

國立中央大學

資訊工程研究所

碩士論文

空氣品質感測器之故障偵測-基於深度時空圖模型的異常偵測框架

Detecting Malfunctioned Air Quality Sensors – an
Anomaly Detecting Framework Base on Deep
Spatial-Temporal Graph Model

研究生：林子恆

指導教授：陳弘軒 博士

中華民國一百一十年六月

國立中央大學圖書館學位論文授權書

填單日期：110/8/5

2019.9 版

授權人姓名	林子恆	學號	108522072
系所名稱	資訊工程學系	學位類別	<input checked="" type="checkbox"/> 碩士 <input type="checkbox"/> 博士
論文名稱	空氣品質感測器之故障偵測—基於深度時空圖模型之異常偵測框架	指導教授	陳弘軒

學位論文網路公開授權

授權本人撰寫之學位論文全文電子檔：

- 在「國立中央大學圖書館博碩士論文系統」，
 - ☒同意立即網路公開
 - ☐同意 於西元_____年_____月_____日網路公開
 - ☐不同意網路公開，原因是：_____
- 在國家圖書館「臺灣博碩士論文知識加值系統」
 - ☒同意立即網路公開
 - ☐同意 於西元_____年_____月_____日網路公開
 - ☐不同意網路公開，原因是：_____

依著作權法規定，非專屬、無償授權國立中央大學、台灣聯合大學系統與國家圖書館，不限地域、時間與次數，以文件、錄影帶、錄音帶、光碟、微縮、數位化或以其他方式將上列授權標的基於非營利目的進行重製。

學位論文紙本延後公開申請 (紙本學位論文立即公開者此欄免填)

本人撰寫之學位論文紙本因以下原因將延後公開

- 延後原因
 - ☐已申請專利並檢附證明，專利申請案號：_____
 - ☐準備以上列論文投稿期刊
 - ☐涉國家機密
 - ☐依法不得提供，請說明：_____

• 公開日期：西元_____年_____月_____日

※繳交教務處註冊組之紙本論文(送繳國家圖書館)若不立即公開，請加填「國家圖書館學位論文延後公開申請書」

研究生簽名：林子恆

指導教授簽名：陳弘軒

國立中央大學碩士班研究生

論文指導教授推薦書

資訊工程碩士班 學系/研究所 林子恆 研究生所

提之論文 空氣品質感測器之故障偵測--基於深度時

空圖模型的異常偵測框架

係由本人指導撰述，同意提付審查。

指導教授 陳弘毅 (簽章)

2021年7月8日

國立中央大學碩士班研究生
論文口試委員審定書

資訊工程學系碩士班 學系/研究所 林子恆 研究生

所提之論文 空氣品質感測器之故障偵測--基於深度時空圖模型的
異常偵測框架

經由委員會審議，認定符合碩士資格標準。

學位考試委員會召集人

委

員

陳弘志

孫弘志

陳弘志

中 華 民 國

110 年 7 月 26 日

1100726

空氣品質感測器之故障偵測-基於深度時空圖模型的異常偵測框架

摘要

近年來空氣汙染已成為重視的議題。台灣行政院環保署規劃，於 2017 年至 2020 年逐步布建 1 萬 200 個空氣品質感測器，由物聯網方式蒐集大量且時間解析度密集的空品數據來分析各地區的空氣污染現象。

由於空氣品質感測器本體小且脆弱，容易受到日曬、雨淋或是其他物理因素使得裝置故障而偵測到錯誤的 $\text{pm}_{2.5}$ 數值，而目前的空氣品質感測器故障偵測方式仰賴現場做檢閱測量。但感測器佈設點過多，以抽檢方式做巡檢是沒有效率的巡檢方式。

我們建構一個基於深度時空圖模型的異常偵測框架，模型包含圖卷積和時間卷積兩個方法，時間卷積處理過往空氣品質感測器偵測到的 $\text{pm}_{2.5}$ 時序資料，找出時間上的相關性，同時於每個時間點，對感測器與其周圍感測器點建立成一個無向聯通圖，由圖卷積的方式，找出空間特徵，最後模型預測當下的 $\text{pm}_{2.5}$ 數值，並與空氣品質感測器偵測到的數值計算 $R^2\text{-score}$ 分數，並對所有空氣品質感測器站點的 $R^2\text{-score}$ 做低到高排序， $R^2\text{-score}$ 越低代表感測器越有可能出現異常。經過實驗，在有限巡檢個數之下，我們模型找到有問題個數的比率跟過往異常偵測的方法和隨機挑選都來的高，優先檢查本模型認定的故障感測器可降低巡檢所需的人力與時間成本。

關鍵字：圖卷積網路、時間卷積網路、時序異常偵測、深度學習

Detecting Malfunctioned Air Quality Sensors – an Anomaly Detecting Framework Base on Deep Spatial-Temporal Graph Model

Abstract

Air pollution is an essential issue in Taiwan. From 2017 to 2020, the Environmental Protection Administration of Executive Yuan in Taiwan has gradually deployed 10,200 air quality sensors, which have collected a large amount of air quality data with high time resolution. However, because the air quality sensors are small and fragile, environmental factors, such as sun, rain, and other physical characteristics, may influence or even damage these sensors. As a result, the monitored PM2.5 values from these malfunctioning sensors are inaccurate. The current strategy to identify the failure sensors requires on-site inspection. However, since the number of sensors is huge and sensors are located all over Taiwan, it is inefficient to discover the malfunctioning sensors by scheduled inspection or random sampling.

We propose an anomaly detection framework to identify the malfunctioning sensors based on a deep spatial-temporal graph model consisting of graph convolution and time convolution.

The time convolution discovers the temporal relationship among the monitored PM2.5 values of a sensor. At each time point, an undirected

connected graph is established between each sensor and its surrounding sensors. The graph convolution utilizes these graphs to learn the spatial characteristics. We leverage this deep spatial-temporal graph model to predict the current PM2.5 value of a target sensor. We calculate the R2-score between the predicted PM2.5 values and the monitored PM2.5 values for each sensor and rank these sensors by their corresponding R2-scores in ascending order. We claim a sensor has malfunctioned if the R2-score is low (i.e., the predicted and the monitored PM2.5 values are very different). Experimental results show that our model identifies more problematic sensors with fewer trials. Consequently, examining the sensors with the order outputted by our model can save labor and time costs.

Keywords: Graph Convolution Network, Temporal Convolution Network, Time Series Anomaly Detection, Deep Learning

目錄

	頁次
摘要	iv
Abstract	v
目錄	vii
圖目錄	x
表目錄	xii
一、緒論	1
1.1 研究動機	1
1.2 研究貢獻	3
1.3 論文架構	3
二、相關研究	5
2.1 統計與規則基礎方法	5
2.2 機器學習方法運用在空氣品質預測	5
2.3 深度時空圖模型方法介紹	6
三、研究方法與流程	8
3.1 產生圖序列資料	9
3.1.1 設定圖形結構中的邊權重	9
3.2 深度時空圖模型	10
3.2.1 圖卷積 Graph Convolution.....	10

3.2.2	時間卷積 Temporal Convolution	16
3.2.3	模型架構	17
3.2.4	STGCN 模型介紹	19
3.3	異常偵測模組	20
四、	實驗結果	22
4.1	實驗流程與細節	22
4.1.1	空汙感測器資料集介紹	22
4.1.2	資料清洗	23
4.1.3	切割訓練資料與測試資料	23
4.1.4	選取周圍感測器個數	23
4.1.5	時間滾動產生訓練特徵與標籤	24
4.1.6	產生圖形時序資料	25
4.1.7	模型訓練	25
4.1.8	異常偵測模組	30
4.2	實驗結果	30
4.2.1	評量指標	30
4.2.2	ROC curve 分數結果	31
4.2.3	前 10 至 50 個感測器其精確率	32
4.2.4	前 10 至 50 個感測器其召回率	33
4.2.5	模型分離與模型統一結果比較	34
4.2.6	全部對全部與多對一結果比較	36
4.2.7	PR curve 分數結果	38
4.2.8	正常感測器與異常感測器其巡檢當日的 R2-score 平 均值	38
4.3	實驗環境介紹	40

五、	結論與未來展望	41
5.1	結論	41
5.2	未來展望	42
	參考文獻	44
	附錄 A 實驗程式碼	47

圖目錄

	頁次
1.1 空氣品質感測器	2
2.1 時空資料示意圖	6
3.1 感測器異常偵測架構圖	8
3.2 時間序列轉圖時序範例示意圖	9
3.3 權重拉普拉斯矩陣計算流程	11
3.4 聚合示意圖	15
3.5 時間卷積示意圖	17
3.6 模型架構 [1]	18
3.7 STGCN 架構圖 [2]	20
3.8 異常偵測模組流程圖	21
4.1 實驗流程圖	23
4.2 中心感測器站點與周圍感測器站點群平均距離箱型分佈圖	24
4.3 時間滾動產生訓練資料。 C_i 代表中心節點，藍色方形代表中心節點的時序值，而 $N_1^{C_i}$, $N_2^{C_i}$ $N_5^{C_i}$ 則代表與 C_i 最相近的 5 個節點。	26

4.4	模型分離訓練方式。每個框線裡的圖形資料代表以一個感測器為中心點，和最近的 5 個感測器建構成一個圖序列資料，因此共有 144 個框線資料，每個框線資料都會由一個模型去訓練，因此共訓練出 144 個模型，144 個都是獨立的模型，其參數都是先隨機初始化之後再根據各自的框線訓練資料進行訓練。	27
4.5	模型統一訓練方式。每個藍色框線裡的圖形資料代表以一個感測器為中心點，和最近的 5 個感測器建構成一個圖序列資料，因此共有 144 個藍色框線資料，圖形結構會根據不同中心點而有不同的邊權重，由 1 個模型訓練 144 個藍色框線資料。	28
4.6	時間滾動產生全部對全部訓練資料	29

表目錄

	頁次
4.1 不同訓練方式總結	30
4.2 不同訓練模型經由異常偵測模組算出其 ROC 分數	32
4.3 機器學習方法前 10 至 50 個感測器其找到有問題感測器精 準率	33
4.4 機器學習方法前 10 至 50 個感測器其找到有問題感測器召 回率	34
4.5 模型分離與模型統一結果比較	35
4.6 模型分離與模型統一前 10 至 50 個感測器其找到有問題感 測器精準率比較	35
4.7 模型分離與模型統一前 10 至 50 個感測器其找到有問題感 測器召回率比較	35
4.8 全部對全部與多對一其 ROC 分數比較	37
4.9 全部對全部與多對一前 10 至 50 個感測器其找到有問題感 測器精準率比較	37
4.10 全部對全部與多對一前 10 至 50 個感測器其找到有問題感 測器精準率比較	37
4.11 不同訓練模型經由異常偵測模組算出其 PR curve 分數	39
4.12 正常感測器與異常感測器其巡檢當日的 R2-score 平均值	39
5.1 不同模型訓練方式可以運用的的場景	42

一、緒論

1.1 研究動機

隨者工業化發展，空氣汙染一直是近幾年重要的環境議題，為了知道生活周遭是否有空氣汙染的現象，空氣品質感測器扮演一個重要的角色，其主要藉由「節點」反映小尺度空間內空氣品質及時情形，並由 24 小時監測空氣品質，掌握汙染源熱區。台灣行政院環保署於 2017 年至 2020 年逐步布建 1 萬 200 個感測器 [3]，空氣品質感測器結構如圖1.1所示 [4]，其由物聯網方式蒐集大量且時間解析度密集的空品數據來分析各地區是空氣汙染現象。

上述的空汙感測器是以物理光散射原理來偵測空氣中的 $\text{pm}_{2.5}$ 濃度，而感測光源的裝置容易受到日曬、雨淋或是其他物理因素，造成偵測的 $\text{pm}_{2.5}$ 數值出現偏差，因此需要定期與標準儀器進行校正維護，確保其監測穩定度。但由於感測器部署於台灣各地區域非常廣，而巡檢的人力成本有限，往往只能隨機挑選小部分的感測器做巡檢校正，但這可能造成抽取的感測器都屬於正常，異常的感測器反而沒有巡檢到，是沒有效率的尋檢方式。

本篇論文的異常指的是感測器裝置其偵測 $\text{pm}_{2.5}$ 的物理光散射儀器因為外在因素而出現故障，因此其偵測到的 $\text{pm}_{2.5}$ 數值出現問題。我們目標是要有問題的 $\text{pm}_{2.5}$ 數值，藉此來判斷感測器為異常。

