

# 感測網資料收集之案例分析

## Case Study of SensorThings Data Collection

國家災害防救科技中心

佐理研究員

研究員

周恆毅

張子瑩

Heng-Yi Chow

Tzu-yin Chang

### 摘 要

淹水災害長期以來都對台灣民眾造成強大的威脅，每每淹水發生，都會造成民眾大量的財務損失，因此本研究重新審視宜蘭、雲林、高雄等近期的淹水熱門地點，以深度學習模型建立地區是否淹水的分類模型，並由拆解這個深度學習模型來探究這些地點淹水的根本原因。本研究依據過去災情狀況與結果，建立告警機制，並以個案分析，對比災情與人工智慧預測模型之成果。

關鍵詞：淹水、模型、機器學習法、告警

### Abstract

Flooding disasters have long been a strong threat to the people of Taiwan. Every time a flood occurs, it will cause a lot of financial losses to the people. Therefore, this study re-examines recent flooding hotspots such as Yilan, Yunlin, and Kaohsiung, and uses a deep learning model to Build a classification model of whether an area is flooded, and disassemble this deep learning model to explore the root cause of flooding in these locations. This study establishes an alarm mechanism based on the past disaster situation and results, and analyzes the results of the disaster situation and the artificial intelligence prediction model based on a case study.

Keywords: flooding, model, machine learning, alarm

### 一、前言

近年來，物聯網系統隨著國土整體規劃及智慧城市建設蓬勃發展，因此各種環境感測、智慧交通、智慧醫療、水源監控和智慧建築等物聯網系統皆蓬勃發展。人工智慧(Artificial intelligence, AI) 與物聯網(Internet of Things, IoT) 的技術崛起，利用創新的 ICT 技術結合 IoT 監測設備，透過持續蒐集資料累積為大數據資料 (Big Data)，經由 AI 技術研發預測與分類模式，替政府與企業提供緊急應變決策，提供更

精準、快速即時的 AIoT 智慧防災資訊加值應用服務。面臨全球極端氣候的衝擊，防災科技的運用已成為世界趨勢，不僅能提昇救災決策的效率，更能提早針對防救災資源進行調度，從容應對災害之發生與變化。

本中心於 107 年即針對防救災相關資訊設計 OGC SensorThings API 標準化規範包括：雨量、水位、氣象、CCTV，並開發資料轉換程式和服務發布模組建立 API 介接平台，使跨單位的感測資料能於標準化服務格式交換，視覺化及空間化方式呈現各項感測資料，並於 108~110 年度持續擴充標準資料服務、產業應用推廣工作示範、運用人工智慧建立預測模型以及擴大數據資料彙集等工作。本計畫將依據數據預測結果建立告警機制，並依據過去災情狀況對比未來預測結果來建立人工智慧預測模型之個案分析。

## 二、感測網現況與架構

### 2.1 感測網架構

#### 2.1.1 系統架構

感測網透過政府開放資料服務 API 收集雨量、水位、氣象及水庫水位等資料，儲存於本中心之伺服器中，並安裝 AI 程式於 AI 專用伺服器以進行 AI 預測。透過網頁伺服器將各項感測資訊呈現於感測網前端頁面，以供使用者查詢觀看。同時將各項數據資料透過 OGC 伺服器進行轉換儲存，此經標準化之數據資料可作為未來開放給公眾使用。

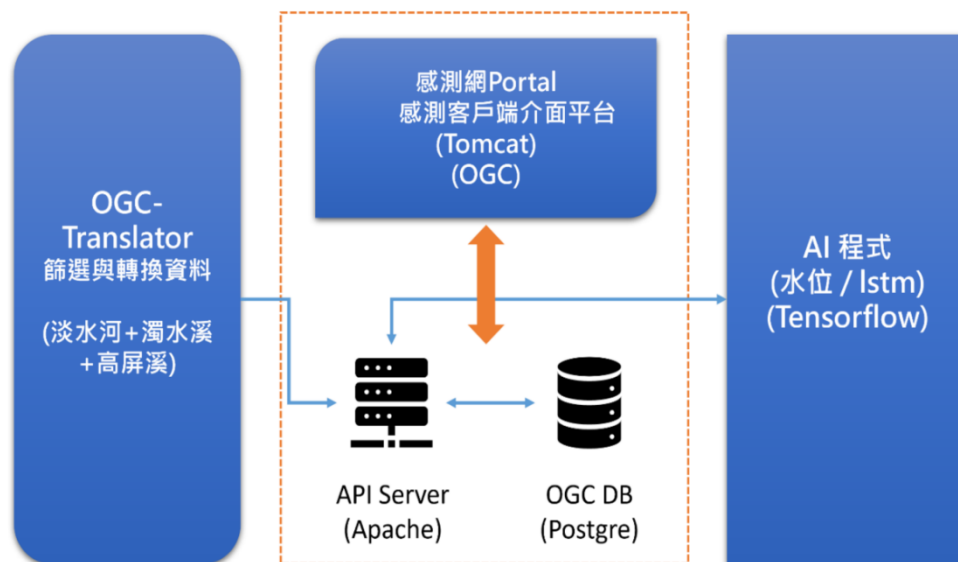


圖 1 感測網系統架構

#### 2.1.2 佈署架構

感測網整體系統佈署架構，包括資料轉換伺服器一台(Sensor AP)、感測網入口網站伺服器一台(Portal)、AI 模式伺服器一台(Sensor AI)如下圖所示。

