

大專新進人員訓練工作報告

姓 名： 王翔禾

單位： 台化工務部自控處

報告類別： ☐ 輪班訓練 ☒ 基層實務工作訓練第（三）次報告

訓練部門： 台化工務部自控處智能專案組

起迄日期： 2023/03/14~2023/06/14

工作項目： 新港公用廠 SK4 淨發電量 AI 模型修模及太陽能數據監視系統自行開發工程案

報告項目： SK4 淨發電量 AI 模型修模及太陽能數據監視系統開發工程

評 核 (評 語)	
(2) 一 級 主 管	(1) 二 級 主 管
<p>王翔禾評估AI模型能加入最佳參數選擇技術，縮短調參時間，提升建模效率；並開發太陽能當路平台設計取捨收集程式增加斷線偵測及備用程度，功能對排車科學工程師高度難得，用字程度令人嘉許！</p> <p align="right">羅美達 6/18</p>	<p>1. 透過 SK4 淨發電量 AI 模型修模，學習 AI 建模技術；另外，經辦設計太陽能數據監視系統，學習設備交訊及資料收集的程式設計技術。</p> <p>2. 該員學習態度積極認真，表現優異。</p> <p align="right">職：曹志旭 6/8</p>
(4) 經 營 主 管	(3) 經 理 室
<p>主動學習，態度積極，用心投入，值得嘉許。</p> <p align="right">張 瑞 6/13</p>	<p align="right">林建忠 6/12</p>

目錄

一、 前言.....	2
二、 新港公用廠 SK4 淨發電量 AI 模型修模.....	4
2-1 定義問題.....	4
2-2 本案目標.....	5
2-3 資料盤點與清理.....	6
2-4 離群值與異常值.....	8
2-5 建立模型及參數調整.....	9
2-6 模型訓練結果呈現.....	12
三、 太陽能數據監視系統自行開發工程案.....	14
3-1 太陽能數據監視系統系統硬體架構.....	15
3-2 太陽能數據監視系統系統軟體架構.....	16
3-3 太陽能數據監視系統資料庫結構.....	17
3-3-1 泰山機房資料庫伺服器資料庫結構.....	17
3-3-2 現場工作站工業電腦資料庫結構.....	19
3-4 太陽能數據監視系統檔案介紹.....	20
3-5 綠電網頁介接.....	29
四、 工作心得.....	31
五、 附錄.....	33

一、前言

職目前承辦工作為「新港公用廠 SK4 淨發電量 AI 模型修模」及「太陽能數據監視系統自行開發工程案」。

「新港公用廠 SK4 淨發電量 AI 模型修模」此模型目的是透過 AI 技術的應用，找出冷卻水塔中「風車」及「泵浦」開台數的最佳操作組合，進一步預測汽機的淨發電量，使淨發電量最大化，以實現最大的發電利益。儘管汽發電機在發電過程中，往往會受到現場操作條件、設備維護狀況等諸多因素的影響，但是透過 AI 模型的輔助，可以更好地瞭解這些影響因素，並且提供相應的操作建議，從而更好地管理和控制淨發電量的變化，而本案則是由於模型上線已將近兩年，預測準確度下降，因此需用最新的資料對模型進行改善。

「太陽能數據監視系統自行開發工程案」本案係配合本公司未來太陽光電案場的增加，為節省建置及運維成本，擬自行開發監控系統。先以新港公用廠-保養廠屋頂太陽能案場為目標，對監控系統進行軟體開發，以替換原本由協力廠商(盛齊綠電)開發的軟硬體架構。

首先，開發 Solaredge 逆變器資料回傳通訊程式，以即時監視太陽能發電系統的運行狀況，確保系統的穩定性；其次，開發日照計和溫度計資料回傳通訊程式，以精確地掌握太陽能發電的環境因素。

為提供太陽能數據監視平台，須建置中央電腦主機資料庫伺服器，並開發相應的數據介接和現場電腦工作站資料暫

存/備份程式，以管理和儲存系統的數據，確保系統運行的可靠性。另外，開發系統主機擷取現場工作站 4G 網路浮動 IP 的程式，以方便遠端系統監控和管理。

最後，為了方便查看系統運行狀況，規劃介接逆變器、日照計和溫度計的資料至已開發的綠電網頁；考慮到未來仍有多個案場增設，本部也設計程式能自動化更新成最新版本功能，這將使程式更新和維護更加及時和高效。

通過以上的軟體自主開發及佈署，將可有效降低監控系統建置及運維費用，並提高太陽能發電系統的效率、穩定性和可靠性。

本季工作心得，以上述二案為主軸，詳如後續說明。

二、新港公用廠 SK4 淨發電量 AI 模型修模

2-1 定義問題

新港 SK4 機組給水系統透過多段加熱工序後產生蒸汽，主要作為推動汽發電機動力，也提供部份 6K/12K/18K 蒸汽送現場使用，經過抽汽後剩餘的高壓蒸汽用來推動汽發電機發電，低壓蒸汽再冷凝成水，周而復始，主蒸汽做功後在冷凝器中與冷卻水熱交換，將汽機末段的低溫蒸汽再冷凝成水，使汽機末段保持一定的真空度，但「隨著熱交換效率及真空度的不同，汽發電機發電效率也有所不同」。

以往隨著夏天與冬天氣溫不同，現場會增開或者減開冷卻水塔風車及泵浦，但風車與泵浦開台數變化後會改變冷凝器冷卻水的熱交換效能，導致汽機排汽溫度與真空度隨之不同，進而影響汽發電機的發電量。

由於各製程數據錶點並未即時計算，其結果未必能符合最佳運轉效益點，因此「冷卻水系統用電量」及「發電量」兩者間存在操作優化空間。

因此開發汽發電機淨發電量預測模型，提供最佳的操作建議，希望透過 AI 找出冷卻水塔的「風車」及「泵浦」最佳開台數操作組合，使淨發電量最大化，獲取最大發電利益。

2-2 本案目標

新港 SK4 機組淨發電量操作優化模型已上線超過兩年。然而，由於訓練資料僅包含 2020/1/1 至 2021/4/30 的數據，無法全面考慮上線後的設備老化、元件更換和環境變動等因素。因此造成目前的淨發電量的預測結果不夠精確。

為解決這個問題，需要對 SK4 機組淨發電量操作優化模型進行修正。首先，需要收集最新的數據，並選用最近一次定檢後的 2022/6/1 至 2023/4/15 的最新數據進行建模。

此外，還需要比較各種不同的機器學習方法，以挑選出最佳模型。通過使用最新數據和最佳模型訓練模型，預測可靠的淨發電量預測結果，進一步回推風車和泵浦開台數的最佳操作建議。同時，為更全面地考慮影響淨發電量的因素，還需進一步處理數據，例如刪除缺失值或離群值等。這些優化措施能夠提高模型的準確性和可靠性，進一步優化 SK4 機組的淨發電量操作。

2-3 資料盤點與清理

本案預測目標為毛發電量及汽機排汽溫度，經重新收集資料期間(2022/6/1 至 2023/4/15) ，收集頻率為每分鐘一筆，總共 655,872 筆，特徵變數選取及模型建立則延續先前建模經驗，，需分兩個模型建模，模型一選出 6 個特徵變數預測汽機排汽溫度，並利用模型一的預測結果當作模型二的其中一個特徵變數建立模型二，總共使 8 個特徵變數預測淨發電量，如下圖 2-1，而本次也再次進行相關性係數分析作為輔助判斷，與先前(舊模型上線之訓練資料)做對比數值並未有明顯改變，如下圖 2-2、圖 2-3 為相關性係數分析結果。