我們設計一個異常偵測框架針對感測器過往的時序資料以及感測器與感測器之間的空間資訊來找出可能異常的感測器，藉此來降低巡檢的

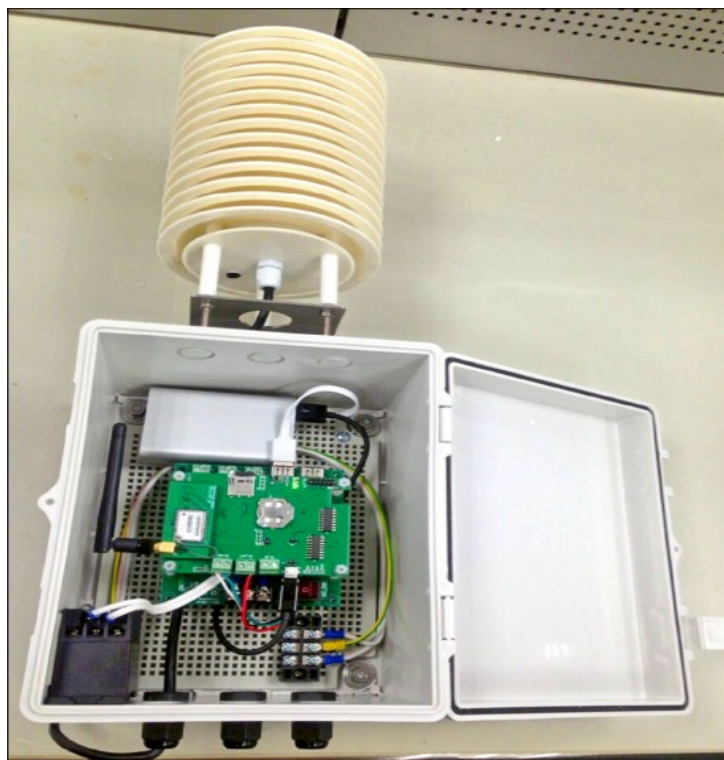


圖 1.1: 空氣品質感測器

人力成本。空氣品質感測器偵測到的 $\text{pm}_{2.5}$ 時序資料，是非線性的高維度分布， $\text{pm}_{2.5}$ 數值的變化以時間觀點中，其容易受到季節、季風風向等影響，空間觀點中，其受到所在地點以及附近是否有污染源影響。其複雜性我們認為以統計或規則方法是無法有效掌握其分布的，我們傾向以機器學習方法訓練一個模型從過往感測器偵測到的長時間 $\text{pm}_{2.5}$ 時序資料，以及每個時間點中，周圍感測器偵測到的 $\text{pm}_{2.5}$ 值形成的空間資訊，來找到時空相關性，一旦感測器出現異常，模型可以偵測到異常 $\text{pm}_{2.5}$ 數值來判斷其為異常感測器。

我們採用的機器學習訓練方式是以回歸分析預測接下來 $\text{pm}_{2.5}$ 的數值，我們會取過往前 30 分鐘 $\text{pm}_{2.5}$ 數值當成特徵，當下 $\text{pm}_{2.5}$ 數值當成目標做回歸訓練，並預測感測器接下來的 $\text{pm}_{2.5}$ 數值。預測的 $\text{pm}_{2.5}$ 數值會和感測器實際偵測到的數值計算 $R^2\text{-score}$ [5]， $R^2\text{-score}$ 越低代表感測器可能越有問題。

而我們不採用二元分類的訓練方式在於，儘管我們知道此感測器出

現異常，但我們不知道其是從哪個時間點開始發生異常。舉例來說假設有個感測器於 7 月 1 號被巡檢人員發現異常，但其實早在 5 月 1 號就出現故障了，我們應該把 1/1 號 (起始時間) 到 5 月 1 號前的標籤設為正常，而 5 月 1 號到 7 月 1 號的標籤設為異常，但問題是我們不知道 5 月 1 號這個時間，如果我們把 1/1 號 (起始時間) 到 7 月 1 號的標籤設為異常，那資料就不會正確，異常偵測也會出現問題，因此相對於二元分類的訓練方式，我們更傾向上述回歸分析預測方式。

1.2 研究貢獻

本篇論文的具體貢獻總結如下：

- 建構一個基於深度時空圖模型的異常偵測框架，模型包含圖卷積和時間卷積兩個方法，再由異常偵測模組找出可能出現異常感測器。
- 比較不同的圖形結構對異常偵測框架所造成影響，以及模型不同的訓練方式對偵測異常的精確率的好壞。我們建構的異常偵測框架與其他過往提出的序列機器學習模型相比其結果都來的好，同時比起統計規則方法準確率都來的高。
- 根據各種模型的實驗結果及感測器的佈建狀況，我們建議了實務上各種情況可考慮採用的異常偵測模型。

1.3 論文架構

本篇論文分五個章節，架構如下：

第一章、說明研究動機與研究貢獻。

第二章、介紹過往提出異常偵測框架方法及機器學習方法運用在空氣品質預測方面。

第三章、介紹基於深度時空圖模型的異常偵測框架裡的詳細方法與模型細節。

第四章、介紹實驗所使用的資料與實驗流程以及探討不同實驗方法其結果差異。

第五章、我們會總結本研究至今為止的發現並且未來可以改進的方向做討論。

二、 相關研究

本章介紹過去研究中偵測可能有問題的感測器方法，並介紹深度圖模型的應用研究。

2.1 統計與規則基礎方法

過往 L.-J. Chen 等人 [6] 基於統計基礎 (Statistics-Based) 和規則基礎 (Rule-based) 方式來找出有問題的感測器，其經由上述兩個方法對感測器定義了空間異常 (spatial anomaly)，時間異常 (temporal anomaly)，時空異常 (spatial-temporal anomaly) 三種狀態，並經由感測器排序模組 (Device Ranking module)，來從感測器分布群中找出異常的裝置。但裝置的時序資料及空間資料往往是非線性的高維度分布，複雜度極高，統計基礎或規則基礎方法可能難以捕捉到感測器的時空資訊分佈來偵測出異常數值。

2.2 機器學習方法運用在空氣品質預測

我們希望經由深度學習技術，從感測器過往長時間時空序列資料中訓練出模型，再經由此模型針對當下感測器偵測到的數值做異常偵測。過往 Z. Qi 等人提出以深度神經網路 (Deep Neural Network) 加上相對熵 (Kullback Leibler divergence) 方法，從過往的時序及空間資料進行學習來預測北京空氣品質 [7]。Y.-S. Chang 等人提出以 LSTM[8] 為基礎的

聚合模型來做空汙數值預測 [9]。上述兩個方法，雖然可以把空間資料放進模型裡做學習，但每個感測器和感測器之間是有距離的，距離的長短會決定空間數據對感測器的影響力，但前兩者無法把距離這項特徵考慮進去。

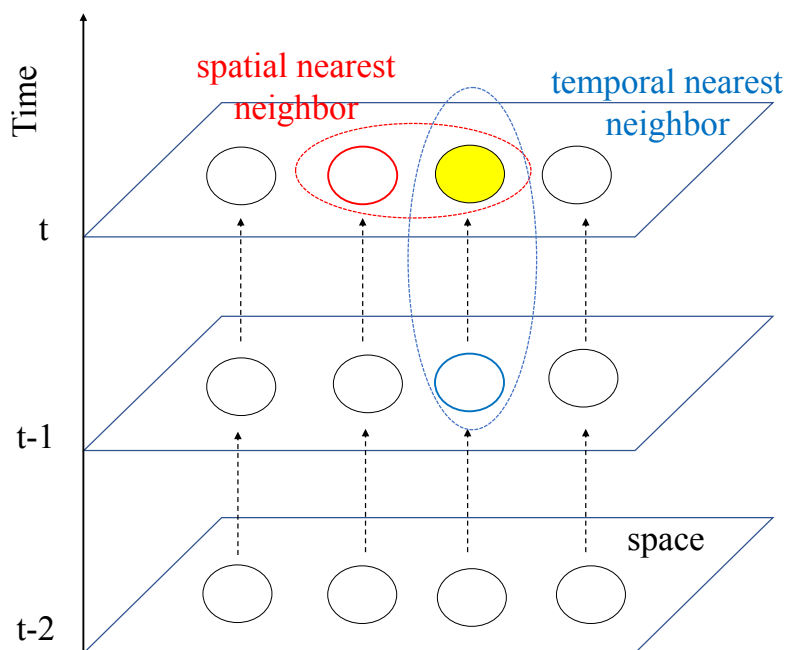


圖 2.1: 時空資料示意圖

2.3 深度時空圖模型方法介紹

Yaguang Li 等人於 2017 把深度時空圖模型運用在交通流量預測上 [10]，其把交通流量偵測器視為一個節點 (node)，並把多個交通流量偵測器建成圖形結構，每個節點都有交通流量時序資料，其時空圖模型結合圖卷積和 Gated Recurrent Unit[11] 來找到交通流量的時間空間相關性。而後 B.Yu 等人結合頻域基礎的圖卷積和因果卷積 (Causal Convolution) [12] 方法運用在相同交通流量資料集中，其結果比前者來的好 [2]。Wu 等人於則是結合圖卷積和時間卷積取得比前兩者更好結果 [1]，其圖卷積是基於 Yaguang Li 等人提出的方法在加入自適應鄰接矩陣 (self-adaptive

adjacency matrix), 時間卷積則是結合因果卷積 (Causal Convolution) [12] 與擴張卷積 (Dilated Convolution) [13]。

我們認為空氣品質感測器偵測到的 pm2.5 值是有時空相關性 (圖2.1), 因此我視感測器為一個節點 (node), 而節點與節點之間的邊權重 (edge weight) 會根據兩個節點的距離來設定, 而深度時空圖模型是使用 Wu 等人提出方法 [1], 再加上異常偵測模組來找出異常感測器, 我們會在下個章節介紹邊權重設定方式, 深度時空圖模型和異常偵測模組的詳細內容。

三、 研究方法與流程

在本章節中，我們會從頭開始建立一個以深度學習為基礎的時序異常框架，此框架包含兩個主要部分：深度時空圖模型和異常偵測模組。我們提出的時序異常框架如圖3.1所示，整體流程為 (1) 產生圖序列資料、(2) 深度時空圖模型訓練 (3) 異常偵測模組找出有問題感測器。接下來我們針對此三個實作進行介紹。

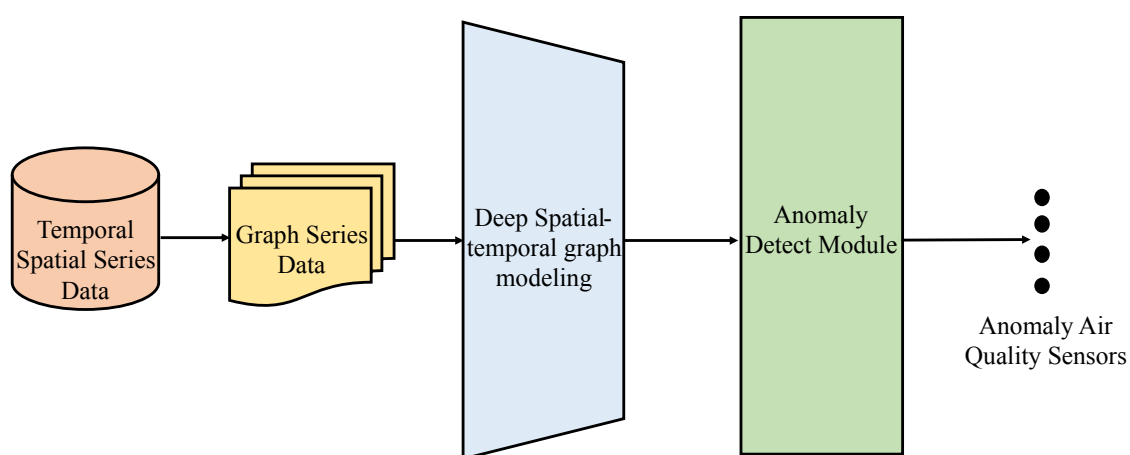


圖 3.1: 感測器異常偵測架構圖

3.1 產生圖序列資料

針對感測器的時序資料，我們會把他轉換成圖形結構的序列資料。轉換的方法為：每個感測器分別視為一個節點 (node)，感測器的 pm2.5 時序值則視為節點的動態輸入特徵 (dynamic input features)，而每個節點與節點之間都有邊 (edge) 相連，形成一個無向全連接圖 (fully connected graph)。圖3.2為一個例子，假設有 6 個感測器時序資料，每個感測器的時序大小為 10，我們會把他轉換成一個有 6 個節點的無向全連接圖形結構，而每個節點的輸入特徵，就是其感測器偵測到的 pm2.5 值，形成共 10 個的圖時序資料。