## 感測網Deploy架構

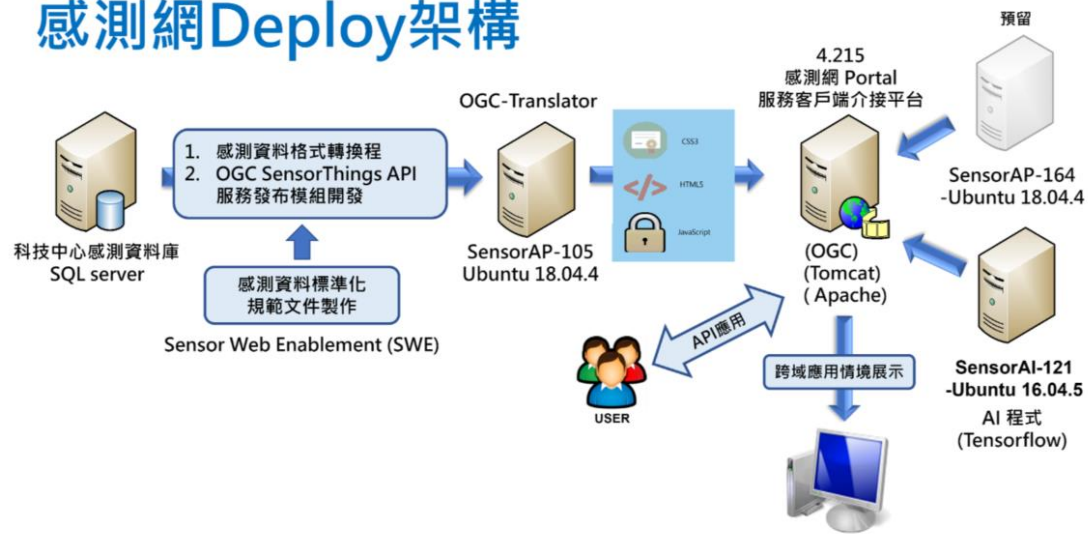


圖 2 感測網佈署架構

### 2.1.3 介面設計

感測網使用者登入後，即可進入感測網首頁，其前台介面設計，如下圖所示。

[https://sensorthings.ncdr.nat.gov.tw/NcdrApiCenter/frontend/Layer\\_platform](https://sensorthings.ncdr.nat.gov.tw/NcdrApiCenter/frontend/Layer_platform)

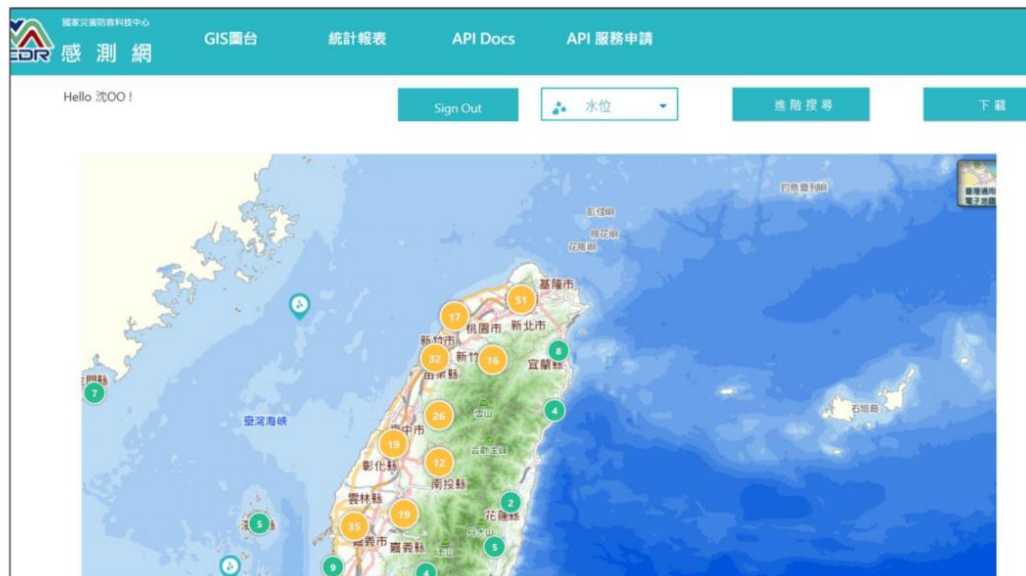


圖 3 感測網前台介面設計

## 三、個案分析

### 3.1 案例探討

淹水預報系統是透過即時資料擷取模組自動化接收即時雨量資料與淹水感測器即時資料，並結合人工智慧技術，進行淹水預報模式架構，滾動式進行即時修正，以提高

淹水預報準確度。本次以實際降雨事件之案例探討分析淹水預報模式與物聯網監測資料，進行前後結果比較。

首先針對淹水事件進行挑選檢核，挑選出的淹水區域範圍內淹水資料進行淹水點和水位站點篩選，以案例(1)北區 2022-0525 強降雨事件和案例(2)南台灣 2020-0522-0527 強降雨事件，作為探討案例。