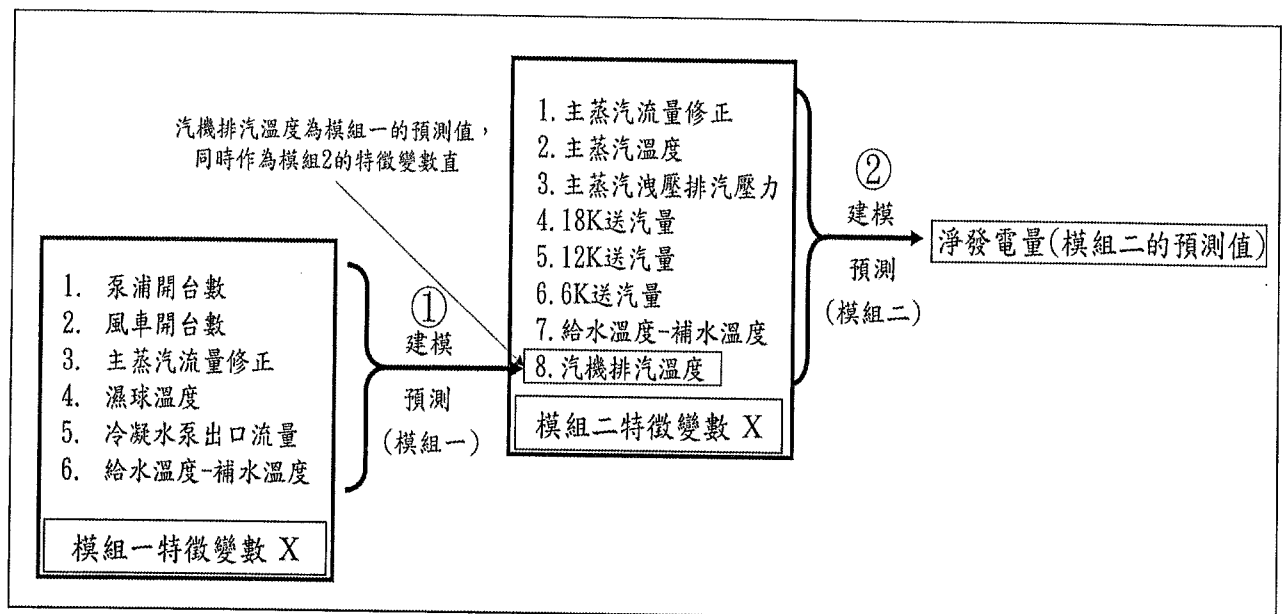


圖 2-1、SK4 機組淨發電量操作優化模型原始特徵變數

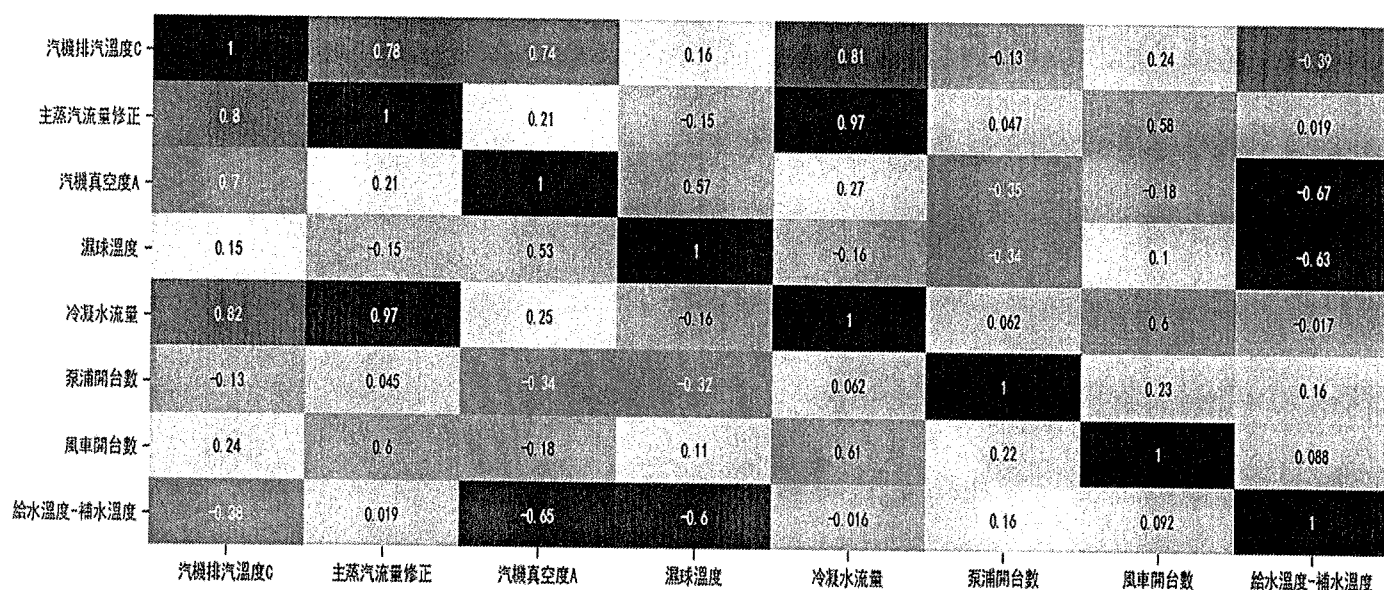


圖 2-2、SK4 淨發電量優化模型變數相關性係數分析圖
(本次數據 2022/6/1~2023/4/15)