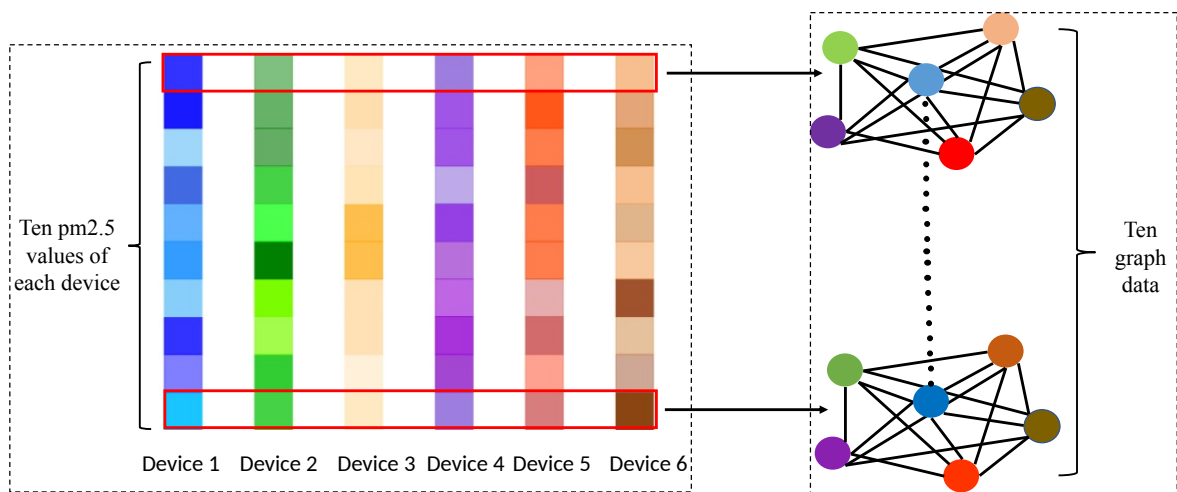


圖 3.2: 時間序列轉圖時序範例示意圖

3.1.1 設定圖形結構中的邊權重

建立完無向全連接圖後，我們要設定圖形結構中的邊 (edge) 其權重，權重值的定義如下公式 1:

$$W_{ij} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) & , i \neq j \text{ and } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \epsilon \\ 0 & , \text{otherwise.} \end{cases} \quad (1)$$

W_{ij} 代表節點 i 與節點 j 的邊權重，而 d_{ij} 代表站點 i 與站點 j 實際距離，距離的算法是以歐氏距離 (euclidean distance) 計算的 [14]，並以公里為單位，而 σ^2 和 ϵ 則代表 threshold 值，我們從 validation set 決定將它們分別設定為 10 和 0.5，也代表說站點 i 與站點 j 距離超過 2.6 公里，邊權重就會設為 0。

3.2 深度時空圖模型

深度時空圖模型基於圖卷積和時間卷積兩個方法分別處理感測器過往的時序資料及空間特徵，藉此來找到感測器的時空相關性。以下分別介紹圖卷積和時間卷積：

3.2.1 圖卷積 Graph Convolution

圖 (Graph) 是一種資料結構，包含節點 (node)、邊 (edge)、以及邊的權重 (weight)，而圖卷積是要找出圖形中節點與節點的交互關係，主要的方法有兩種：(1) 頻域基礎 Spectral-based、(2) 空間基礎 Spatial-based，以下分別介紹這些方法：

3.2.1.1 頻域基礎 Spectral-based

Shuman 於 2013 年提出頻域基礎運用在圖形資料方法 [15]，一般卷積公式可以使用傅立葉轉換方式做轉換，如式 2：

$$x * g = F^{-1}(F(x) \odot F(g)) \quad (2)$$

其中 x 代表輸入值， g 為濾波器 (filter)， $*$ 為卷積運算， F 為傅立葉轉換函數， \odot 代表 element wise 相乘運算子，因此 x 和 g 的卷積等同於 x 和 g 分別先做傅立葉轉換，然後相乘，最後再做反傅立葉轉換。

而要對圖形資料進行這系列計算，我們必須先找到圖傅立葉轉換。圖傅立葉轉換可以透過權重拉普拉斯矩陣達成，計算方式以圖3.3為例，首先先計算出權重矩陣 (Weight Matrix)、鄰接矩陣 (Adjacency Matrix) 和度矩陣 (Degree Matrix)，求得這三種矩陣之後，將鄰接矩陣和權重矩陣做相乘，再由度矩陣減去相乘後的結果，得到最後的權重拉普拉斯矩陣。

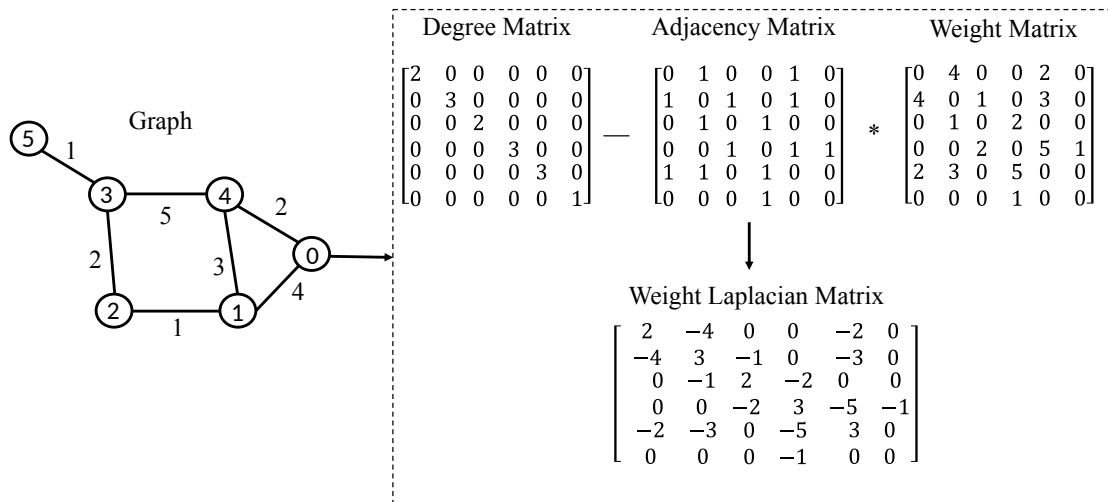


圖 3.3: 權重拉普拉斯矩陣計算流程

圖論中常見的拉普拉斯矩陣，為了避免鄰居節點數量不同的影響，需要進行正規化 (除以鄰居節點數量)，得到最後的正規化權重拉普拉斯矩陣。

陣，公式如式 3

$$\tilde{L}_{ij} = \begin{cases} 1 & , \text{ if } i = j \text{ and } \deg(v_i) \neq 0 \\ -\frac{1}{\sqrt{\deg(v_i)\deg(v_j)}}W_{ij} & , \text{ if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & , \text{ otherwise.} \end{cases} \quad (3)$$

W 代表權重矩陣， $\deg(v_i)$ 代表節點 i 度的數目。

正規化權重拉普拉斯矩陣也可以藉由權重鄰接矩陣及度矩陣運算得到，如式 4:

$$\tilde{L} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \quad (4)$$

其中 W 代表權重鄰接矩陣， D 代表度矩陣， I 代表單位矩陣 (identity matrix)， \tilde{L} 代表正規化權重拉普拉斯矩陣。

接著對正規化權重拉普拉斯矩陣進行特徵分解 [16]，如式 5:

$$\tilde{L} = U\Lambda U^T \quad (5)$$

U 代表特徵向量組成的矩陣，每個列代表特徵向量，而 Λ 代表對角矩陣，對角的元素為特徵值。因此我們可以得到基於頻譜的圖卷積如式 6:

$$x * g = U(U^T x \odot U^T g) \quad (6)$$

3.2.1.2 Chebyshev 多項式近似方法

由於頻譜圖卷積有以下兩種問題: (1) 參數複雜度為 $O(n)$ ， n 為節點個數，(2) 特徵分解的時間複雜度為 $O(n^3)$ 。因此在 2016 年提出用

Chebyshev [17] 多項式來近似卷積核運算如式 7 解決時間複雜度問題:

$$U^T g = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}) , \text{where } \tilde{\Lambda} = \frac{2\Lambda}{\lambda} - I \quad (7)$$

$\theta \in R^K$ 為一個多項式係數向量, λ 值我們設定為 2, 而 T 函數運算方式如式 8:

$$T_0(x) = 1 , T_1(x) = x , T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \quad (8)$$

因此圖卷積如式 9:

$$x * g \approx U \left(\sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}) \odot U^T x \right) \quad (9)$$

因為 Chebyshev 多項式作用在對角矩陣上不會影響矩陣運算。所以改變一下運算順序, 先把矩陣運算放進去, 得到式 10:

$$x * g \approx \sum_{k=0}^{K-1} \theta_k T_k(U\tilde{\Lambda}U^T)x \quad (10)$$

而 $U\tilde{\Lambda}U^T = \tilde{L}$ 所以最後圖卷積的式子為式 11:

$$x * g \approx \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x , \text{where } \tilde{L} = L - I \quad (11)$$

因此權重拉普拉斯矩陣不需做特徵分解, 降低運算所需時間。

3.2.1.3 1st order 近似方法

1st order 近似方法, 是把上述式 12 裡的 K 設定為 2, 則圖卷積可以寫成式 12:

$$x * g \approx \theta_0 x + \theta_1 (L - I)x = \theta_0 x - \theta_1 (D^{-\frac{1}{2}} W D^{-\frac{1}{2}})x \quad (12)$$

其中 W 為權重鄰接矩陣 (weight adjacency matrix) 1^{st} order 近似法會假設 $\theta_0 = -\theta_1$ ，則可以寫成式 13:

$$x * g \approx \theta(I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}})x \quad (13)$$

再分別對權重鄰接矩陣 (weight adjacency matrix) 和度矩陣 (degree matrix) 分別做重新正規化 (renormalization):

$$\widetilde{W} = W + I, \quad \widetilde{D}_{ii} = \sum_j \widetilde{W}_{ij} \quad (14)$$

得到最後的卷積公式

$$x * g \approx \theta(\widetilde{D}^{-\frac{1}{2}} \widetilde{W} \widetilde{D}^{-\frac{1}{2}})x \quad (15)$$

其時間複雜度下降為 $O(E)$ ，同時也巧妙避開計算特徵向量所需的時間。

3.2.1.4 空間基礎 Spatial-based

空間基礎的卷積方法，如圖3.4，把節點其鄰居節點資訊做聚合 (aggregation) 成為自己下一個狀態。

Micheli 等人以聚合方法為基礎，於 2009 年提出 Neural Network for Graphs 方法 [18]，其方法如式 16:

$$\begin{cases} x_1(v) = f \left(\sum_{j=0}^L \bar{w}_{1j} l_j(v) \right) \\ x_i(v) = f \left(\sum_{j=0}^L \bar{w}_{ij} l_j(v) + \sum_{j=1}^{i-1} \hat{w}_{ij} \sum_{t \in Nei(v)} x_j(t) \right) \quad i = 2, \dots, N \\ Nei(v) = \{u \in Vertex(g) | (u, v) \in edge(v)\} \end{cases} \quad (16)$$