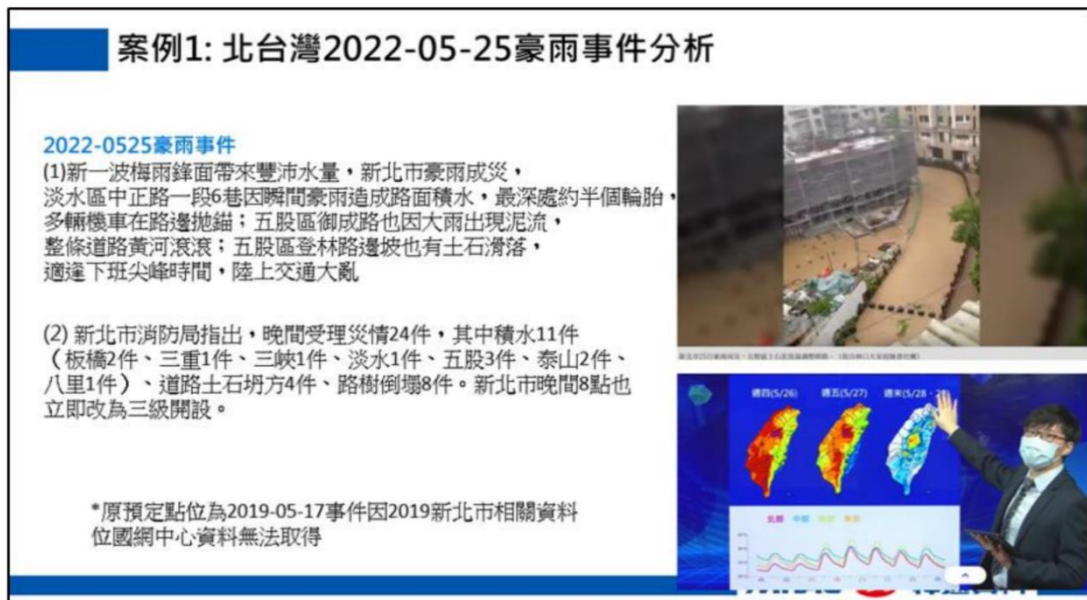


圖 3 案例 (1)北區 2022-0525 強降雨事件



圖 4 案例(2) 南台灣 2020-0522-0527 強降雨事件

資料處理之步驟如下所述：

1. 透過篩選出之淹水點，進行空間模式(SOM)之建模，確定出各集水區淹水範圍之

淹水情況。

2. 以各研究區域篩選出之淹水點為基準，建置時序性模式，並分析各研究區域內所有感測器應用於模式之適用性，並將 SOM 應用於感測器型態之探討。

3. 結合即時淹水感測器資料與 LSTM AI 預報模式，結合空間模式 SOM 結合時序模式，將可產生未來至少三小時之預報即時淹水預報結果。

4. 收集降雨事件資料，進行結果比對。

### 3.2 運算模組開發

本架構是整合回饋式非線性自回歸與競爭學習神經網路(SOM)，開發與設計如下。

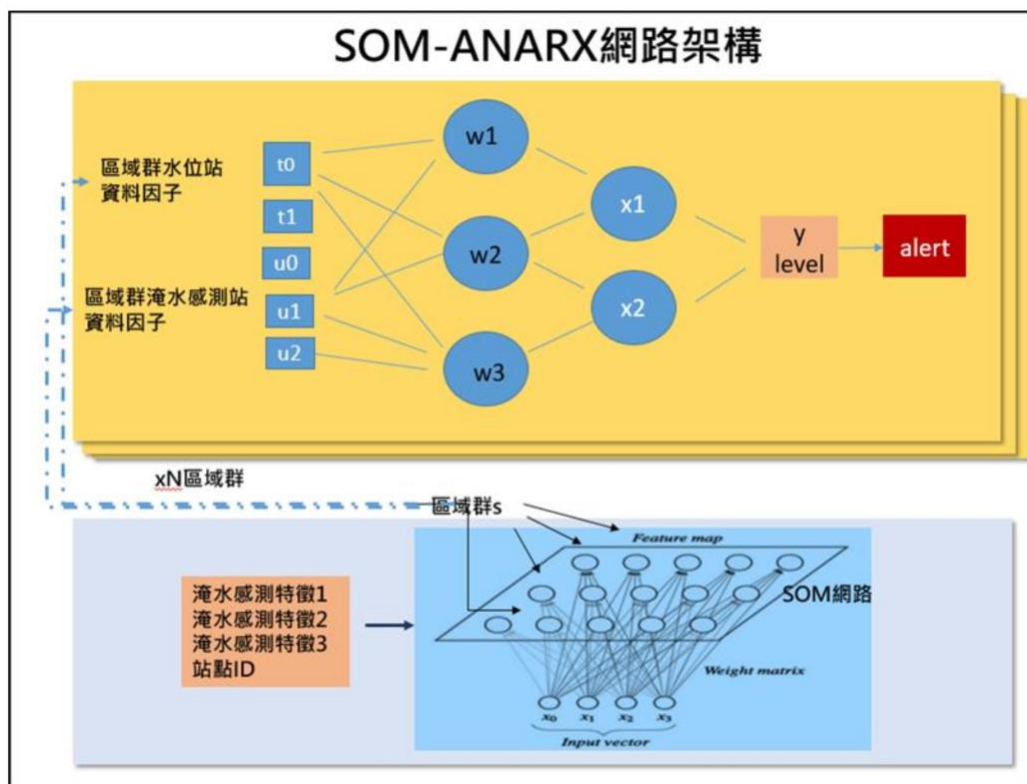


圖 5 整合回饋式非線性自回歸與競爭學習神經網路(SOM)