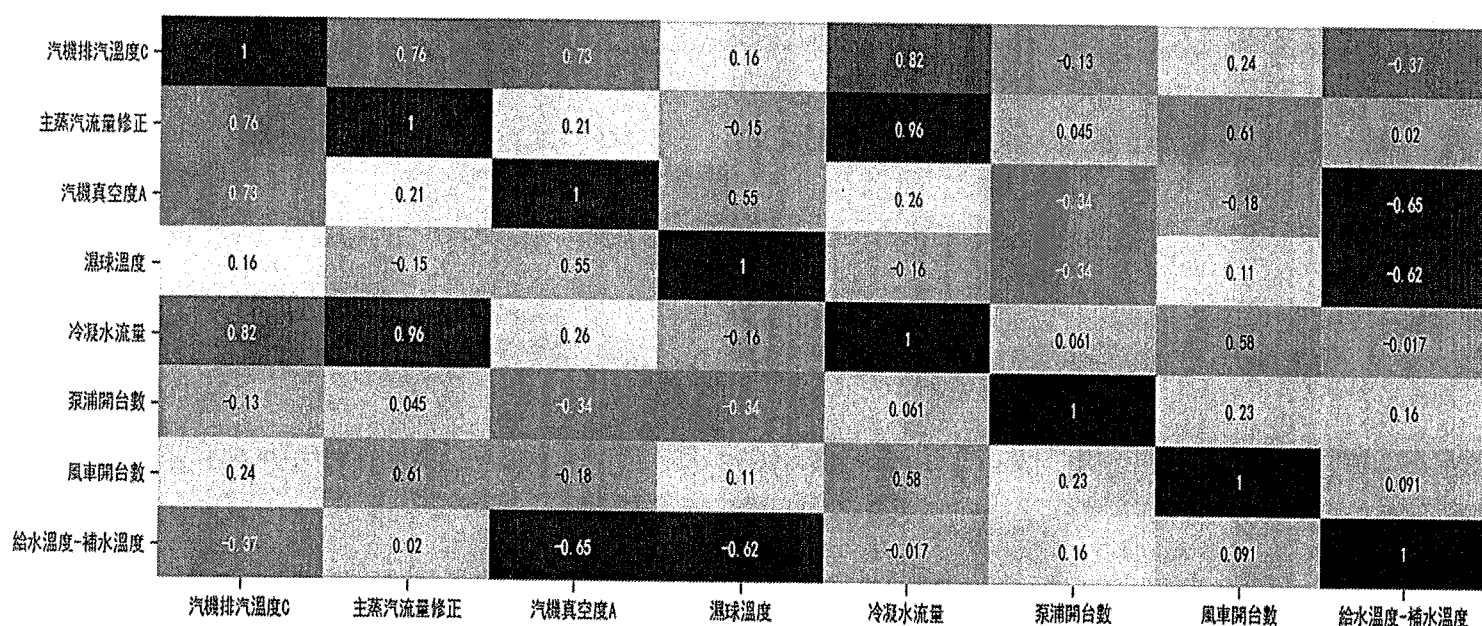


圖 2-3、SK4 淨發電量優化模型變數相關性係數分析圖
(舊數據 2020/1/1~2021/1/30)

2-4 離群值與異常值

在蒐集完資料後，需要進行資料的離群值與異常值處理，在與製程人員檢討後須去除下列幾種異常值

- 將 PI 資料庫無法正常顯示的數據(如 I/O Timeout、Not Connect)刪除，共 8 筆。
- 將發電量低於 60MW 及主蒸汽量小於 350T/H 的資料刪除(非正常運轉狀態)，共 812 筆，

經資料清洗後，由 655,872 筆降至 655,052 筆。

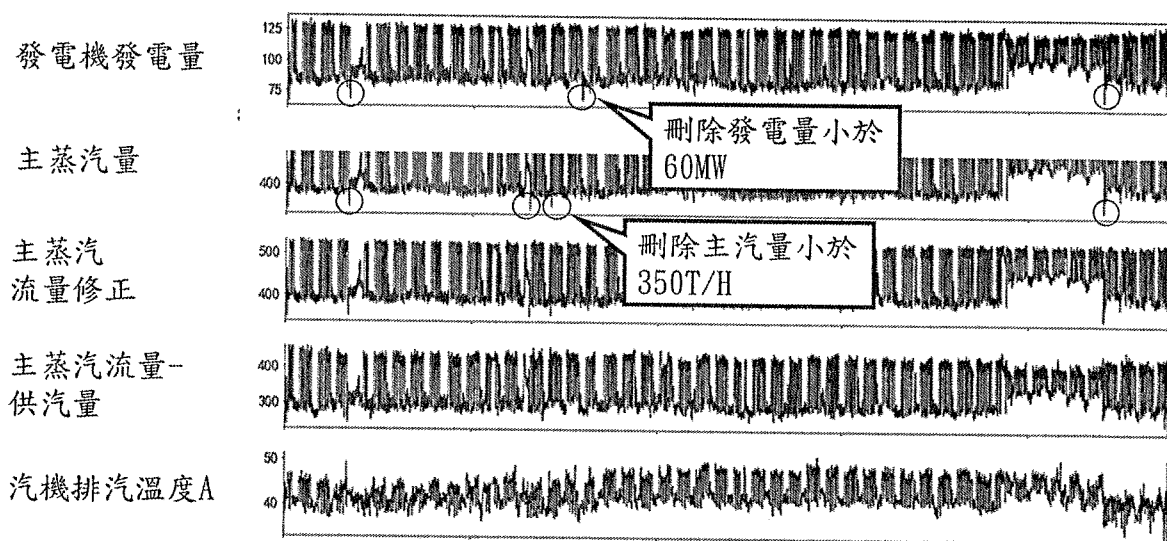


圖 2-4、SK4 淨發電量優化模型變數離群值與異常值

2-5 建立模型及參數調整

本案模型一為汽機排汽溫度預測模型及模型二為淨發電量預測模型，原始模型均選用 Linear-Regression 演算法作為最終建模之演算法，而本次則選用五種常見的機器學習演算法建模做為測試，分別為：Linear-Regression、Ridge-Regression、Lasso-Regression、ElasticNet 及 XGBoost。

選擇使用這些演算法的原因是因為它們都是廣泛應用於回歸問題，並在不同的情況下表現出色，且為配合已上線的程式(使用機器學習演算法)能方便抽換訓練後的模型，因此，此次演算法篩選以機器學習演算法為主，以下是這些演算法的優點和適用情況的總結：

- Linear Regression: 是一種簡單的線性演算法，易於理解和實現。它適用於簡單的回歸問題，其中預測變量和自變量之間呈線性關係。
- Ridge Regression: 是一種正則化線性演算法，可用於控制過度擬合。它適用於多元共線性問題，其中自變量之間存在高度相關性。
- Lasso Regression: 也是一種正則化線性演算法，它具有特徵選擇的能力，可用於選擇最重要的特徵。它適用於高維度數據集，其中大多數特徵對目標變量沒有影響。
- ElasticNet: 結合 Ridge Regression 和 Lasso Regression 的優點，同時具有控制過度擬合和特徵選擇的能力。

- XGBoost: 是一種梯度提升樹演算法，可用於建立高準確度的模型。它適用於大型數據集並具有防止過度擬合的能力。

在建模的過程中，使用差異性較大的演算法作比較之優點在於，可以比較各種演算法的表現和特徵，找出最好的演算法進行建模。此外，使用不同種演算法建模也能瞭解數據的性質，以及哪些演算法對於特定問題或數據集表現得更好。

為了選擇最適當的演算法建立模型，模型一選用汽機排汽溫度的六個特徵值分別使用五種不同演算法建模，模型二選用淨發電量的八個特徵值也用五種不同的演算法建模，並將資料集以 9:1 的方式切分為訓練集及測試集。同時，在比較這些模型的訓練表現時，使用了 RandomizedSearchCV(隨機參數搜索)方法來選擇最佳的參數組合。

目前常用的參數優化方法有兩種，一種是 RandomizedSearchCV(隨機參數搜尋)，另一種是 GridSearchCV(網格參數搜索)，因 RandomizedSearchCV 具有以下優點因此本案選用其來找出最佳的參數組合，相關優點說明如下：

- 時間效率：相對於 GridSearchCV，RandomizedSearchCV 通常需要測試的參數組合數量較少，因此在搜索過程中更加高效。這對於較大的參

數空間和複雜的模型來說尤為重要。

- 探索性更全面: RandomizedSearchCV 在給定的參數範圍內隨機選擇參數組合，這樣可以更全面地探索參數空間，增加了發現最佳組合的機會。而 GridSearchCV 則是對有限空間的參數組合進行計算，不一定能找到最優解且浪費大量的時間資源。
- 可控性：通過設定隨機選擇的次數，可以根據時間和計算資源的限制來調整搜索範圍。可以通過控制參數搜索的反覆運算次數，平衡搜索的廣度和深度。

在此案中，使用 RandomizedSearchCV 方法可以在訓練過程中自動搜索五種演算法建立的模型找出最佳參數組合，該方法可以減少手動調參的工作量，同時更全面地搜索參數空間，以找到更好的組合。