其中 f 代表 linear 或是 sigmoid 函式， $x_1(v)$ 代表節點 v 的第 1 層狀

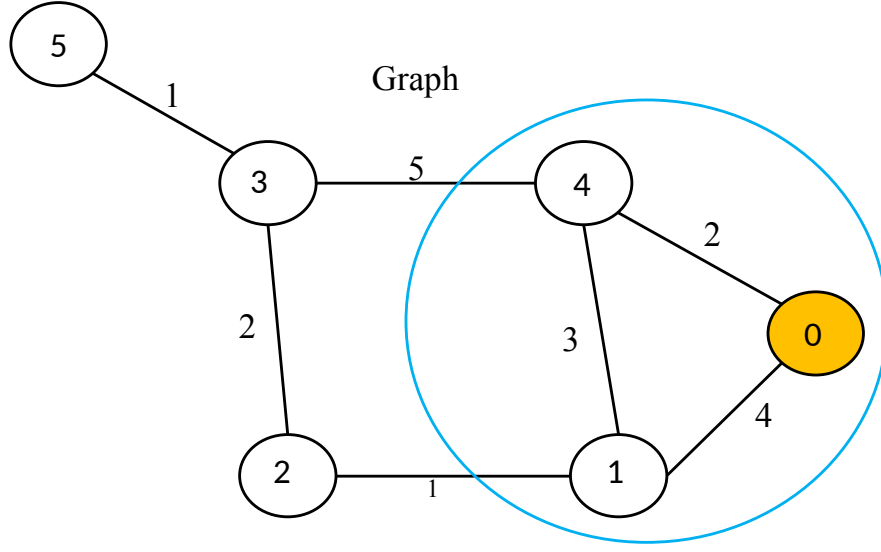


圖 3.4: 聚合示意圖

態，而 $x_i(v)$ 指的是節點 v 的第 i 層狀態， $l(v)$ 代表節點有多個數入特徵， $l_j(v)$ 代表第 j 輸入特徵值，共有 L 個輸入特徵，而 \bar{w} 代表權重處理節點 v 每一層的輸入特徵。 $Nei(v)$ 指的是與節點 v 相鄰的其他節點， \hat{w} 則是權重處理每一層屬於 $Nei(v)$ 節點中每個節點狀態值的總和，而 N 代表共有 N 個層數。

Yaguang Li 等人於 2017 年結合 Diffusion Convolution[19] 方法和 Gated Recurrent Unit 方法 [11] 運用在交通流量預測上，其圖卷積的方法中同時把圖形結構裡邊 (Edge) 的方向性也一起考慮 [10]，方法如式 17:

$$\begin{cases} Z = \sum_{k=0}^K P_f^k X \theta_{k1} + P_b^k X \theta_{k2} \\ P_f^k = (W / \text{rowsum}(W))^k \\ P_b^k = W^T / \text{rowsum}(W^T))^k \end{cases} \quad (17)$$

其中 P_f^k 和 P_b^k 分別代表轉置矩陣， P_f^k 指的是邊的 forward 方向， P_b^k 指的是邊的 backward 方向， W 代表權重鄰接矩陣 (weight adjacency matrix)

), X 代表輸入特徵, θ 代表模型權重。

Zonghan Wu 等人於 2019 年 [1] 基於 Yaguang Li 等人提出的圖卷積方法 [19] 再加入自適應鄰接矩陣 (self-adaptive adjacency matrix) 運用在圖形結構上, 如式 18:

$$\begin{cases} Z = \sum_{k=0}^K P_f^k X \theta_{k1} + P_b^k X \theta_{k2} + \tilde{A}_{apt}^k X \theta_{k3} \\ \tilde{A}_{apt} = SoftMax(ReLU(E_1 E_2^T)) \\ E_1, E_2 \in R^{N \times c} \end{cases} \quad (18)$$

其中 \tilde{A}_{apt} 代表自適應鄰接矩陣, 由 E_1 和 E_2 內積而成, E_1 代表來源節點的 embedding, E_2 代表目標節點的 embedding, embedding 維度為 $R^{N \times c}$, N 代表節點 (node) 個數, c 為向量維度, θ_{k1} 、 θ_{k2} 、 θ_{k3} 代表模型權重, 而 \tilde{A}_{apt} 裡的數值會經由神經網路的訓練過程中做學習更改。

我們圖卷積方法使用 Zonghan Wu 等人提出的方法, 但因為我們的圖形資料是無向的, 所以最後的圖卷積如式 19:

$$Z = \sum_{k=0}^K P^k X \theta_{k1} + \tilde{A}_{apt}^k X \theta_{k2} \quad (19)$$

3.2.2 時間卷積 Temporal Convolution

時間卷積目的是從節點過往的時序資料中找到相關性, 其卷積層結合了因果卷積 (Causal Convolution) [12] 與擴張卷積 (Dilated Convolution) 兩種結構 [13]。因果卷積定義了輸出 y_t 只受到過去資訊 ($X_t, X_{t-1}, X_{t-2}, \dots$) 並不受未來數據 (X_{t+1}, X_{t+2}, \dots) 影響。擴張卷積解決一般卷積只能看到局部性的資訊問題, 如圖3.5, 以 2 的指數遞增, 在第幾層就會以 2 的指數級別隔幾個空格做卷積運算, 並把結果傳遞至下一層。

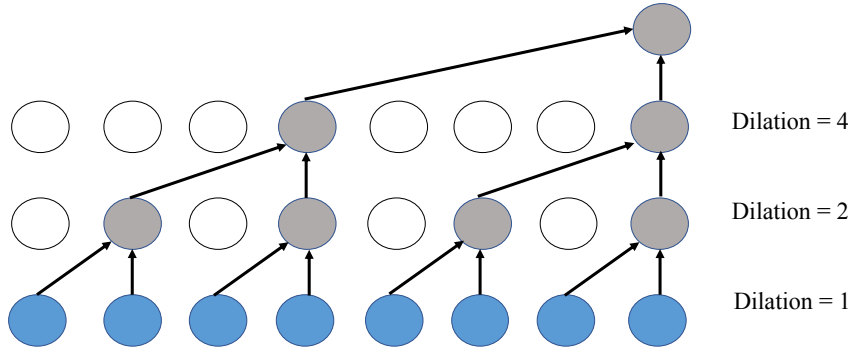


圖 3.5: 時間卷積示意圖

因此時間卷積的公式如式 20:

$$x \star f(t) = \sum_{s=0}^{K-1} f(s)x(t - d \times s) \quad (20)$$

其中 x 代表輸入特徵值， $f(t)$ 代表卷積核函式， d 代表擴張值。

3.2.3 模型架構

我們模型以 Zonghan Wu 於 2019 [1] 年提出的 Graph WaveNet 架構為基礎再做些微更改，其包含圖卷積和時間卷積兩個方法，但由於我們的圖形資料不包含邊 (Edge) 的方向性，因此我們圖卷積方法不會考慮方向性特徵。模型架構如圖3.6。

模型架構中，左邊是模型的輸入部分，堆疊了 K 層時空層，提取時空相關性特徵。每個時空層包括一個 gated TCN 模組 [20] 和 GCN 模組。其中的 gated TCN 用於提取時間相關性，包括兩個 TCN 模組，分別是 TCN-a 和 TCN-b。TCN 模組提取到時間相關性特徵之後，傳到 GCN 層提取空間相關性。右側是輸出層，相當於一個 MLP（多層感知器）。TCN 模組運作細節如3.2.2節介紹，GCN 模組則如3.2.1.4節介紹。TCN-a

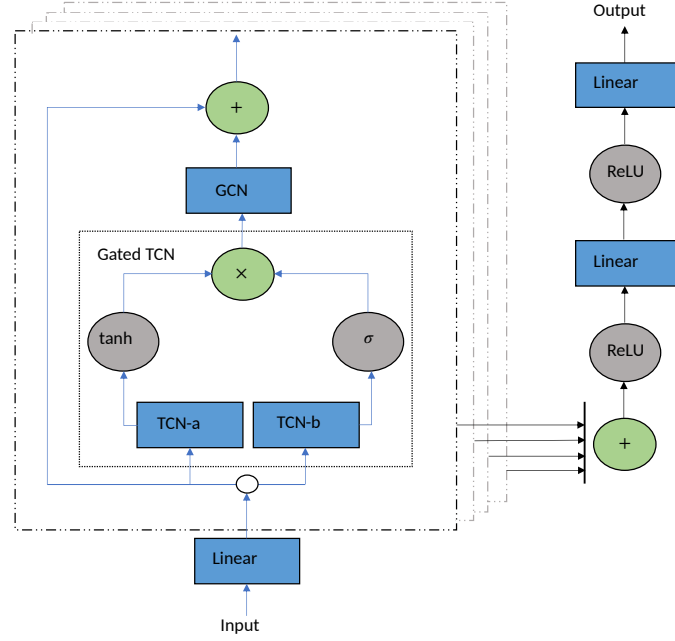


圖 3.6: 模型架構 [1]

與 TCN-b 兩個結構分別接受輸入資料經過一個 Linear 層轉換後的張量值，並獨立做時間卷積流程 (參閱3.2.2節)，最後得到兩個輸出，兩個輸出會以 Gate Mechanism 做結合 [20]。公式如式 21:

$$h = \tanh(\theta_1) \odot \sigma(\theta_2) \quad (21)$$

其中 θ_1 是 TCN-a 模組的輸出， θ_2 是 TCN-b 模組的輸出，兩者分別以 tanh 和 sigmoid 激活函數產生輸出後做 element wise 相乘，產生最後輸出 h 。

我們把每個 TCN 模組裡的層數設置為 3，而時空層的 k 值設置為 4，並把 4 層的時空層各自輸出結果作相加，得到的結果再經由 MLP 轉換成輸出。

3.2.4 STGCN 模型介紹

B.Yu 等人提出 STGCN(Spatio-Temporal Graph Convolutional Networks) 模型運用在交通流量預測上 [2]，其結合頻域基礎的圖卷積和因果卷積，我們也會實作 STGCN 模型運用在異常空氣品質感測器上，並和 GraphWaveNet 及其他模型做比較。STGCN 模型架構如圖3.7所示 [2]，圖中的 W 代表 weight laplacian matrix，整體架構有兩個 ST-Conv Block 和一個 output block，ST-Conv Block 是由兩個 Temporal Gated-Conv 模組及夾者 Graph Conv 模塊，Graph Conv 模塊是採用頻域基礎的圖卷積 (參閱3.2.1.1、3.2.1.2節及3.2.1.3)，而 Temporal Gated-Conv 運作流程是輸入資料分別經由兩個 casual convolution 得到各自的輸出，其中一個的輸出跟輸入做相加 (Residual)，最後兩個在做 GLU 運算得到輸出。GLU 運算公式如式 22:

$$h = \theta_1 \odot \sigma(\theta_2) \quad (22)$$

其中 θ_1 代表其中一個的 casual convolution 的輸出跟輸入資料做相加 (Residual) 結果， θ_2 代表另一個的 casual convolution 的輸出， σ 代表 sigmoid 激活函數，兩者得到的結果再做 element wise 相乘。最後 output block 把經由兩個 ST-Conv Block 的結果再做線性轉換得到輸出。

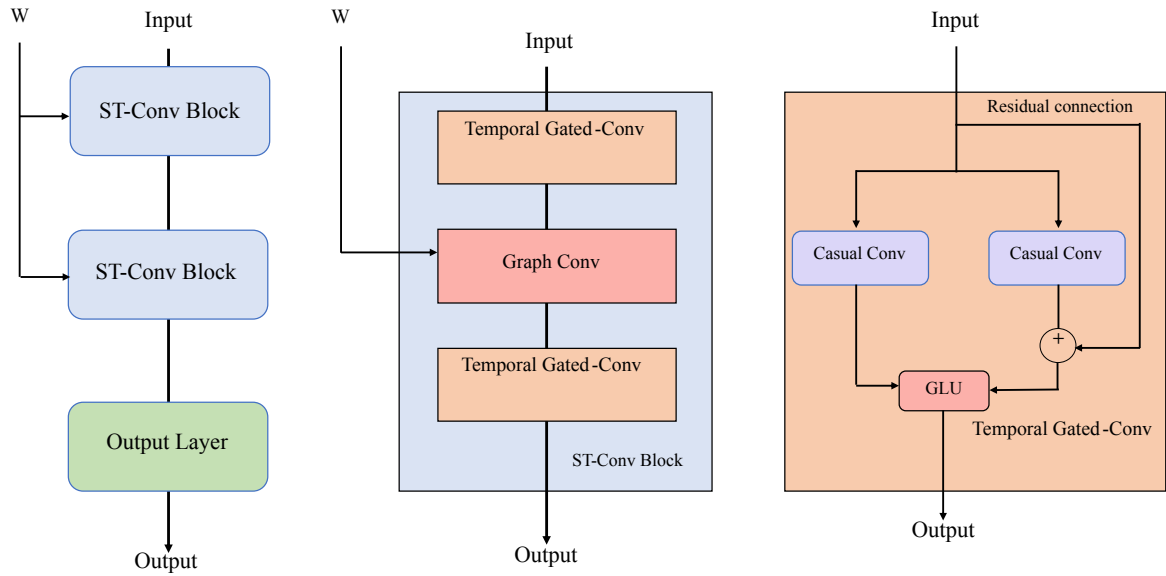


圖 3.7: STGCN 架構圖 [2]

