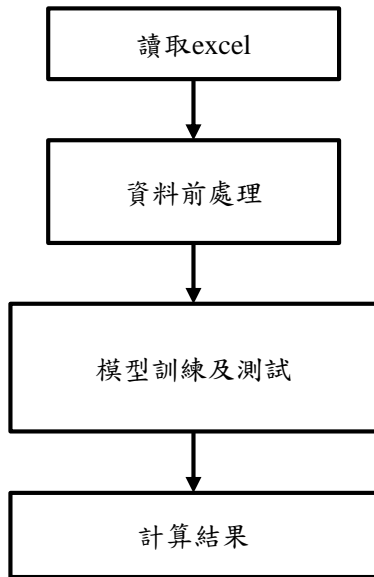
- 在許多加工五金產業中，鍍膜的厚度將大幅影響產品表面烤漆的品質。。
  - 針對製程參數加以控制鍍膜厚度為一項重要的需求。
  - 本範例針對不同的製程參數，以預測鍍膜厚度作為一範例。
  - 以支持向量迴歸(SVR)、隨機森林樹(Random Forest)、極限梯度提升(XGBoost)分別作為本範例預測膜厚的模型作為預測模型。
- 
- 問題屬性:迴歸分析



## 資料集描述

- 此範例共有3份Excel，分別為training.xlsx、testing.xlsx以及testing\_ans.xlsx
- training.xlsx為訓練資料，使用4個製程中設備之感測器量測數據預測工件表面之膜厚度，訓練資料欄位共131個欄位包含6個sensor為實際膜厚度以及125欄位為感測器量測數據。
- testing.xls為測試資料，共有125個欄位為感測器量測數據去預測膜厚度。
- testing\_ans.xlsx為testing.xlsx實際膜厚度，此份用於驗證模型預測的正確性。

# 方法流程



資料前處理包括處理遺失資料、分割輸入與輸出資料、分割訓練與測試資料、資料正規化等。

## □ *SVR*

- SVR模型主要建構在SVM模型的迴歸模型。
- SVR模型主要建構在SVM模型做衍伸，SVR也是需要找到一個基準決策平面。
- SVR模型是找到數據點距離決策平面的最小化，以得到最佳的基準決策平面。

## □ *RandomForest*

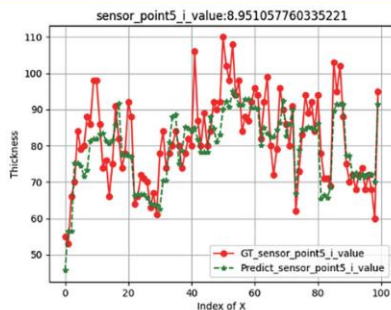
- 隨機森林樹主要為決策樹的延伸模型。
- 主要由多棵決策樹所構成的模型
- 因抽取出的每棵樹的特徵數量皆不同，因此最後以平均機制將決策樹的數量做平均。

## □ *XGBoost*

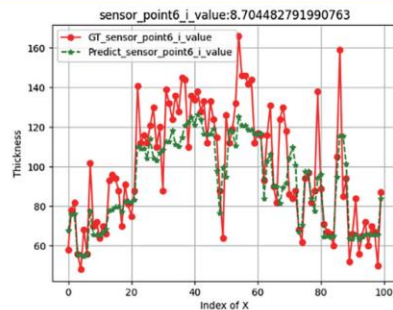
- 以 Gradient Boosting 為基礎下延伸，結合 Bagging 和 Boosting 的優點。
- XGboost 是採用特徵隨機採樣的技巧
- 採用L1/L2 Regularization續性數值的避免過度擬合。