最後分別再比較五種不同演算法之測試集決定係數(R-Squared)及平均誤差百分比 (MAPE)。從下方表 2-1、表 2-2 可知，最終比較汽機排汽溫度預測模型及毛發電量預測模型二者所選用之演算法，均顯示出使用 XGBoost 演算法之 R-Squared 較高且 MAPE 數值較小，因此最終均選用 XGBoost 演算法建模，並上線應用。

表一： 汽機排汽溫度建模選 用之演算法比較表	Linear- Regression	Ridge- Regression	Lasso- Regression	ElasticNet	XGBoost
R2	0.4128	0.7856	0.7712	0.9243	0.9716
MAPE	0.2117	0.0914	0.11077	0.0584	0.017

表 2-1、汽機排汽溫度建模選用演算法比較表

表二： 毛發電量建模選用 之演算法比較表	Linear- Regression	Ridge- Regression	Lasso- Regression	ElasticNet	XGBoost
R2	0.2379	0.7436	0.7601	0.8927	0.9881
MAPE	0.3674	0.1187	0.0873	0.0584	0.015

表 2-2、淨發電量建模選用演算法比較表

2-6 模型訓練結果呈現

經過上述不同演算法進行建模並挑選出最適化模型，現已成功將模型佈署至先前建立的儀表板，供使用者查看預測結果。透過圖 2-5、2-6 的呈現，我們可以觀察到模型的預測結果與實際值之間非常接近。

綜上所述，模型在上線應用中展現出優異的預測結果，與實際值非常貼近，如圖 2-7 所示。這表明我們的模型在準確性和穩定性方面取得不錯成果，並具有良好的實用性和可靠性，這有助於幫助操作人員，做出準確的預測和決策。

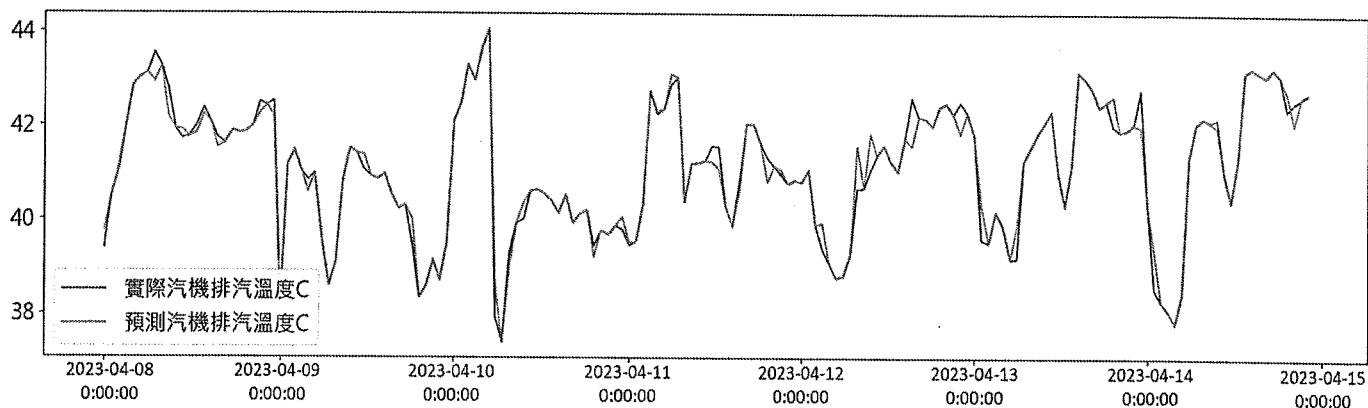


圖 2-5、汽機排汽溫度模型預測/實際值比較圖

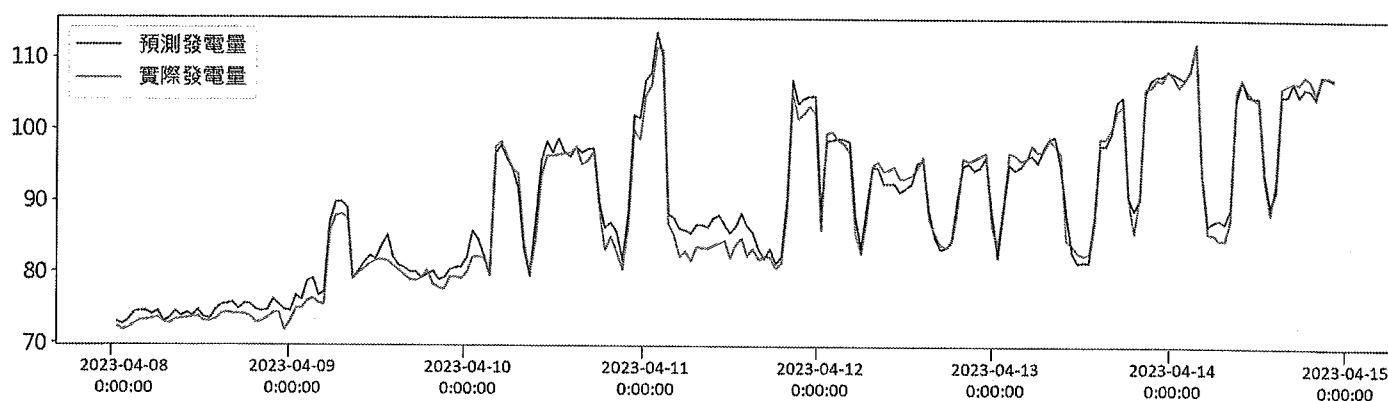


圖 2-6、淨發電量模型預測/實際值比較圖

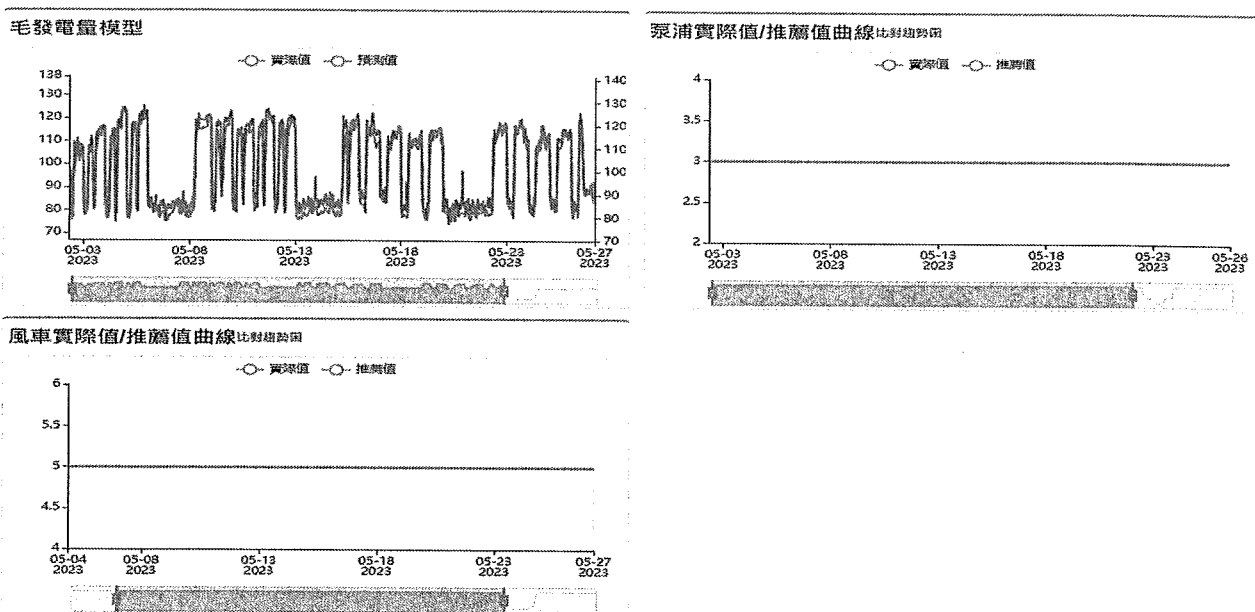


圖 2-7、風車、泵浦 實際值/推薦值曲線比較圖

三、太陽能數據監視系統自行開發工程案

因本公司太陽光電案場陸續增加，為節省建置及運維成本，各案場發電監視系統擬採自行開發，本案執行太陽能案廠自行開發工程之軟體設計部分，以新港公用廠-保養廠屋頂為試行案場進行下列軟體開發監控系統，將監控系統由原本協力廠商盛齊綠電所開發的軟硬體架構替換成台化自主開發的監控系統，此案主要開發的程式部份有下列幾項：