3.3 異常偵測模組

異常偵測模組則是將訓練完的深度時空圖模型去預測每個空氣品質感測器接下來一天的 pm2.5 時序值，針對每個感測器，此模組會分別對感測器接下來一天所偵到的 pm2.5 時序值和模型對此感測器預測到 pm2.5 時序值算出 R2-score [5]，因此每個感測器都有其對應的 R2-score 值，以及還有其對應標籤值，如果感測器是異常其標籤值為 1，正常的標籤值為 0。最後對每個感測器的 R2-score 做由低到高排序，R2-score 越低代表此感測器越可能是異常的，我們會對排序過的感測器以及每個感測器的對應標籤算出整體 ROC curve 分數來顯示模型的好壞。圖3.8顯示異常偵測模組的整體流程。

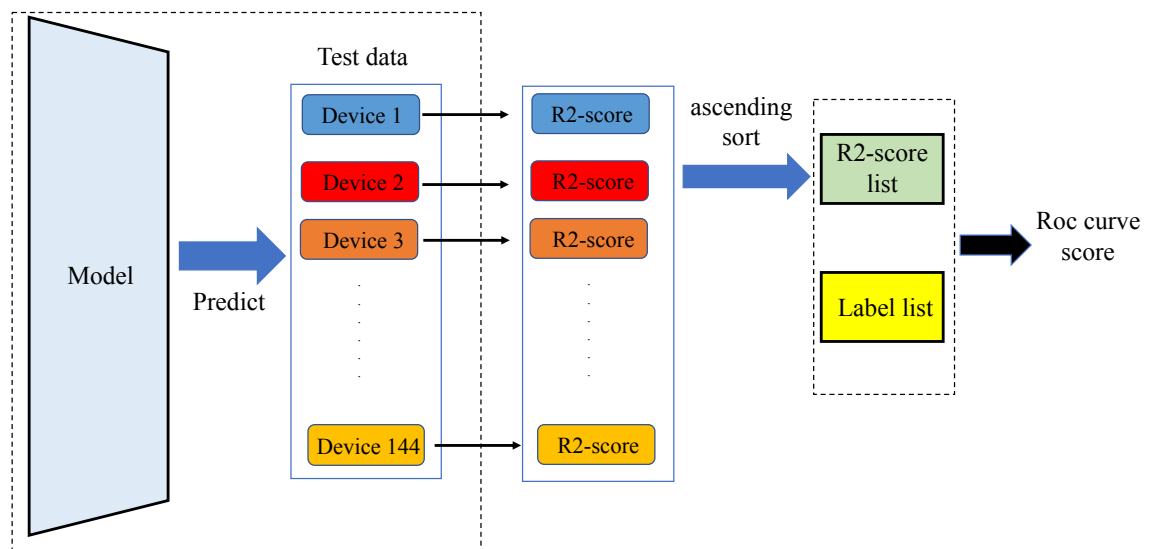


圖 3.8: 異常偵測模組流程圖

四、實驗結果

在本章節中，我們會先介紹所使用的資料集，接著說明實驗的流程與詳細的操作方法，然後展示不同方法其實驗結果，最後我們會對實驗結果中所發現的現象進行進一步討論。

4.1 實驗流程與細節

本研究全程使用 Python3.6.9 版本作為實驗開發環境，圖4.1展示了整體的實驗流程，我們會在依序子章節介紹每個細節。

4.1.1 空汙感測器資料集介紹

實驗的空汙感測器巡檢資料是由工研院綠能與環境研究所提供的，共有 155 個巡檢感測器，移除 11 個資料缺失嚴重的感測器，剩下 144 個感測器，其中 116 個是正常的，我們把其標籤設為 0，28 個是異常的，其標籤設為 1，每個感測器的 pm2.5 時序資料從 2018 年 1 月至 2018 年 12 月，以每分鐘為單位，感測器分布地點是在整個台灣台中市地區，且所在地方主要是工業區，每個感測器都是在室外。

感測器兩兩之間最近的距離約為 200 公尺，兩兩最遠的距離約為 30 公里。感測器是否有故障是經由工研院綠能與環境研究人員親自帶著一個正常的感測器到要巡檢的感測器地點進行一整天的數據比對，如果兩者數據差異過大，則判斷被巡檢的感測器是有問題。

4.1.2 資料清洗

針對缺失的 pm2.5 值，會以前一分鐘做填補。如果連續缺失超過一天，會以前一天的 pm2.5 序列值填補到缺失的那天。

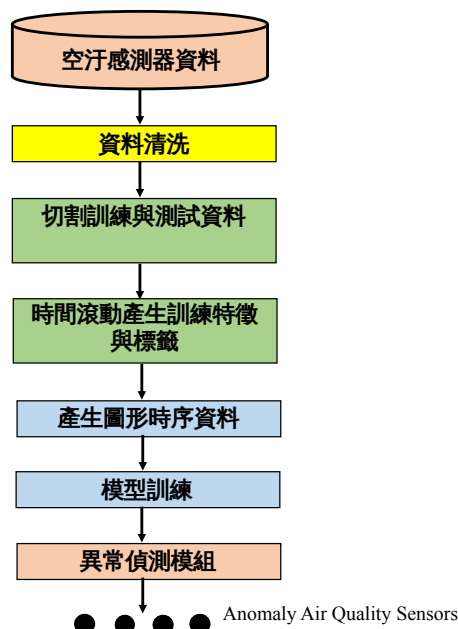


圖 4.1: 實驗流程圖

4.1.3 切割訓練資料與測試資料

我們以巡檢的日期的資料當作測試資料，巡檢的日期前的資料當作訓練資料。舉例來說，如果某個感測器其被巡檢日期是 2018 年 5 月 29 號則測試資料就是該感測器 2018 年 5 月 29 號偵測到的數值，共 1440 筆 (24×60，分鐘為單位)，2018 年 1 月 1 號至 2018 年 5 月 28 號的時序資料就是訓練資料。

4.1.4 選取周圍感測器個數

由於感測器會根據其所在的地點，而有不同數據空間解析度，如果分布地點在城市或工業區，數據空間解析度為 100~300 公尺，如果分布

在校園外圍或居家外圍，其數據空間解析度為 1 公里 [21]。

圖4.2為工研院綠能與環境研究所提供的資料集裡每個感測器站點對周圍感測器站點群中，兩兩之間距離取平均，144 個站點的平均距離箱型分佈。其中 x 軸的 N_3 代表每個感測器取周圍最近的 3 個感測器，算出與這 3 個感測器兩兩距離再取平均，然後畫出 144 個感測器的平均距離箱型分佈圖，y 軸顯示分佈圖結果，且以公里為單位，而 N_4 到 N_12 也是上述的方式以此類推。由於我們的空汙感測器絕大部分都是分佈於工業區，且考慮感測器與周圍感測器平均距離分佈情形 (圖4.2)，我們決定取周圍最近的 5 個感測器前 30 分鐘 pm2.5 值當作訓練特徵。

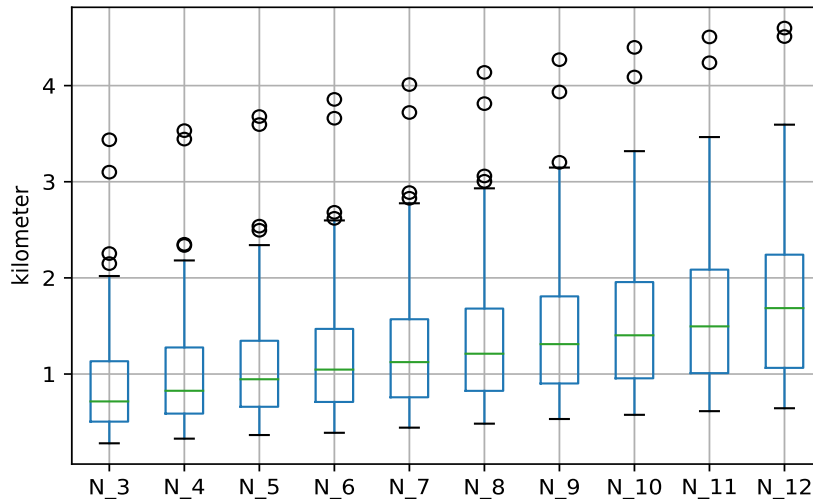


圖 4.2: 中心感測器站點與周圍感測器站點群平均距離箱型分佈圖

4.1.5 時間滾動產生訓練特徵與標籤

切割完訓練資料和測試資料後，我們需要對模型進行回歸訓練，由於我們是取中心感測器前 30 分鐘 pm2.5 值和周圍最近的 5 個感測器前 30 分鐘 pm2.5 值當作訓練特徵，而標籤值則是中心感測器下一分鐘的數值，因此時間滾動產生方式來產生上述的訓練資料，圖 4.3顯示了這樣的流程，其中 $(T_1, T_2, \dots, T_{N-1}, T_N)$ 代表時間軸。

4.1.6 產生圖形時序資料

時間滾動產生完訓練特徵與標籤後，我們會對中心感測器和最近的 5 個感測器建立成一個無向全連接圖，我們根據圖4.2中每個感測器與其周圍 5 個感測器的平均距離分佈情形，設定當兩點距離大於 2.6 公里時，邊權重為 0(建構方式參閱3.1節)，而每個感測器前 30 分鐘 pm2.5 就是其對應節點的輸入特徵。以此建構出圖形時序資料，且每個節點的 pm2.5 時序值會做 z-score 正規化，z-score 公式如式 23:

$$y = \frac{x - \mu}{\sigma} \quad (23)$$

其中 x 代表要被標準化的原始數值， μ 代表原始數值的平均值， σ 代表原始數值的標準差， y 代表正規化的結果。

4.1.7 模型訓練

模型訓練方式會根據輸入模型的節點 (node) 數量及要預測下一分鐘的節點數量而分為不同方式，以圖4.3為例，輸入模型的節點包含中心節點 C_i 和其周圍最近的 5 個節點 $N_1^{C_i}, N_2^{C_i} \dots N_5^{C_i}$ ，來預測中心節點下一分鐘 pm2.5 值，但我們也可以輸入 144 個節點 (總共感測器個數) 來預測每個節點 (共 144 個) 下一分鐘 pm2.5 值。前者方式我們稱為多對一，後者的方式我們稱為全部對全部，我們分別介紹兩者訓練方式。

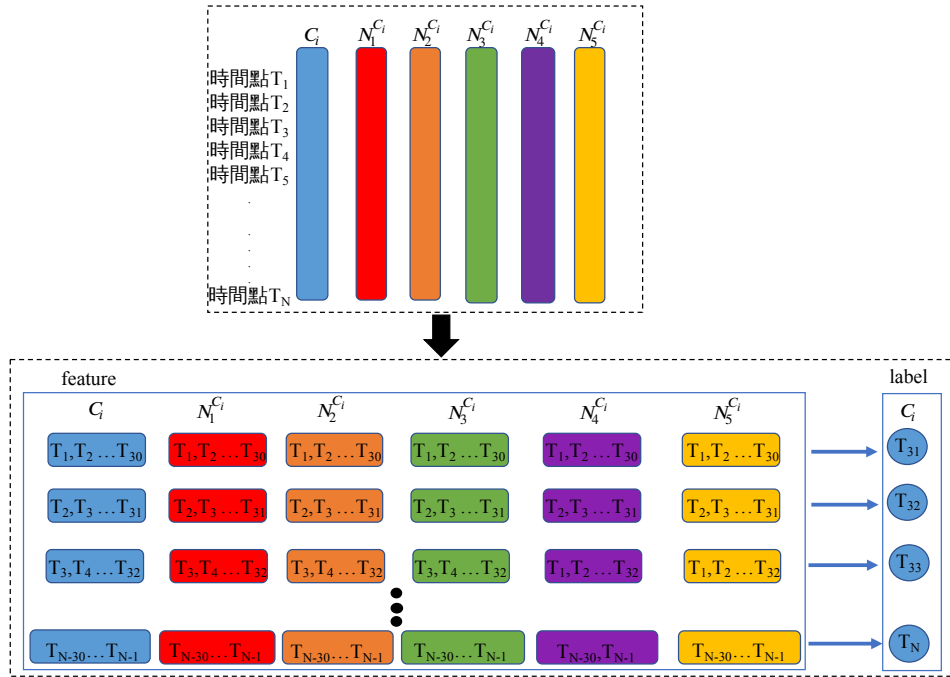


圖 4.3: 時間滾動產生訓練資料。 C_i 代表中心節點，藍色方形代表中心節點的時序值，而 $N_1^{C_i}, N_2^{C_i}, \dots, N_5^{C_i}$ 則代表與 C_i 最相近的 5 個節點。