如上圖所示，整個流程分為兩個階段和四個程序，分別為：

(一)SOM 模型訓練

運用 Python 和 Jupyter 內含機器學習套件建構。

(二)SOM 分群

Python 搭配檢測 ML 和繪圖套件，建構不同感測器的特徵。

(三)各群模型訓練

NARX/LSTM 運用 Python 搭配 Tensorflow 套件。

(四)檢測和異常分析



Python 搭配基本數值套件。

### 3.3 運算案例探討與測試

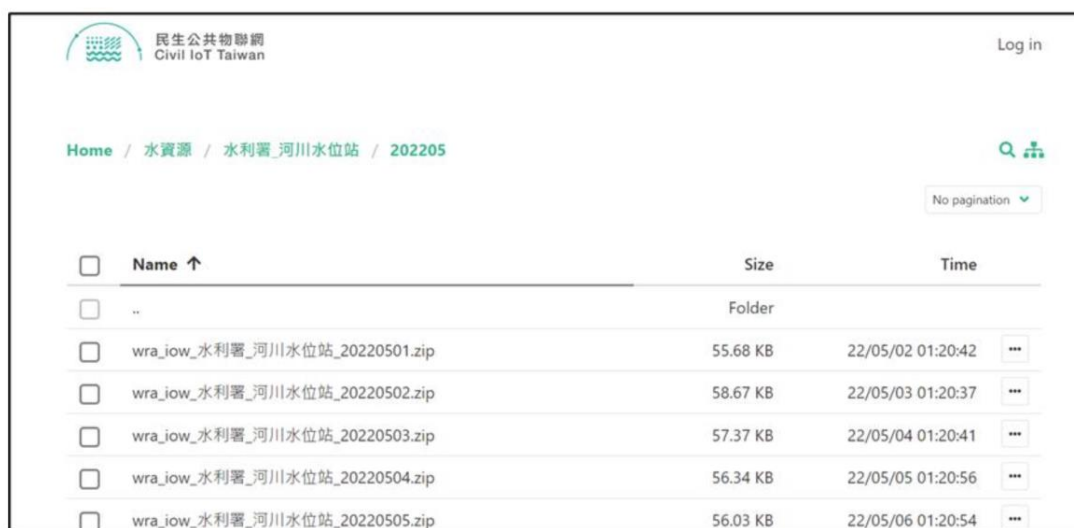
整合上述時序性淹水預報模式與淹水空間分布分類模式，以機器學習訓練並調整其網路架構以模擬該區域淹水之變化關係，再建置具有時序性與空間變化之自動化即時淹水預報模式，以進行未來 2 小時全區域之淹水預報。

首先建置之淹水空間分布分類模式(SOM 模式)，其次建立 NARX/LSTM 淹水時序神經網路預測模式，透過輸入及輸出之關係，模式將可自我學習並進行參數調整。

淹水空間分布分類模式根據資料為民生公共物聯網之淹水感測器紀錄之資料，進行 SOM 拓樸分類，利用各神經元與資料之間的關係，進行權重調整，進而演算出淹水區域拓樸；淹水時間序列分析模式則根據輸入之雨量資料與同一區之河川水位資料平均淹水深混合運算，此動態網路會將該時刻之預報輸出值回饋作為下時刻輸入因子，進行滾動式預報，以期輸出資料與模擬結果誤差最小。

以下敘述方法與試作過程：

#### (一)資料說明



The screenshot shows the 'Civil IoT Taiwan' web interface. The breadcrumb path is 'Home / 水資源 / 水利署\_河川水位站 / 202205'. A table lists five zip files with their names, sizes, and timestamps. Each row has a checkbox on the left and a three-dot menu on the right.

| <input type="checkbox"/> | Name ↑                         | Size     | Time              |     |
|--------------------------|--------------------------------|----------|-------------------|-----|
| <input type="checkbox"/> | ..                             | Folder   |                   |     |
| <input type="checkbox"/> | wra_iow_水利署_河川水位站_20220501.zip | 55.68 KB | 22/05/02 01:20:42 | ... |
| <input type="checkbox"/> | wra_iow_水利署_河川水位站_20220502.zip | 58.67 KB | 22/05/03 01:20:37 | ... |
| <input type="checkbox"/> | wra_iow_水利署_河川水位站_20220503.zip | 57.37 KB | 22/05/04 01:20:41 | ... |
| <input type="checkbox"/> | wra_iow_水利署_河川水位站_20220504.zip | 56.34 KB | 22/05/05 01:20:56 | ... |
| <input type="checkbox"/> | wra_iow_水利署_河川水位站_20220505.zip | 56.03 KB | 22/05/06 01:20:54 | ... |

圖 6 資料來源

[illegible]

圖 7 資料樣貌

本案例實驗期初規劃資料以淹水感測器歷史資料、雨量和水位站資料為主，後經實驗後發現水位資料相關性低，故整理各站點資料後模型刪除水位感測資料，以淹水感測器和雨量資料作為輸入。

## (二)SOM 模式

採取各淹水點的[高峰深度、高峰日期時間、經度、緯度、站點 id]向量作為輸入分布，運用簡單開發的 Python GUI 介面，進行 SOM5x5、6X6 與 7X7 個別分群：