- Solaredge 逆變器資料回傳通訊程式開發，以便即時監控太陽能發電系統運行狀況。
- 日照計、溫度計資料回傳通訊程式開發，以更精確地掌握太陽能發電的環境因素。
- 中央電腦主機資料庫建置、數據介接、現場電腦工作站資料緩存/備份程式開發，以實現系統的數據管理與儲存，保證系統運行的可靠性。
- 系統電腦主機擷取現場工作站 4G 浮動 IP 軟體開發，以便更好地進行系統監控和管理。
- 逆變器、日照計、溫度計資料介接至已開發之綠電網頁，以便於查看各案場系統運行狀況。
- 程式更新自動化設計，考量到之後會有多個案場，須將程式更新自動化，以便更及時地更新和維護系統。

3-1 太陽能數據監視系統系統硬體架構

以下為本案工程系統架構圖，如圖 3-1-A 及圖 3-1-B 所示，圖 3-1-A 為既有系統架構、圖 3-1-B 為自主開發系統架構，自主開發系統架構將於泰山機房建立資料庫伺服器，接收來自各案場的逆變器、日照計、溫度計等運轉數據，最後介接至綠電網頁。

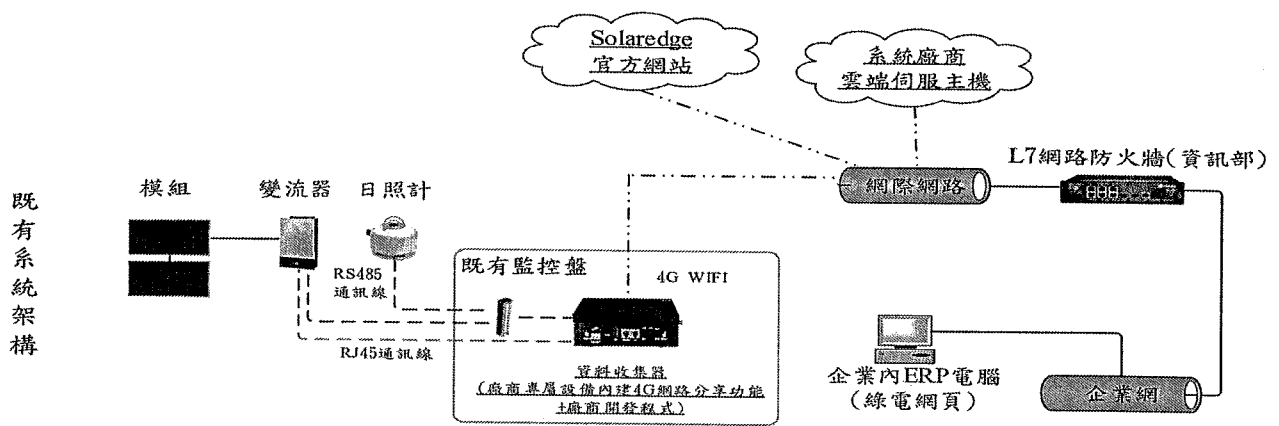


圖 3-1-A、太陽能數據監視既有系統架構圖

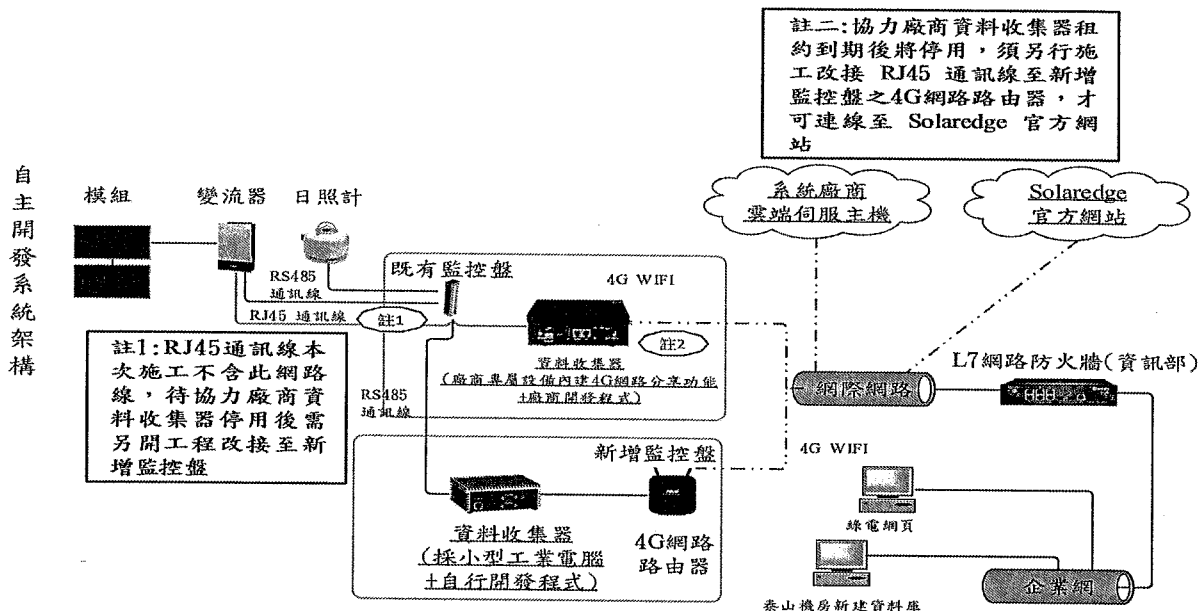


圖 3-1-B、太陽能數據監視系統自主開發系統架構圖

3-2 太陽能數據監視系統系統軟體架構

此案規劃之軟體設計流程如下圖 3-2 所示，藉由現場工作站(採用小型工業電腦安裝自行開發程式並搭配 4G 路由器所組成)抓取現場逆變器及感測器(溫度計及日照計)之數據，並將數據儲存至泰山機房新建資料庫伺服器中，另外也開發監控程式監控現場工作站的工作狀態(包含資料庫通訊狀態、資料儲存狀態及網路通訊狀態等)，以達到穩定收集資料的目的，下圖中的數字編號代表所用到的檔案，將在後續 3-4 小節依序介紹。