4.1.7.1 多對一 (模型統一與模型分離)

多對一的訓練方式會取一個感測器視為中心點，和離中心最近的 5 個感測器建構一個圖序列資料進行特徵，直到重覆 144 個感測器都執行上述步驟。同時此方法又可以分為「模型分離」及「模型統一」兩種訓練方式。

圖4.4顯示模型分離的訓練，圖片中 C_1 代表以感測器 1 為中心節點，而 $(N_1^{C_1}, N_2^{C_1}, N_3^{C_1}, N_4^{C_1}, N_5^{C_1})$ 代表離 C_1 中圍最近的五個點，同時由 Model 1 進行資料訓練以及測試。以此方法類推到第 2 至 144 的中心節點，需要特別注意的是，Model 1, Model 2...Model 144 都是獨立的模型，其參數都是先隨機初始化之後再根據各自的訓練資料進行訓練。

圖4.5則是「模型統一」的訓練方式，由一個模型去訓練不同中心點其圖形時序資料，圖形結構會根據中心點的不同做改變，我們會分別實作這兩種訓練方式並顯示其結果差異。

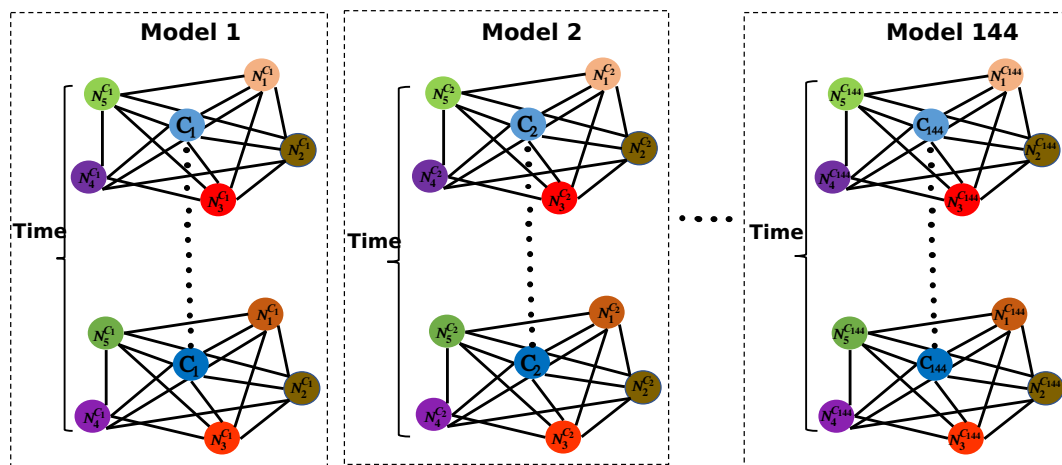


圖 4.4: 模型分離訓練方式。每個框線裡的圖形資料代表以一個感測器為中心點，和最近的 5 個感測器建構成一個圖序列資料，因此共有 144 個框線資料，每個框線資料都會由一個模型去訓練，因此共訓練出 144 個模型，144 個都是獨立的模型，其參數都是先隨機初始化之後再根據各自的框線訓練資料進行訓練。

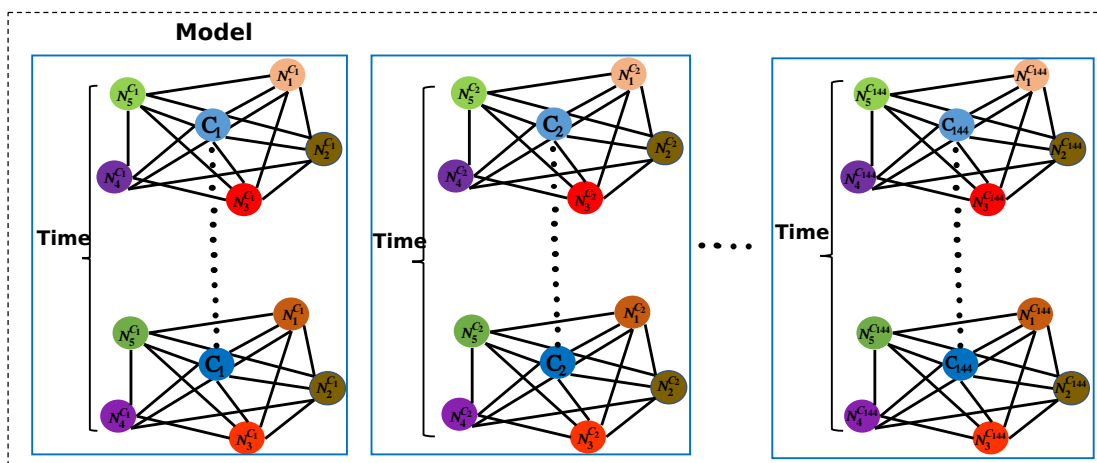


圖 4.5: 模型統一訓練方式。每個藍色框線裡的圖形資料代表以一個感測器為中心點，和最近的 5 個感測器建構成一個圖序列資料，因此共有 144 個藍色框線資料，圖形結構會根據不同中心點而有不同的邊權重，由 1 個模型訓練 144 個藍色框線資料。

4.1.7.2 全部對全部

要達成全部對全部的模型訓練方式，首先要經由時間滾動方式產生 144 個前三十分鐘序列值 (總共感測器個數) 來預測每個感測器 (共 144 個) 下一分鐘 pm2.5 值，圖4.6描述了這樣的運作流程。時間滾動產生完訓練特徵與標籤後，我們會對 144 個感測器建立成一個無向全連接圖，並輸入模型進行訓練。我們也會實作此方法並做比較。

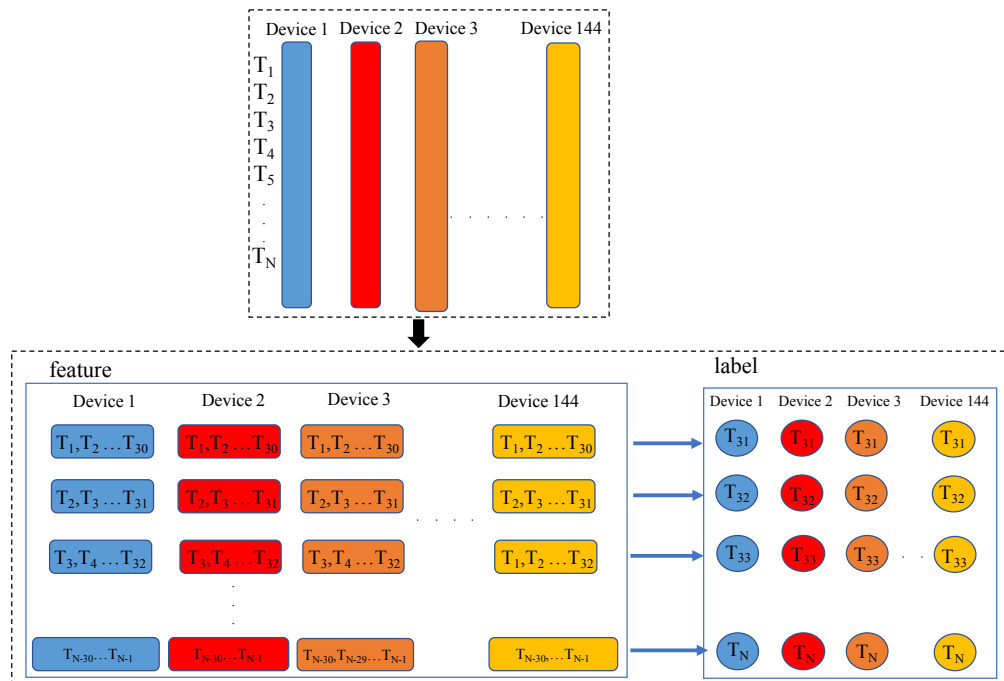


圖 4.6: 時間滾動產生全部對全部訓練資料

4.1.7.3 不同訓練方式總結

表4.1總結了不同種類的訓練方法的特性，根據上面敘述，訓練的方式共有三種:(1) 模型分離訓練、(2) 模型統一訓練、(3) 全部對全部。前兩個訓練方式屬於多對一 (參閱4.1.7.1節)，圖形的節點個數都只有 6 個，但兩者差別在於，模型分離訓練流程裡每個中心節點產生的圖時序資料都分別由一個模型來訓練，共會產生 144 個模型 (參閱4.4解說)。而模型統一訓練裡由一個模型來訓練所有中心節點的圖時序資料，因此輸入模型

表 4.1: 不同訓練方式總結

Type	圖形節點個數	輸入至每個模型的圖形結構種類	產生模型個數
模型分離訓練	6	1	144
模型統一訓練	6	144	1
全部對全部	144	1	1

共有 144 個不同種類圖形結構及邊權重，但產生出的模型只有一個。(參閱4.5節解說)。

而全部對全部訓練流程中，圖形的節點個數共有 144 個，其輸入至每個模型的圖形結構有 1 個種類，產生出的模型只有 1 個。(詳細流程請參閱4.1.7.2節)。

我們會實作上述三種訓練方式，並比較其實驗結果差異並提出會造成差異的原因，最後也會總結這三種訓練方式適合運用的情境。

4.1.8 異常偵測模組

模型訓練完之後，會經由異常偵測模組對感測器做排序 (排序方式請參考3.3節)，我們會顯示出排序完後其 ROC curve 分數 [22, 23] 以及前 10~50 個感測器其找到有問題的精確度 (precision@n) 和召回率 (recall@n)。

4.2 實驗結果

4.2.1 評量指標

實驗結果的評量指標以異常偵測模組對感測器做排序之後得到其 ROC curve 分數 [22, 23]，同時我們主要以此分數來評量訓練模型的好壞，同時也會顯示排序過後前 10~50 個感測器其找到有問題的精確度 (precision@n) 和召回率 (recall@n)，以及最後我們會顯示每個模型其 PR curve 分數。

精確度 (precision@n) 定義方式如式 24:

$$P @ n = \frac{n \text{個感測器中有問題個數}}{n} \quad (24)$$

而召回率 (recall@n) 的定義方式如式 25:

$$R @ n = \frac{n \text{個感測器中有問題個數}}{\text{總共異常感測器個數}} \quad (25)$$

4.2.2 ROC curve 分數結果

表 4.2顯示出以統計基礎方法和不同機器學習模型經過感測器訓練資料後，對每個感測器的測試資料算出 R2-score[5]，經由我們提出的異常偵測模組根據感測器的 R2-score 做排序，再計算整體 ROC 分數。我們實作 ADF[6]、Random Forest[24, 25]、Lasso Regression[26, 27]、Ridge Regression[28, 29, 30]、Deep Neural Network [31, 32]、Long Short-Term Memory(LSTM) [8]、Temporal Convolution Network [33]。需要注意的是，上述的所有模型在實驗流程中 (參閱圖4.1)，是沒有產生圖形時序資料這項步驟的，其直接把時序資料輸入模型作訓練，並對測試資料算出 R2-score。時空圖模型包括 STGCN(Spatio-Temporal Graph Convolutional Networks) [2] 模型和 GraphwaveNet 模型 [1]，上述兩個模型其實驗流程是有產生圖形時序資料這項步驟。

表 4.2中，每個機器學習訓練方法是以多對一方式 (參閱4.1.7.1節)，且以模型分離方式做學習 (參閱圖4.4)。我們對每個模型都跑了 5 次，並在表4.2顯示 5 次的 ROC 平均值和標準差。實驗結果中可以看到時空圖模型比起其他機器學習模型其 ROC 分數都來的好，因此圖形結構資料對感測器異常偵測是有影響的。而時空圖模型中，GraphWaveNet 模型與 STGCN 模型來的 ROC 分數相差不大，但 GraphWaveNet 模型稍好一些。