# XGBoost測試結果

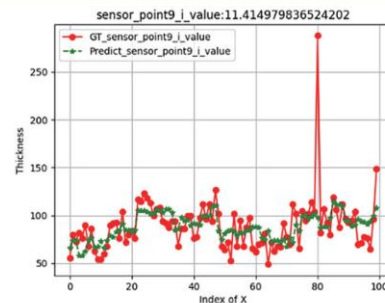
RMSE=8.9



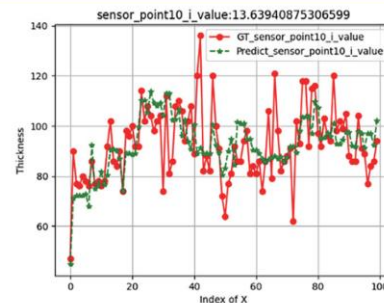
RMSE=8.7



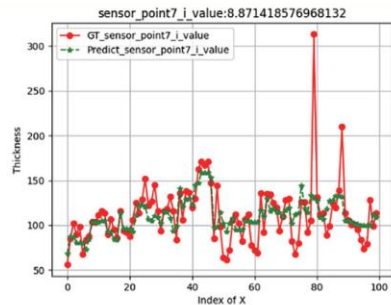
RMSE=11.4



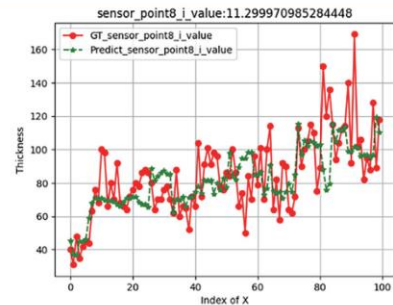
RMSE=13.6



RMSE=8.8



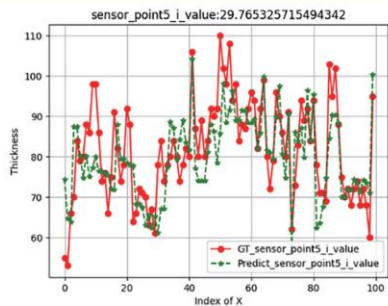
RMSE=11.2



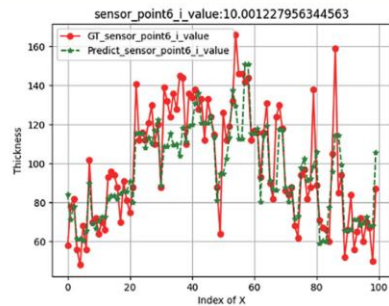
Average RMSE = 10.43

# SVM測試結果

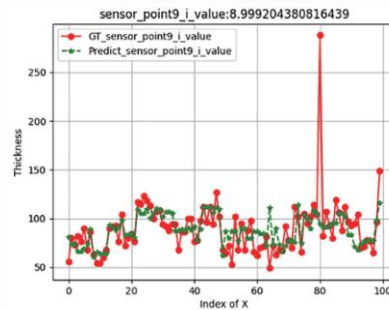
RMSE=29.7



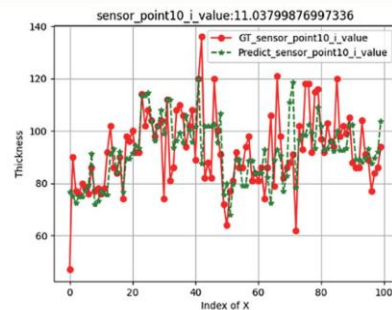
RMSE=10.0



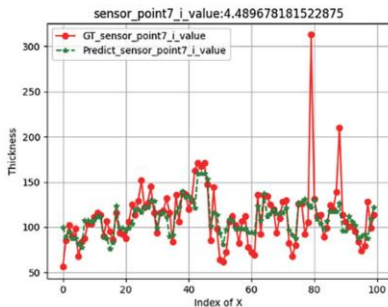
RMSE=8.9



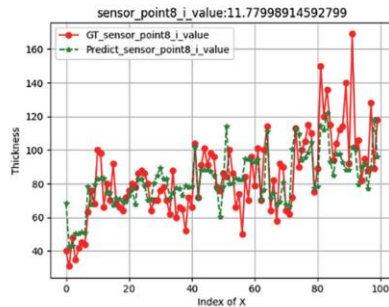
RMSE=11.03



RMSE=4.4



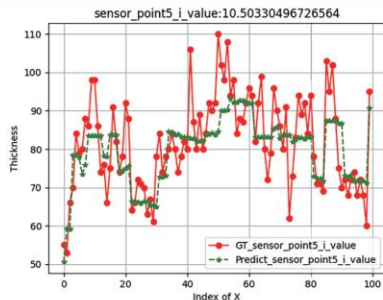
RMSE=11.7



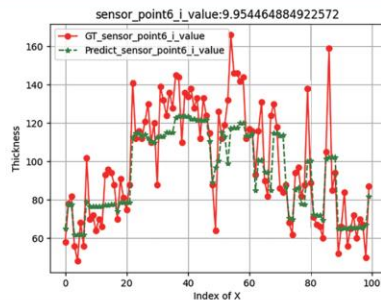
Average RMSE = 12.62

# RandomForest測試結果

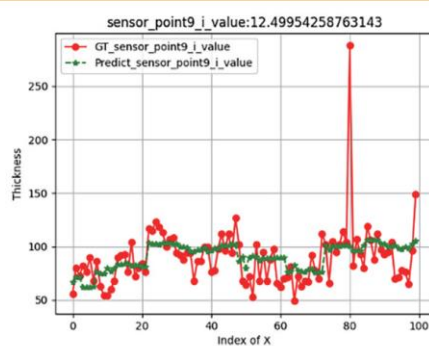
RMSE=10.5



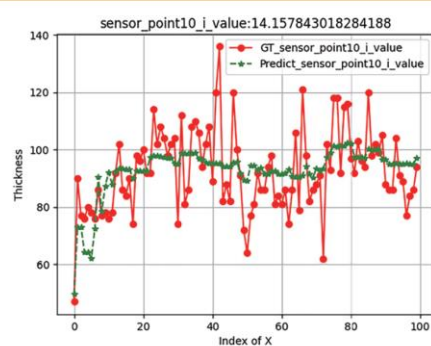
RMSE=9.9



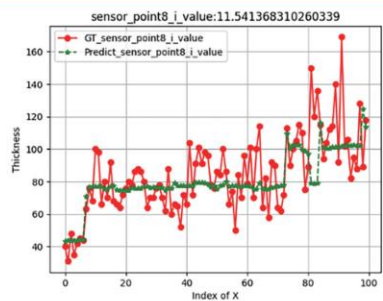
RMSE=12.4



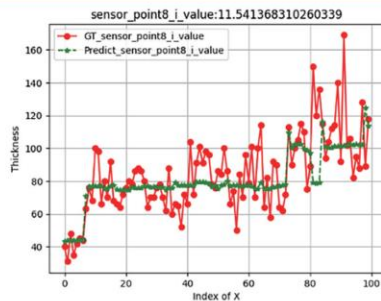
RMSE=14.15



RMSE=11.5



RMSE=11.5



Average RMSE = 11.66

# Thank You for Listening

## Q&A

**Chunhui Chung (鍾俊輝)**

**Associate Professor**

**Department of Mechanical Engineering**

**National Cheng Kung University**

**E-mail: [chchung@mail.ncku.edu.tw](mailto:chchung@mail.ncku.edu.tw)**