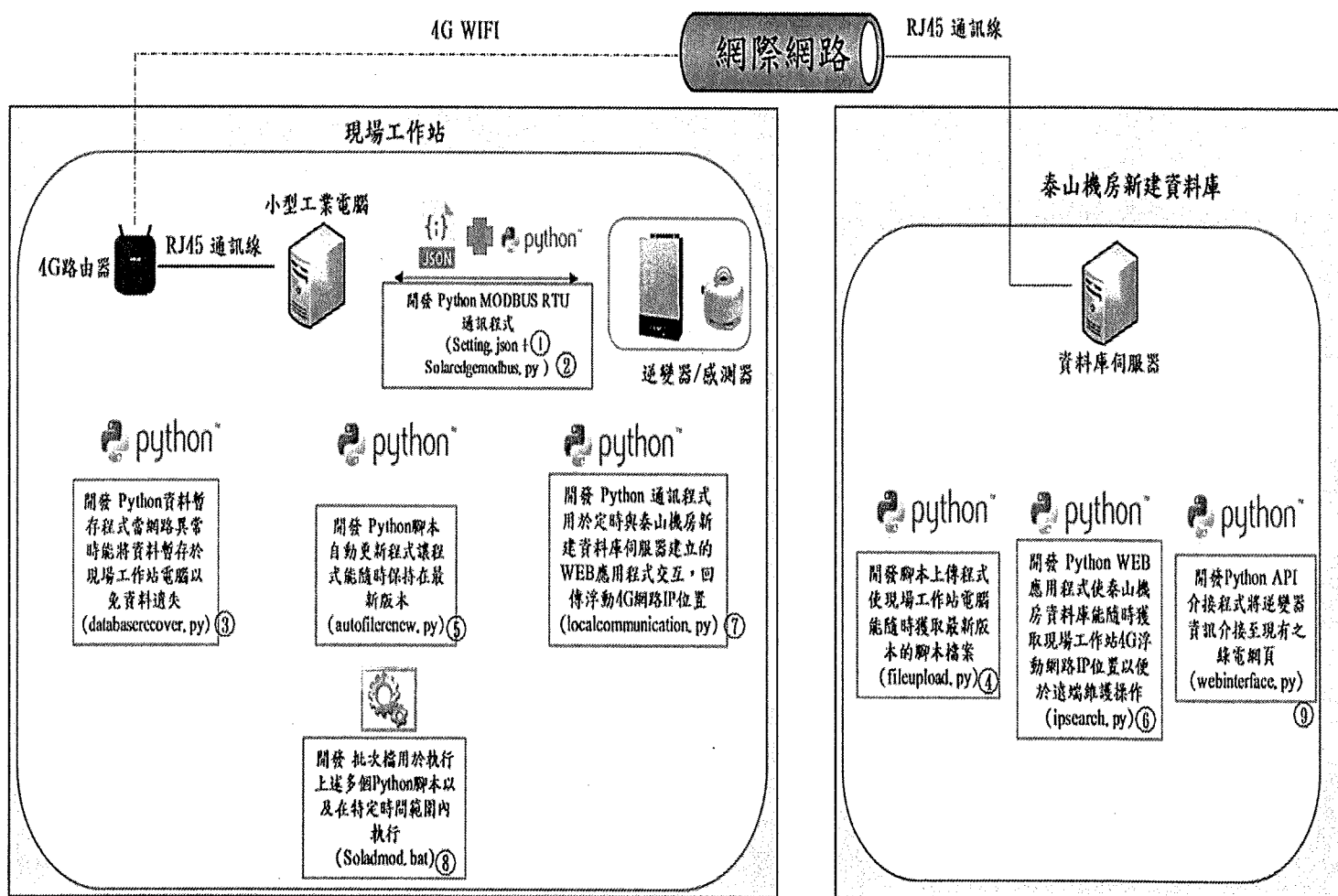


圖 3-2、太陽能數據監視系統軟體架構圖

3-3 太陽能數據監視系統資料庫結構

3-3-1 泰山機房資料庫伺服器資料庫結構

此案須於泰山機房新增資料庫伺服器，並新增一資料庫及三張資料表，且此資料庫需要一對外的固定 IP 提供各案廠存取資料庫，其結構如下圖 3-3 所示：

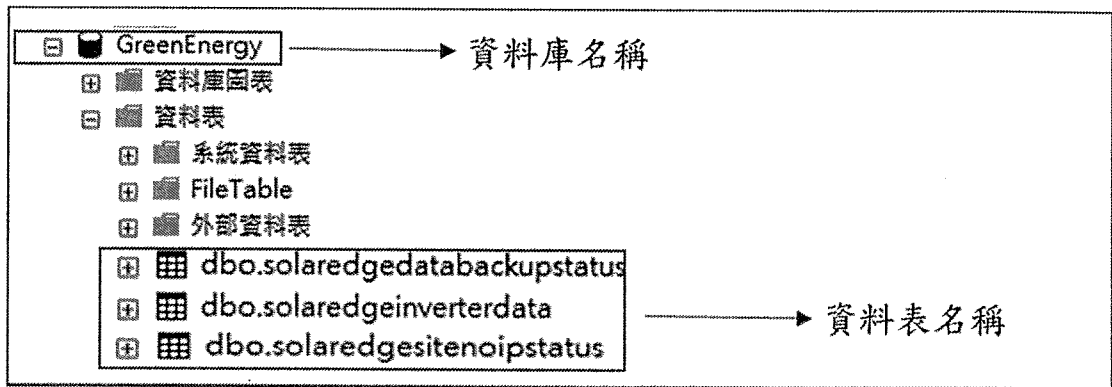


圖 3-3、泰山機房資料庫伺服器資料庫結構

其中 GreenEnergy 為泰山機房資料庫名稱，其內部包含三張表分別為：

- solaredgeinverterdata:
用於存儲現場逆變器、日照計、溫度計資料共包含 49 個資料列，其資料表結構如下表 3-1 所示，其對照意義請參照附錄表 5-1 solaredgeinverterdata 對照表。
- solaredgesitenoipstatus:
用於存儲現場工作站電腦之 ip 位置，以方便遠端連線維護其資料表結構如下表 3-2 所示。
- solaredgedatabackupstatus:
用於存儲現場工作站與之暫存資料回傳至泰山機房資料庫的狀況，如下表 3-3 所示。

列名	id	siteno	datano	datetime	c_ manufacturer	c_model	c_version	c_ serialnumber	c_ deviceaddress	c_sunspec_did
資料型態	Int	Nvarchar(20)	Nvarchar(20)	datetime	Nvarchar(20)	Nvarchar(20)	Nvarchar(20)	Nvarchar(20)	float	float
列名	current	l1_current	l2_current	l3_current	current_scale	l1_voltage	l2_voltage	l3_voltage	l1n_voltage	l2n_voltage
資料型態	float	float	float	float	float	float	float	float	float	float
列名	l3n_voltage	voltage_scale	power_ac	power_ ac_scale	frequency	frequency_ scale	power_ apparent	power_ apparent_scale	power_ reactive	power_ reactive_scale
資料型態	float	float	float	float	float	float	float	float	float	float
列名	power_factor	power_ factor_scale	energy_total	energy_ total_scale	current_dc	current_ dc_scale	voltage_dc	voltage_ dc_scale	power_dc	power_ dc_scale
資料型態	float	float	float	float	float	float	float	float	float	float
列名	temperature	temperature_ scale	status	vendor_status	rrcr_state	active_power_ limit	cosphi	irradiance	backTemp	
資料型態	float	float	float	float	float	float	float	float	float	

表 3-1、solaredgeinverterdata 資料表結構

列名	siteno	ip	datetime	localserverstatus
資料型態	Nvarchar(20)	Nvarchar(20)	datetime	Nvarchar(20)
中英對照	案場名稱	現場工作站 IP位置	紀錄時間	現場工作站狀態