Som x

5

Som y

5

Map type

toroid ▾

Grid shape

rectangular ▾

Run k-means clustering

☐

Clusters

2

Min

10

Max

Initializations

2

Output folder

C:\Users\cclin\AppData\Local\Temp\G

Show Advanced parameters

Run

Results

圖 8 5x5 格點 10 輪(epoch)運算設定

### (三)深度學習模式演算

首先進行點位的建模，選擇關注關鍵的點位(淹水感測和雨量感測器)之關注事件。



圖 9 案例 1-北臺灣 2022-0525 相關區域感測器



圖 10 資料前處理

對於不同型態的感測器，預期將以分類的方式再加以分別，加入將 SOM 分群應用於分類不同區位淹水型態討論和實作，此外加入的程序說明如下：

### 1. 案例一資料前處理

篩選水利署目前與縣市合作布建之感測器，並以事件附近日期之呈現觀察合適性，初始資料包含：

- (1) 疏洪六路/疏洪一路交叉路口
- (2) 疏洪一路 1k+300
- (3) 疏洪一路福德宮



(4) 疏洪八路/疏洪一路交叉口

(5) 長坑國小雨量站

等 5 處，經過資料關聯性前處理後，篩選出相關性高的標的：

● 疏洪六路/疏洪一路交叉口

● 疏洪八路/疏洪一路交叉口

● 長坑國小雨量站

其中淹水感測器皆為水利署（與縣市政府合建）之連續型淹水感測器。然後進入訓練模式：



圖 12 長坑國小雨量資料



圖 13 合併雨量資料

## 2. 訓練

經過資料整理和重取樣(Resample)，得到下列資料集：

總共 20220523-20220526 合併整理成共 500 筆，有 3 個維度資料(2 個淹水感測 + 一個雨量)，因淹水事件發生在 0525 左右，故把資料分為兩部分：

- train = merged\_data[0:400]-預設為正常區
- test = merged\_data[400:500] -預設為異常區

在資料量不多的情況下，訓練模型採 LSTM Autoencoder 模型，loss 函數是 MAE，參數總量為 1824，說明如下圖：

| Model: "model_6"                     |                |         |
|--------------------------------------|----------------|---------|
| Layer (type)                         | Output Shape   | Param # |
| =====                                |                |         |
| input_7 (InputLayer)                 | [(None, 1, 4)] | 0       |
| lstm_24 (LSTM)                       | (None, 1, 10)  | 600     |
| lstm_25 (LSTM)                       | (None, 5)      | 320     |
| repeat_vector_6 (RepeatVector)       | (None, 1, 5)   | 0       |
| lstm_26 (LSTM)                       | (None, 1, 5)   | 220     |
| lstm_27 (LSTM)                       | (None, 1, 10)  | 640     |
| time_distributed_6 (TimeDistributed) | (None, 1, 4)   | 44      |
| =====                                |                |         |
| Total params: 1,824                  |                |         |
| Trainable params: 1,824              |                |         |
| Non-trainable params: 0              |                |         |

圖 14 LSTM 參數

訓練過程如下，開始過程有 underfit，18 個 epoch 回合後 loss 和 val\_loss 有均衡回來收斂。

```

Epoch 8/20
76/76 [=====] - 0s 4ms/step - loss: 0.1720 - val_loss: 0.1131
Epoch 9/20
76/76 [=====] - 0s 4ms/step - loss: 0.1675 - val_loss: 0.1204
Epoch 10/20
76/76 [=====] - 0s 4ms/step - loss: 0.1623 - val_loss: 0.1220
Epoch 11/20
76/76 [=====] - 0s 5ms/step - loss: 0.1581 - val_loss: 0.1262
Epoch 12/20
76/76 [=====] - 0s 4ms/step - loss: 0.1536 - val_loss: 0.1281
Epoch 13/20
76/76 [=====] - 0s 4ms/step - loss: 0.1489 - val_loss: 0.1343
Epoch 14/20
76/76 [=====] - 0s 5ms/step - loss: 0.1450 - val_loss: 0.1353
Epoch 15/20
76/76 [=====] - 0s 4ms/step - loss: 0.1430 - val_loss: 0.1335
Epoch 16/20
76/76 [=====] - 0s 4ms/step - loss: 0.1400 - val_loss: 0.1327
Epoch 17/20
76/76 [=====] - 0s 5ms/step - loss: 0.1388 - val_loss: 0.1344
Epoch 18/20
76/76 [=====] - 0s 4ms/step - loss: 0.1377 - val_loss: 0.1315
Epoch 19/20
76/76 [=====] - 0s 4ms/step - loss: 0.1355 - val_loss: 0.1290
Epoch 20/20
76/76 [=====] - 0s 4ms/step - loss: 0.1347 - val_loss: 0.1301