表 4.2: 不同訓練模型經由異常偵測模組算出其 ROC 分數

Type	Model	ROC mean	ROC std
Rule based	ADF-5 (5 是 [6] 中給的超參數值)	0.624	0.0
	ADF-10(ROC Best)	0.694	0.0
ML(無圖卷積)	Random Forest	0.6878	0.006261
	Lasso	0.7000	0.015652
	Ridge	0.7085	0.013472
	TCN	0.7066	0.007701
	DNN	0.6940	0.007211
	LSTM	0.7090	0.007211
ML(圖卷積)	GraphWaveNet	0.7260	0.010826
	STGCN	0.7214	0.018569

我們也實作 ADF 偵測方法 [6]，其方法運用在我們空汙感測器資料，經由感測器排序後其 ROC 分數。我們分別顯示出規則裡的超參數設定後最好的 ROC 分數與 [6] 中預設的超參數數值。其最好的 ROC 分數是設定取中心感測器其周圍 10 個感測器經由統計規則方法判斷是否為空間異常 (表4.2中的 ADF-10)，ADF-5 則是取周圍 5 個感測器 ([6] 的預設值)。儘管超參數做了調整，其結果依然沒有比我們提出的基於時空圖模型異常偵測框架來的好。

4.2.3 前 10 至 50 個感測器其精確率

表4.3顯示我們取排序後其前 10~50 個感測器中，真實異常感測器的精確率。會這樣計算的原因在於，空汙感測器巡檢的人力成本往往是有限，但從工研院綠能與環境研究所隨機挑選的 144 個感測器中，只有找到 28 個感測器是異常的，精確率只有 0.194 左右，其巡檢所花費時間要一年左右。但經由我們提出的基於時空圖模型異常偵測框架 (GraphWaveNet 和 STGCN)，可以發現若取框架偵測出來 10 個最有可能異常的感測器中，其精準率有 60% 以上，30 個最有可能異常的感測器中，其精準率有 40% 左右，50 個最有可能異常的感測器中，其精準率可以保持 35% 以

表 4.3: 機器學習方法前 10 至 50 個感測器其找到有問題感測器精準率

Type	Model	P@10	P@20	P@30	P@40	P@50
隨機巡檢		0.194	0.194	0.194	0.194	0.194
Rule based	ADF-5	0.300	0.350	0.270	0.330	0.320
	ADF-10(ROC Best)	0.500	0.500	0.400	0.380	0.320
ML(無圖卷積)	Random Forest	0.380	0.370	0.400	0.342	0.320
	Lasso	0.580	0.430	0.394	0.370	0.320
	Ridge	0.600	0.433	0.395	0.375	0.337
	TCN	0.600	0.410	0.412	0.338	0.320
	DNN	0.500	0.430	0.374	0.344	0.312
	LSTM	0.600	0.410	0.368	0.332	0.336
ML(圖卷積)	GraphWaveNet	0.600	0.417	0.417	0.380	0.353
	STGCN	0.640	0.450	0.398	0.386	0.360

上，是可以降低整體巡檢成本。

4.2.4 前 10 至 50 個感測器其召回率

表4.4顯示我們取排序後其前 10~50 個感測器中，其召回率。可以發現 GraphWaveNet 和 STGCN 兩者召回率相差不大，但整體來說 STGCN 略好些，同時也可以發現有圖卷積方法模型的召回率比其他機器學習或規則基礎模型都來的高。若取框架偵測出來 10 個最有可能異常的感測器中，其召回率有 20% 以上，30 個最有可能異常的感測器中，其召回率有 42% 以上，50 個最有可能異常的感測器中，其召回率可以保持 60% 以上。

表 4.4: 機器學習方法前 10 至 50 個感測器其找到有問題感測器召回率

Type	Model	R@10	R@20	R@30	R@40	R@50
隨機巡檢		0.069	0.139	0.208	0.278	0.347
Rule based	ADF-5	0.110	0.250	0.290	0.460	0.570
	ADF-10(ROC Best)	0.180	0.360	0.430	0.540	0.570
ML(無圖卷積)	Random Forest	0.136	0.266	0.428	0.484	0.570
	Lasso	0.204	0.306	0.422	0.524	0.570
	Ridge	0.210	0.308	0.423	0.533	0.603
	TCN	0.212	0.296	0.442	0.476	0.570
	DNN	0.180	0.308	0.398	0.484	0.560
	LSTM	0.214	0.293	0.394	0.474	0.600
ML(圖卷積)	GraphWaveNet	0.214	0.300	0.447	0.543	0.630
	STGCN	0.230	0.322	0.428	0.550	0.642

4.2.5 模型分離與模型統一結果比較

表4.5顯示模型分離訓練方式與模型統一訓練方式其 ROC 比較結果，可以發現訓練方式從模型分離改為模型統一之後，GraphWaveNet 模型和 STGCN 模型兩者的結果並沒有相差太多，但 STGCN 模型分數稍微高些。模型統一訓練方式中，其單一模型可以學到 144 種不同結構的圖形資料，每個圖形資料都包含 6 個節點 (參閱圖4.5)，再根據每個中心節點的測試資料去做預測。我們認為，空汙感測器群其分布位置的地理資訊是相似的話，則每個中心感測器形成的圖形結構是也會是相似的，這時候模型統一訓練方式可能會比較適合，因為單一模型可以同時學到這些相似的圖形結構。相反的，如果地理資訊差異性很大時，模型分離的訓練方式會比較適合，因為單一模型只針對某個中心節點形成的圖形結構做訓練，部會受到其他圖形結構的雜訊影結果。

表4.6顯示模型分離與模型統一前 10 至 50 個感測器其找到有問題感測器精準率比較，同樣在模型統一訓練中 STGCN 在前 10 至 50 精準率模型分數比 GraphWaveNet 高，但同樣可以發現 10 個最有可能異常的感測器中，其精準率有 54% 以上，30 個最有可能異常的感測器中，其精準率有 40% 以上，50 個最有可能異常的感測器中，其精準率可以保持 35%

表 4.5: 模型分離與模型統一結果比較

Type	Model	ROC mean	ROC std
模型分離	GraphWaveNet	0.7260	0.010826
	STGCN	0.7214	0.018569
模型統一	GraphWaveNet	0.7216	0.014690
	STGCN	0.7220	0.014182

表 4.6: 模型分離與模型統一前 10 至 50 個感測器其找到有問題感測器精準率比較

Type	Model	P@10	P@20	P@30	P@40	P@50
模型分離	GraphWaveNet	0.600	0.417	0.417	0.380	0.360
	STGCN	0.640	0.450	0.398	0.386	0.360
模型統一	GraphWaveNet	0.540	0.420	0.400	0.378	0.348
	STGCN	0.588	0.438	0.419	0.414	0.372

左右，是可以降低整體巡檢成本。

表4.7顯示模型分離與模型統一前 10 至 50 個感測器其找到有問題感測器召回率比較，在 STGCN 模型中，在前 10~20 感測器中，STGCN 模型在模型分離訓練方式結果比模型統一來的好，但在 30~50 中，模型統一訓練方式結果比模型分離來的好。GraphWaveNet 中，以模型分離訓練方式整體結果比模型統一結果來的好。

表 4.7: 模型分離與模型統一前 10 至 50 個感測器其找到有問題感測器召回率比較

Type	Model	R@10	R@20	R@30	R@40	R@50
模型分離	GraphWaveNet	0.214	0.300	0.447	0.543	0.630
	STGCN	0.230	0.322	0.428	0.550	0.642
模型統一	GraphWaveNet	0.190	0.302	0.428	0.538	0.622
	STGCN	0.209	0.314	0.449	0.594	0.665

4.2.6 全部對全部與多對一結果比較

表 4.8顯示全部對全部訓練方式與多對一訓練方式的比較結果，每個模型都是實驗過 5 次並顯示結果平均值和標準差，可以發現 GraphWaveNet 模型對 144 個感測器建構而成的圖時序資料，可以掌握更精確的時空相關性，其訓練結果比以一個感測器視為中心點，和離最近的 5 個感測器建構而成的圖序列資料效果來的好。但相反的 STGCN 模型對於 144 個感測器建構而成的圖時序資料反而比較差。

我們猜測原因在於 STGCN 模型的圖卷機方法是使用 Chebyshev [17] 多項式來近似卷積核，來降低運算時間複雜度，但對於 144 個節點資訊時，多項式近似可能會造成精準度下降許多。同樣是中心點和離最近的 5 個感測器的圖資訊，其只有 6 個節點空間資訊而已，多項式近似不太會造成精準率影響。而 GraphWaveNet 模型的圖卷積是基於 Diffusion Convolution 方法再加入自適應鄰接矩陣 [1]，其圖卷積方法是沒有用近似方式做取代，因此對於 144 個大範圍節點資訊時，其訓練時間雖然比 STGCN 模型久，但相對找到空間精準度也比較好。

此外在找尋時間相關性的方法差異中，GraphWaveNet 模型使用因果卷積（Causal Convolution）與擴張卷積（Dilated Convolution）兩種結構，在多個時序資料中，可以更精確抓準時間精準度。綜合上述這兩項原因，GraphWaveNet 模型在全部對全部訓練方式中比 STGCN 模型以任何訓練方式效果來的好。同樣的模型中，GraphWaveNet 在全部對全部訓練結果也比模型統一或模型分離來的好。

表 4.9顯示全部對全部與多對一前 10 至 50 個感測器其找到有問題感測器精準率比較，同樣可以發現 GraphWaveNet 模型在全部對全部訓練方式前 20 至 50 個的精確率都比其他方法來得高。可以得見基於 GraphWaveNet 模型的異常偵測框架在大範圍圖形序列資訊中，其異常偵測的準確率比較高。

表4.10顯示全部對全部與多對一前 10 至 50 個感測器其找到有問題

表 4.8: 全部對全部與多對一其 ROC 分數比較

Type	Model	ROC mean	ROC std
多對一 (模型分離)	GraphWaveNet	0.7260	0.010826
	STGCN	0.7214	0.018569
多對一 (模型統一)	GraphWaveNet	0.7216	0.014690
	STGCN	0.7220	0.014182
全部對全部	GraphWaveNet	0.7615	0.017547
	STGCN	0.7168	0.010521

表 4.9: 全部對全部與多對一前 10 至 50 個感測器其找到有問題感測器精準率比較

Type	Model	P@10	P@20	P@30	P@40	P@50
多對一 (模型分離)	GraphWaveNet	0.600	0.417	0.417	0.380	0.360
	STGCN	0.640	0.450	0.398	0.386	0.360
多對一 (模型統一)	GraphWaveNet	0.540	0.420	0.400	0.378	0.348
	STGCN	0.588	0.438	0.419	0.414	0.372
全部對全部	GraphWaveNet	0.550	0.450	0.427	0.428	0.377
	STGCN	0.400	0.400	0.386	0.396	0.376

感測器召回率比較，同樣 GraphWaveNet 模型在全部對全部訓練方式前 20 至 50 個的召回率都比其他方法來得高，整體結果也是最好，相反的 STGCN 模型在全部對全部訓練方式中，其召回率反而沒有比多對一來的佳。可以得見基於 GraphWaveNet 模型的異常偵測框架在大範圍圖形序列資訊中，其異常偵測的召回率比較高。

表 4.10: 全部對全部與多對一前 10 至 50 個感測器其找到有問題感測器精準率比較

Type	Model	R@10	R@20	R@30	R@40	R@50
多對一 (模型分離)	GraphWaveNet	0.214	0.300	0.447	0.543	0.630
	STGCN	0.230	0.322	0.428	0.550	0.642
多對一 (模型統一)	GraphWaveNet	0.190	0.302	0.428	0.538	0.622
	STGCN	0.209	0.314	0.449	0.594	0.665
全部對全部	GraphWaveNet	0.193	0.322	0.457	0.613	0.675
	STGCN	0.142	0.286	0.414	0.564	0.670

4.2.7 PR curve 分數結果

表4.11顯示出以統計基礎方法和不同機器學習模型經由不同訓練方式訓練之後，對每個感測器的測試資料算出 R2-score[5]，經由我們提出的異常偵測模組根據感測器的 R2-score 做排序，再計算整體 PR curve 分數。

其結果最好的結果是 GraphWaveNet 模型以多對多訓練方式，而 ADF 偵測方法中，設定取中心感測器其周圍 10 個感測器得到的 PR 分數比其他機器學習方法都來的好，即便 Roc 分數比起其他機器學習模型並沒有來的比較高。此外 STGCN 模型在多對一訓練方式中，整體結果都比 GraphWaveNet 模型好。而 GraphWaveNet 模型則是在全部對全部訓練結果比 STGCN 模型高出不少。