表 3-2、solaredgesitenoipstatus 資料表結構

列名	id	siteno	backupstatus	backupmessage	backuptime
資料型態	Int	Nvarchar(20)	Int	datetime	Nvarchar(20)
中英對照	Id	案場名稱	暫存資料倒回資料庫狀態	暫存資料倒回資料庫訊息	暫存資料倒回資料庫時間

表 3-3、solaredgedatabackupstatus 資料表結構

3-3-2 現場工作站工業電腦資料庫結構

此案須於新港公用廠-保養廠之現場工作站工業電腦中，新增一資料庫及兩張資料表，如圖 3-4 所示

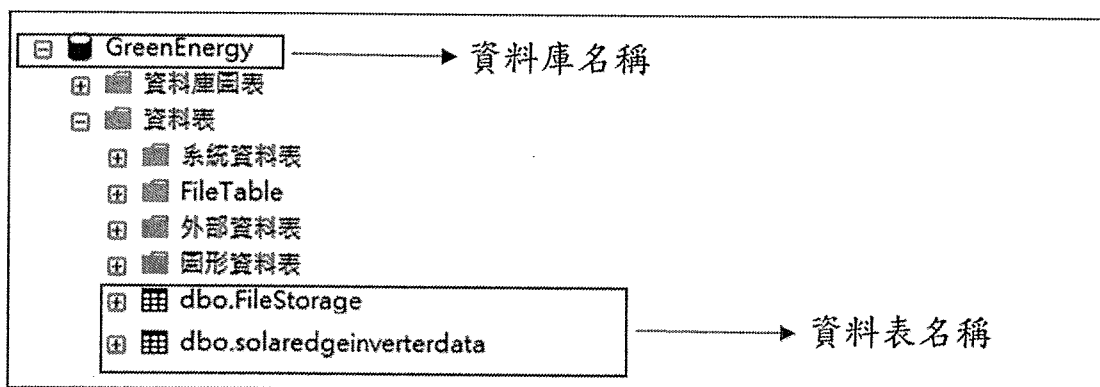


圖 3-4、本地工作站資料庫結構

其中 GreenEnergy 為現場工作站工業電腦所新建的資料庫名稱，其內部包含兩張表分別為：

- solaredgeinverterdata:
用於存儲現場逆變器、日照計、溫度計資料共包含 49 個資料列，其資料表結構與上表 3-1 相同
- FileStorage:
用於儲存現場工作站所需之腳本檔案及檔案更新時間，如下表 3-4 所示

列名	id	siteno	filename	Filecontent	created_at
資料型態	Int	Nvarchar(20)	Nvarchar(20)	varbinary(MAX)	datetime
中英對照	id	案場名稱	檔案名稱	檔案二進位內容	檔案創建時間

表 3-4、FileStorage 資料表結構

3-4 太陽能數據監視系統檔案介紹

本案共開發 7 支不同的執行程式，總共含有 7 個腳本檔案、1 個設定檔、及 1 個批次檔，共 9 個主要檔案，如下表 3-5 所示：

檔案名稱	Setting .json	Solaredgemodbus .py	Databaserecover .py
檔案名稱	autofilerenew .py	Fileupload .py	Ipsearch .py
檔案名稱	Localcommunication .py	Soladmod .bat	webinterface. py

表 3-5、太陽能數據監視系統主要檔案列表

以下將依序介紹各個檔案之作用：

1. 檔案一:setting.json 設定檔，其內容如下圖 3-5 所示

```
{
  "lis": [
    230
  ],
  "siten": "sitenotEST",
  "retries": 3,
  "device": "COM6",
  "baud": 115200,
  "server": "10.90.53.67",
  "user": "sa",
  "password": "Could721",
  "database": "GreenEnergy",
  "backupserver": "127.0.0.1",
  "backupuser": "sa",
  "backuppassword": "N000184003",
  "ipport": 4003,
  "columns": [ ... ],
  "backupday": 90,
  "renewtime": "2023-05-11 17:24:14"
}
```

圖 3-5、setting.json 設定檔內容

此 JSON 檔案包含了程式運行所需的各種設定參數。
下述各個參數的解釋：

- lis: 逆變器列表，代表要讀取的太陽能逆變器的編號。
- siteno: 太陽能案廠的代號。
- retries: 連線和數據庫插入的最大重試次數。
- device: 連接太陽能逆變器的設備名稱或串口號。
- baud: 與逆變器通信時使用的串口串列傳輸速率。
- server: 泰山機房資料庫的 IP 位址或主機名稱。
- user: 泰山機房資料庫資料庫的使用者名稱。
- password: 泰山機房資料庫資料庫的密碼。
- database: GreenEnergy 要將數據插入的資料庫名稱。
- backupserver: 現場工作站資料庫的 IP 位址。
- backupuser: 現場工作站資料庫的使用者名稱。
- backuppassword: 現場工作站資料庫的密碼。
- ipport: 泰山機房資料庫通訊連接埠號。
- columns: 要插入資料庫的欄位列表。
- backupday: 現場工作站資料庫保留數據的天數。
- renewtime: 程式的最後更新時間，用於追蹤和記錄程式的更新歷史。

2. 檔案二：

solaredgemodbus.py 腳本檔，其用於讀取太陽能逆變器數據並將其插入上述 3-2 小節提到的 solaredgeinverterdata 資料表中，其主要步驟包括以下部分：

- 讀取設定檔：從 setting.json 設定檔中讀取連線和配置參數，包括連接太陽能逆變器的設備、串口串列傳輸速率、資料庫連接資訊等。
- 建立 datano 字典：根據設定檔中的數字列表 lis，生成對應的太陽能逆變器資料代碼 datano，以便識別每個逆變器的數據。

- 定義 solarmod 函式：該函式用於讀取單個逆變器的數據並插入資料庫。它通過 solaredge_modbus 模型與逆變器進行通信，並嘗試將數據插入泰山機房資料庫。如果連線失敗，它會嘗試將數據插入本地資料庫作為備份。
- 迴圈執行 solarmod 函式：根據設定檔中的逆變器數字列表 lis，對每個逆變器調用 solarmod 函式，逐個讀取數據並插入資料庫。

總結來說，此份檔案用於自動化太陽能逆變器數據讀取並執行數據插入資料庫功能。如果泰山機房資料庫連線失敗，則將數據插入本地資料庫，其主要程式碼如下圖 3-6 所示。

```
def solarmod(id):
    inverter = solaredge_modbus.Inverter(device=device, baud=baud ,unit=id)
    status = inverter.connect()
    if status:
        print("MODBUS 連接成功")
    else:
        logging.error('MODBUS 連接失敗')
        print("Failed to connect to modbus")
        exit()

    data = inverter.read_all()

    if data:
        print(id,':Inverter 數據讀取成功')
    else:
        logging.error(f'{id}:Inverter 數據讀取失敗')
        print(id,':Inverter 數據讀取失敗')
        inverter.disconnect()
        data={}
        # return
        exit()

    inverter.disconnect()

    data['siteno'] = siteno
    data['datano'] = datano[id]
```

圖 3-6、solaredgemodbus.py 腳本檔主函數程式碼

3. 檔案三: databaserecover.py 腳本檔

其用於將現場工作站之暫存資料倒回泰山機房資料庫，於每日晚上 21:00，執行資料倒回動作，而其執行結果則儲存於上述 3-2 小節 solaredgedatabackupstatus 資料表中，其主要步驟包括以下部分：