```

圖 15 訓練過程一部分

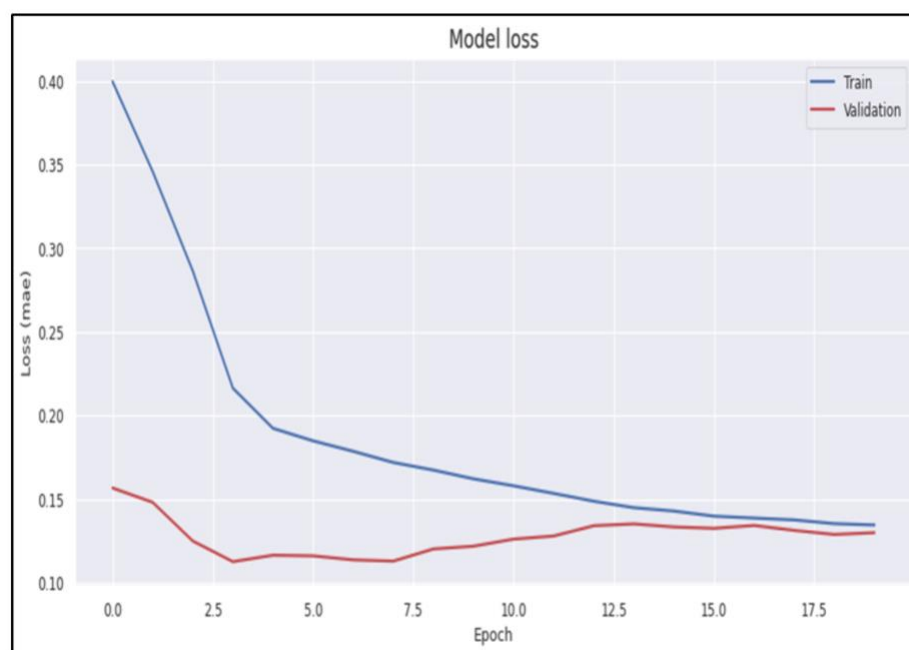


圖 16 訓練過程收斂情形

接下來將各資料點之 loss 圖畫出分布：

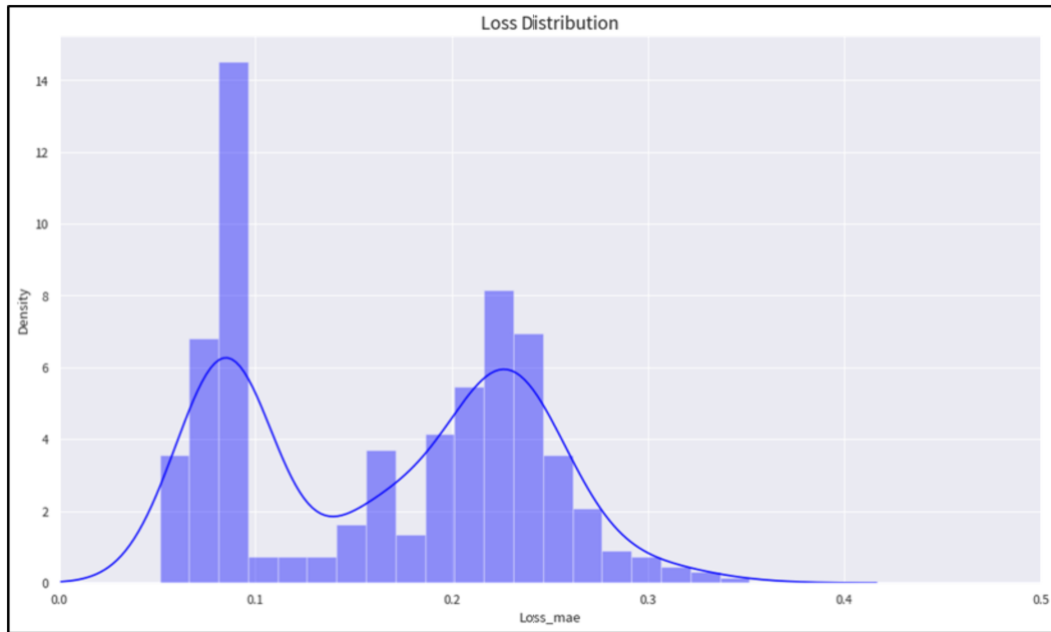


圖 17 Loss(MAE)之分布圖

可以觀察到呈現雙分布之情形，猜測是資料收集尚不足所造成，仍假設是以 normal 常態分布視之(0-0.423)。

### 3. 檢測方式

在獲得模型之後，將資料點的 loss 圖做篩選，得到  $\text{mean} = 0.166$ ,  $\text{Sigma} = 0.07$  初步將  $\text{loss} = 0.27$  設為 Threshold(約為 1.5 個 sigma, 約 86%的比例)當作閾值，可以得到如下之異常圖。

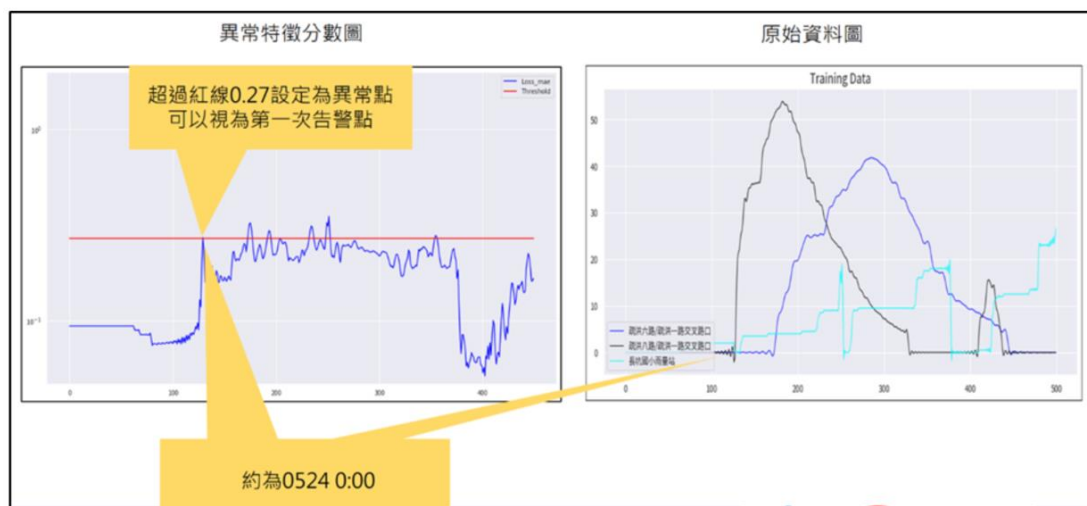


圖 18 常分布圖

觀察我們設定的閾值，可以看到第一個突破異常值點為 0524 00:00 左右，對於異常偵測的目的來說，這個時間點可以做為第一觀測和告警點，之後的突破告警點位可以考慮加強觀測。