4.2.8 正常感測器與異常感測器其巡檢當日的 R2-score 平均值

表4.12顯示每個模型對 144 個感測器裡，正常群 (共 118 個) 和異常群 (共 26 個) 感測器的 R2-score 分數平均值。每個感測器都有其對應的 R2-score 值，R2-score 數值是由感測器於巡檢日當天偵測到的 pm2.5 數值 (共 1440 筆，分鐘為單位)，和模型預測的 pm2.5 數值計算出來的，如果兩者數據相近的話，R2-score 分數會越接近 1，相反的，會接近負無限。理論上我們會希望正常群數值越接近 1，異常群數值越低越好，且正常群和異常群的 R2-score 差值越大越好。

表中正常群 R2-score 平均值最高的是 LSTM 模型，分數為 0.8467，但其異常群分數卻有 0.7002，正常群和異常群差值不大。而 GraphWaveNet 模型在全部對全部訓練方式中，其異常群平均 R2-score 是最低的，且正常群和異常群差值也是最高的。同時也可以發現深度圖時空模型以任何訓練方式，其正常群和異常群差值都有 0.15 以上。

表 4.11: 不同訓練模型經由異常偵測模組算出其 PR curve 分數

Type	Model	PR mean	PR std
rule based	ADF-5	0.2920	0
	ADF-10	0.4390	0
多對一 (模型分離)	Random Forest	0.3348	0.018322
	Lasso	0.3960	0.040249
	Ridge	0.4300	0.048990
	TCN	0.3796	0.008905
	DNN	0.3738	0.009834
	LSTM	0.3998	0.010243
	GraphWaveNet	0.4013	0.012307
	STGCN	0.4354	0.036004
多對一 (模型統一)	GraphWaveNet	0.4010	0.016593
	STGCN	0.4298	0.027874
全部對全部	GraphWaveNet	0.4600	0.022572
	STGCN	0.3624	0.043730

表 4.12: 正常感測器與異常感測器其巡檢當日的 R2-score 平均值

Type	Model	正常感測器 群 R2 平均值	異常感測器 群 R2 平均值	正常群與異 常群 R2 差值
多對一 (模型分離)	Random Forest	0.7574	0.6156	0.1418
	Lasso	0.8335	0.6778	0.1557
	Ridge	0.8215	0.6005	0.2210
	TCN	0.8452	0.7011	0.1441
	DNN	0.8320	0.6939	0.1381
	LSTM	0.8467	0.7002	0.1465
	GraphWaveNet	0.8159	0.6532	0.1627
	STGCN	0.7671	0.5630	0.2041
多對一 (模型統一)	GraphWaveNet	0.7764	0.5570	0.2194
	STGCN	0.8233	0.6290	0.1943
全部對全部	GraphWaveNet	0.7563	0.4842	0.2721
	STGCN	0.7566	0.5703	0.1863

4.3 實驗環境介紹

本實驗全程以 python 語言撰寫，版本為 3.6.9，運行環境在 Ubuntu 18.04.3 作業系統上，圖形處理器型號為 GeForce GTX 1080。實驗的模型中，Random Forest、Lasso、Ridge 模型是使用 scikit-learn 機器學習函示庫，TCN、DNN、LSTM、GraphWaveNet、STGCN 模型則是基於 pytorch 深度學習框架撰寫而成的。而資料清洗、時間滾動、產生圖形時序資料等其他實驗流程主要是使用 Numpy 和 Pandas 函示庫完成的。

五、 結論與未來展望

5.1 結論

空氣品質感測器的 pm2.5 時間空間資料，其往往是非線性的高維度分布，要從資料中找出異常數值，以統計基礎或規則基礎方法是很難難精準地找出。本論文中我們提出基於深度時空圖模型異常偵測框架來偵測出異常的 pm2.5 數值，藉此來找到異常的空氣品質感測器。

經由實驗結果，我們發現基於深度時空圖模型異常偵測框架，與其他機器學習方法以及統計規則異常偵測方法相比，無論是排序後的整體 ROC curve 分數，或是前 10 至 50 個感測器其找到有問題感測器精準率比較，基於深度時空圖模型異常偵測框架的結果都來的比較好。

我們提出的異常偵測框架，針對每個感測器，同時考慮其 pm2.5 值的時序資訊，以及同一時間點周圍感測器偵測到 pm2.5 的空間資訊，再加上兩兩之間的距離這些特徵，因此即使感測器有突發性污染源，也可以從周圍感測器的 pm2.5 數值來清楚分辨是周圍有突發性污染源還是感測器偵測到的 pm2.5 數值出現異常，藉此來進一步判斷感測器是否為異常。

同樣我們也在多對一訓練中進行模型分離和模型統一兩者比較，我們認為模型分離適用於空汙感測器群其分布位置的地理資訊是差異很大的，模型統一方式適用於空汙感測器群其分布位置的地理資訊是相似的。

此外我們也對深度時空圖模型異常偵測框架進行全部對全部和多對一兩種不同訓練資料的進一步比較，可以發現基於 GraphWaveNet[1] 訓

表 5.1: 不同模型訓練方式可以運用的的場景

Type	遷移學習	適合運用的情形	適用偵測範圍
模型統一	是 (不需要微調)	判斷單一感測器 是否有故障情形	區域範圍且 地理資訊相似
模型獨立	是 (需要微調)	判斷單一感測器 是否有故障情形	區域範圍且 地理資訊差異大
全部對全部	否	大範圍感測器數目中 找出有問題感測器群	全國範圍

練模型在大範圍的圖形結構中，其可以從 pm2.5 過往資料中找到更精確的時空相關性，在異常偵測中表現也比基於 STGCN[2] 訓練來的好。但全部對全部的訓練方式缺點是其所需的硬體資源較多，而多對一方式，雖然其精準度不如全部對全部方式，但其訓練硬體資源較低，且其可以進行遷移學習。假設今天在日本地區同樣要做空汙感測器異常偵測我們可以拿台中地區以多對一訓練好的模型，再對日本當地地區資料做訓練，這是全部對全部訓練方式無法做到的。

表5.1總結不同訓練方式可以運用的情境，多對一訓練方式裡，無論是模型統一還是模型獨立，其都可以做遷移學習，是全部對全部方法無法做到的。同時我們認為多對一方式可以用來判斷單一感測器是否有故障情況，並適用於區域範圍，如果區域範圍的地理資訊差異很大，建議使用模型分離方式，如果區域範圍的地理資訊差相似，建議使用模型統一方式。相反的，全部對全部方式可以從大範圍感測器數目中找出有問題的感測器群，並適用於區域範圍。巡檢人員可以根據詢檢地點的感測器分布狀況屬於那一種類型來選擇合適的模型訓練方式。

5.2 未來展望

本篇實驗評量模型指標的 precision@n, Recall@n, 都是以每個感測器巡檢日當天的測試資料 (1440) 算出，未來我們也可以一起考慮巡檢日的前幾天 pm2.5, 形成多筆測試資料，分別取出每份測試資料中 R2-score

最差的 n 個感測器，再從這些感測器中，去看實際有問題的感測器有幾個，藉此來判斷模型好壞。

本篇研究使用圖卷積和時間卷積來學習 $\text{pm}_{2.5}$ 值時空分布，而我們的圖形結構是採用無向全連接方式，其代表兩兩感測器與感測器我們認為是有互相影響，在實際環境中，可能受到風向影響造成其影響是單方面，可是受限空氣品質感測器無法偵測其所在地點的風向，因此無法根據每天的風向變化來動態調整輸入模型的圖形結構。

但我們可以根據季風方向來做圖形結構調整，例如台灣夏天是吹西南季風，其位在緯度較南的感測器地點會影響緯度較北的感測器地點，而緯度較北的感測器不會影響緯度較南感測器，因此圖形結構就會具有方向性，而不是無向的，反之台灣冬天吹的是東北季風，上述的影響關係就會呈現相反，因此圖形結構的方向性會隨者季節動態改變，根據感測器所在的季節時間，調整圖形結構並輸入至模型。

此外，某些感測器間可能受到地形阻隔影響，之間是沒有交互影響，因此在圖形結構上某些節點與節點之間是不具備邊 (Edge) 的。總結上述，我們希望未來能加入更多地理資訊像是季風、地形，來得到更精確的圖形結構資訊以此來增加異常偵測的準確率。

參考文獻

- [1] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph wavenet for deep spatial-temporal graph modeling,” *arXiv preprint arXiv:1906.00121*, 2019.
- [2] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” *arXiv preprint arXiv:1709.04875*, 2017.
- [3] 行政院交通環境資源處, “空污感測物聯網應用於環保稽查推動成果.” <https://www.ey.gov.tw/Page/448DE008087A1971/4d1b964c-9294-4505-8814-6d14d57ae05d>.
- [4] 行政院環保署, “空氣品質感測器架構圖.” <https://img.ltn.com.tw/Upload/news/600/2017/07/03/118.jpg>.
- [5] S. Glantz and B. Slinker, *Primer of Applied Regression & Analysis of Variance*, ed. McGraw-Hill, Inc., New York, 2001.
- [6] L.-J. Chen, Y.-H. Ho, H.-H. Hsieh, S.-T. Huang, H.-C. Lee, and S. Mahajan, “Adf: An anomaly detection framework for large-scale pm_{2.5} sensing systems,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 559–570, 2017.
- [7] Z. Qi, T. Wang, G. Song, W. Hu, X. Li, and Z. Zhang, “Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2285–2297, 2018.
- [8] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [9] Y.-S. Chang, H.-T. Chiao, S. Abimannan, Y.-P. Huang, Y.-T. Tsai, and K.-M. Lin, “An lstm-based aggregated model for air pollution forecasting,” *Atmospheric Pollution Research*, vol. 11, no. 8, pp. 1451–1463, 2020.
- [10] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” *arXiv preprint arXiv:1707.01926*, 2017.
- [11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

- [12] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.
- [13] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.
- [14] M. Deza and E. Deza, “Encyclopedia of distances,” 2014.
- [15] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [16] W. N. Anderson Jr and T. D. Morley, “Eigenvalues of the laplacian of a graph,” *Linear and multilinear algebra*, vol. 18, no. 2, pp. 141–145, 1985.
- [17] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” *arXiv preprint arXiv:1606.09375*, 2016.
- [18] A. Micheli, “Neural network for graphs: A contextual constructive approach,” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.
- [19] J. Atwood and D. Towsley, “Diffusion-convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1993–2001, 2016.
- [20] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, “Language modeling with gated convolutional networks,” in *International conference on machine learning*, pp. 933–941, PMLR, 2017.
- [21] 行政院環境保護署, “認識空污感測物聯網.” https://airtw.epa.gov.tw/CHT/Encyclopedia/AirSensor/AirSensor_2.aspx.
- [22] J. A. Hanley and B. J. McNeil, “A method of comparing the areas under receiver operating characteristic curves derived from the same cases,” *Radiology*, vol. 148, no. 3, pp. 839–843, 1983.
- [23] K. H. Zou, A. J. O’Malley, and L. Mauri, “Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models,” *Circulation*, vol. 115, no. 5, pp. 654–657, 2007.
- [24] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
- [25] T. Ho, “A data complexity analysis of comparative advantages of decision forest constructors,” *Pattern Anal. Appl.*, vol. 5, pp. 102–112, 06 2002.

- [26] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
- [27] L. Breiman, “Better subset regression using the nonnegative garrote,” *Technometrics*, vol. 37, pp. 373–384, 1995.
- [28] D. E. Hilt and D. W. Seegrist, *Ridge, a computer program for calculating ridge regression estimates*, vol. 236. Department of Agriculture, Forest Service, Northeastern Forest Experiment ..., 1977.
- [29] M. H. Gruber, *Improving efficiency by shrinkage: the James-Stein and ridge regression estimators*. Routledge, 2017.
- [30] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [31] Y. Chauvin and D. E. Rumelhart, *Backpropagation: theory, architectures, and applications*. Psychology press, 1995.
- [32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [33] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.

附錄 A 實驗程式碼

本實驗室的程式碼公開於 Github 軟體原始碼代管服務平台

連結網址: <https://github.com/ncu-dart/Anomaly-Detecting-Framework-Base-on-Deep-Spatial-Temporal-Graph-Model>

詳細操作方法請參閱 README.md