#### 4. 應用到案例二

案例二資料前處理與案例一考慮相同，因台灣南北在 SOM 分布已經被分到不同區域，因此本案例將北部區域的模型方法，重訓練一次，觀察南區案例和感測器的表現。資料篩選水利署目前與縣市合作布建之感測器，並以事件附近日期之呈現觀察合適性，因資料可選擇不多，故選擇：

- 東港派出所淹水感測器

- 水利署東港雨量站

等 2 處，經過資料關聯性前處理後：

篩選其他合適感測器資料，例如水位和雨量感測數值



圖 19 例 2-南臺灣 2020-0522-0526 相關區域感測器

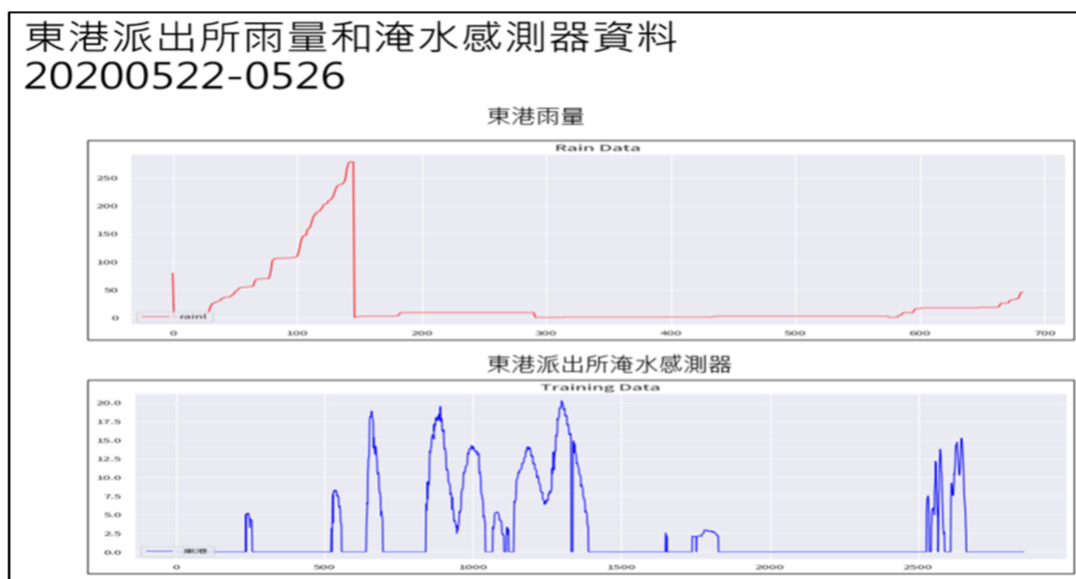


圖 20 南區案例資料整理

完成後並將以此模式作為預報或告警模型基礎，探討其誤差和效能相關因素，建立即時預報之功能。

經過一樣的資料轉換，資料量為：

```
train = merged_data[0:700]
```

```
test = merged_data[700:800]
```

訓練過程與北區相同，模型參數量不變。得到第二個模型，應用到 MAE loss 函數後，其分部如下圖。

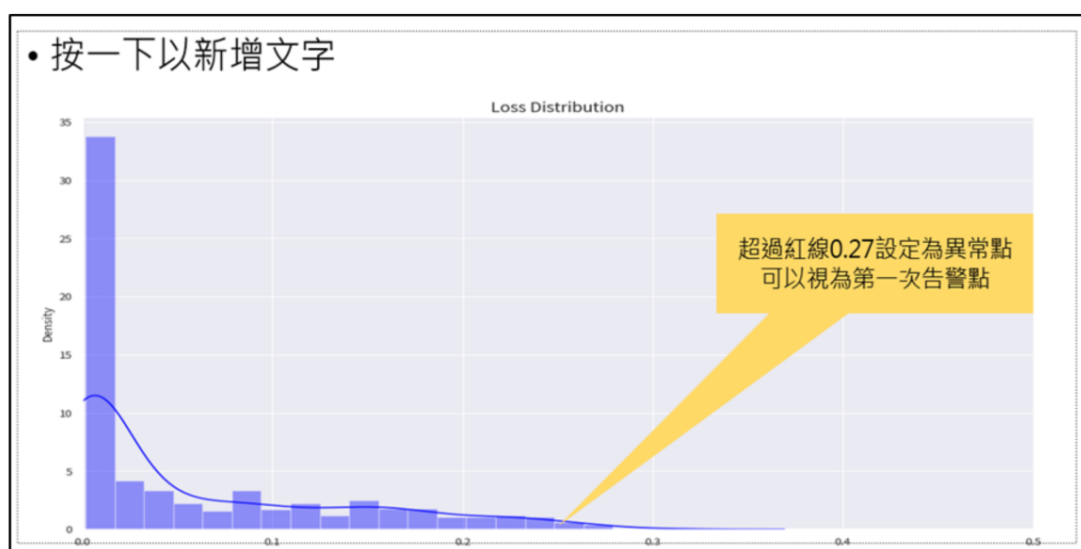


圖 21 東港派出所測站資料模型誤差分布(MAE)

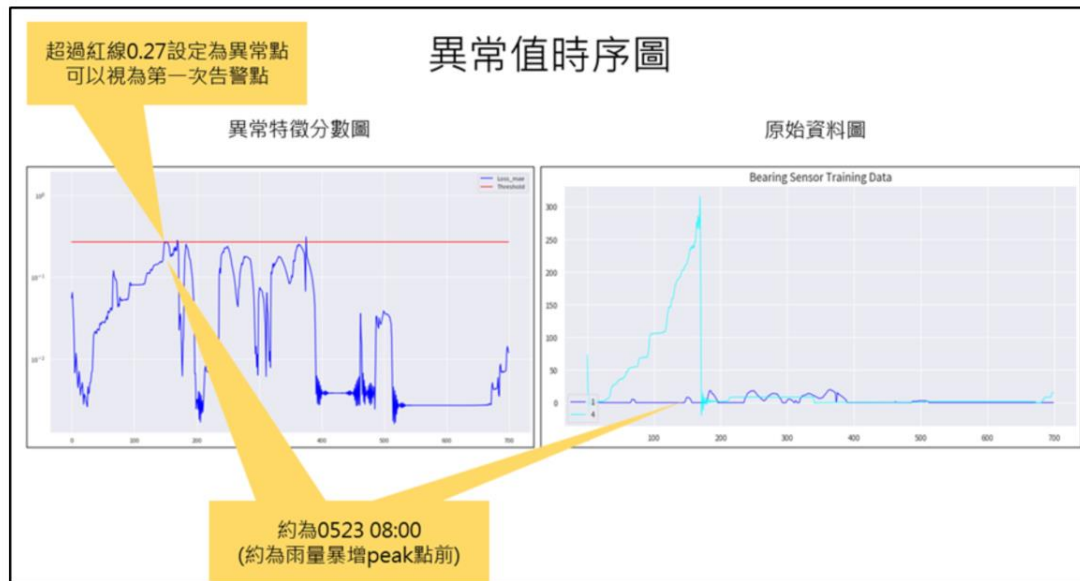


圖 22 異常時序圖

進行異常時序分析如下圖，雖然分布圖不相同，但仍舊設定一樣的 MAE 閾值(0.27)實驗，可以得到：

- (1) 第一次告警時間約為 0523 08:00，可以做為第一次參考發告警點。
- (2) 其後的告警點並不多，可以選擇做為持續觀察警戒點。

#### 四、檢討與結論

本年度的案例是經過收集、檢驗、比較不同區域的淹水感測器和雨量感測器的關聯，試圖藉由其關聯度進行深度學習與異常告警的分析，初步實驗的結論可以得到下列：

- (1) 感測資料部分收集困難，常常有淹水的新聞案例，感測器並沒有數值，或是數值變化太小，以至於未達到足夠大量的資料進行演算。因此未來資料量再加大，就可進行檢視和建模。
- (2) 初步檢視建模的可行性，不論是北區或南區，可以運用同一種建模方式，或許也可以嘗試使用同一個模型，建立不同地區的閾值當作告警依據。
- (3) 南區雖然只有兩個感測器，訓練模型仍能收斂，因此雨量和淹水感測有相關強的關聯性。
- (4) 未來建議在資料量夠的情況下，可以將目前的模型修改，擴展到進一步做更精確時間的異常預測方式(如 LSTM Autoencoder)，可以進一步驗證預測之功效。

## 參考文獻

- [1] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586 – 600, 2000. <https://ieeexplore.ieee.org/document/846731>
- [2] Application of Self-Organizing Maps on Time Series Data for identifying interpretable Driving Manoeuvres, *European Transport Research Review*(2020) 12:25
- [3] Y. Mohammad, K. Matsumoto, and K. Hoashi, "Key feature learning and selection for activity recognition," *Proceeding on Annual ACM Symposium on Applied Computing*, pp. 930–939, 2018.
- [4] Y. C. Chen and D. C. Li, "Selection of key features for PM2.5 prediction using a wavelet model and RBF-LSTM," *Applied Intelligence*, vol. 51, pp. 2534–2555, 2021.
- [5] Yi-Chung Chen, Sheng-Min Chiu, Louis, Cheng-Ju Kuo, Li-Chun Hung, Min-Hsiung Hung, and Chao-Chun Chen, "Using Fuzzified Deep Recurrent Neural Networks to Retrieve Deep Features to Enhance the Accuracy of Tool Wear Prediction," *IEEE International Conference on Automation Science and Engineering (CASE)*, 2019–08.
- [6] 賴紫平，使用深度學習模型結合二維高斯函數探討交通站點之用量與成因，國立雲林科技大學工業工程與管理系碩士論文，2020。
- [7] 江吉晟，應用三維高斯函數結合深度學習之區域PM2.5預測，國立雲林科技大學工業工程與管理系碩士論文，2020。