- 從工作站資料庫中讀取暫存數據
- 在泰山機房資料庫中檢查數據是否已經存在，如果不存在，則將數據插入主資料庫，並從備份資料庫中刪除該數據
- 如果插入數據或刪除數據的過程中出現異常，將錯誤資訊記錄在日誌檔中
- 從備份資料庫中刪除超過指定天數的數據。

```
# 備份檔案
try:
    # Connect to database A
    conn_a = pymysql.connect(server=server, user=user, password=password, database=database)
    cursor_a = conn_a.cursor()

    # Connect to database B
    conn_b = pymysql.connect(server=backupserver, user=backupuser, password=backuppassword, database=database)
    cursor_b = conn_b.cursor()

    # Read data from database B
    cursor_b.execute("SELECT " + " , ".join(columns) + " FROM solaredgeinverterdata")
    rows = cursor_b.fetchall()

    # Prepare the insert query, 去掉 id
    values = ', '.join(['%s' for _ in range(len(columns))])
    insert_query = f"INSERT INTO solaredgeinverterdata ({', '.join(columns)}) VALUES ({values})"
    delete_query = "DELETE FROM solaredgeinverterdata WHERE datetime = %s AND datano = %s"

    success_insert = True
    success_delete = True
    failis = []

    # Insert data into database A and delete from database B
    for row in rows:
        # Check if the data already exists in database A
        cursor_a.execute("SELECT COUNT(*) FROM solaredgeinverterdata WHERE datetime = %s AND datano = %s", (row[2], row[1]))
        count = cursor_a.fetchone()[0]

        if count == 0:
            try:
                # If the data does not exist, insert it into database A
                cursor_a.execute(insert_query, row)
                cursor_b.execute(delete_query, (row[2], row[1]))
                conn_a.commit()
                conn_b.commit()
                print(f"插入數據成功: {row[2]}-{row[1]}")
            except Exception as e:
                print(f"插入數據失敗: {e}")
                logging.error(f"插入數據失敗: {e}")
                success_insert = False
                success_delete = False
                failis.append((row[2], row[0], row[1]))
        else:
            print(f"數據已存在, 保留{backupday}天')
```

圖 3-7、databaserecover.py 腳本檔主函數程式碼

4. 檔案四：fileupload.py 腳本檔，
其用於將指定的檔 solaredgemodbus.py、
databaserecover.py 和 setting.json 等檔案轉換為二進制
文件，儲存至上述 3-2 小節所提到的 FileStorage 資料表
中。

```
try:
    # Connect to database A
    conn_b = pymysql.connect(server=backupserver, user=backupuser, password=backuppassword, database=database)
    cursor_b = conn_b.cursor()
    timestamp = datetime.datetime.now()

    with open("setting.json", "r") as f:
        config = json.load(f)
    print(timestamp.strftime("%Y-%m-%d %H:%M:%S"))
    config["renewtime"] = timestamp.strftime("%Y-%m-%d %H:%M:%S")

    with open("setting.json", "w") as f:
        json.dump(config, f, indent=4)

    files = [
        'solaredgemodbus.py',
        'databaserecover.py',
        'setting.json'
    ]

    for filename in files:
        # Read file
        with open(filename, 'rb') as file:
            file_content = file.read()

        # Insert file into FileStorage table
        cursor_b.execute(
            "INSERT INTO FileStorage (siteno, filename, filecontent, created_at) VALUES (%s, %s, CONVERT(varbinary(max), %s), %s)",
            (siteno, filename, file_content, timestamp)
        )
        print(f'{filename} finished successfully')
```

圖 3-8、fileupload.py 腳本檔主函數程式碼

5. 檔案五：autofilerenew.py 腳本檔

其用於讓現場工作站電腦能從泰山機房資料庫的 FileStorage 資料表中獲取最新腳本檔並將其保存到本地目錄，讓程式能隨時保持在最新版本。

```
# Connect to database B
conn_b = pymysql.connect(server=backupserver, user=backupuser, password=backuppassword, database=database)
cursor_b = conn_b.cursor()

cursor_b.execute("SELECT created_at FROM FileStorage WHERE siteno = %s", (siteno,))
timestamp = cursor_b.fetchone()[0]
print('timestamp', str(timestamp.strftime("%Y-%m-%d %H:%M:%S")))
timestamp=str(timestamp.strftime("%Y-%m-%d %H:%M:%S"))

if timestamp != renewtime:
    print('檔案需更新')
    # Define the output directory where the files will be saved
    output_dir = os.getcwd()+ '/'
    print(output_dir)

    # Retrieve the file records from the FileStorage table
    cursor_b.execute("SELECT filename, filecontent FROM FileStorage WHERE siteno= %s", (siteno,))
    files = cursor_b.fetchall()

    # Iterate over the files and save them to the output directory
    for filename, filecontent in files:
        print(filename)
        with open(output_dir + filename, 'wb') as file:
            file.write(filecontent)

    # Close the connection
    conn_b.close()
    print('檔案更新完畢')
else:
    print('檔案不需更新，已為最新檔')
```

圖 3-9、autofilerenew.py 腳本檔主函數程式碼

6. 檔案六：ipsearch.py 腳本檔

用於回傳現場工作站之 IP 位置，由於現場工作站電腦之網路為浮動 IP，無法以傳統固定 IP 之形式進行遠端維護，因此需要在泰山機房資料庫建立一個 WEB 應用程式讓現場工作站電腦能定期與此 WEB 應用程式交互，其主要步驟如下：

- 建立 WEB 應用程式，網址為
<http://{泰山機房資料庫固定IP}:4003>

- WEB 應用程式接獲現場工作站電腦之 get 請求後，回傳現場工作站之 IP 位置，
- 更新伺服器的狀態，此腳本會定時檢索上述 3-2 小節提到的 solaredgesitenoipstatus 資料表中的所有案場資料。計算當前時間與案場 IP 位置最後更新時間之的差異，如果差異超過一定時間，則將 localserverstatus 更新為 0(異常)，否則更新為 1(正常)。

```
@app.get("/")
def get_client_ip(request: Request):
    client_host = request.client.host

    return {"ip": client_host}

if __name__ == "__main__":
    # 使用 APScheduler 設定每五分鐘執行一次檢測任務
    scheduler = BackgroundScheduler()
    scheduler.add_job(update_server_status, 'interval', minutes=10)
    # scheduler.add_job(update_server_status, 'interval', seconds=10)
    scheduler.start()

    # 啟動 FastAPI 伺服器
    uvicorn.run(app, host="0.0.0.0", port=ipport)
```

圖 3-10、ipsearch.py 腳本檔主函數程式碼

7. 檔案七：localcommunication.py 腳本檔

用於定時與 ipsearch.py 建立的 WEB 應用程式交互，並將所得到的浮動 IP 位置儲存 solaredgesitenoipstatus 資料表中，其主要步驟如下：

- 定時向指定的上述提到的 WEB 發出 HTTP GET 請 (http://{泰山機房資料庫固定 IP}:4003)求，並獲取返回的 IP。

- 將獲取的 IP 位置存儲到 SQL Server 資料庫中。它首先建立到資料庫的連接，然後檢查是否已存在具有相同 siteno 的資料。如果已存在，則刪除原有的資料。然後，它將 siteno、ip、create_time 和 1（表示本地伺服器狀態正常）插入到 solaredgesitenoipstatus 表

```
def visit_server():
    url = f"http://10.3.50.104:{ipport}"
    response = requests.get(url)
    json_data = response.json()
    ip=json_data['ip']
    create_time = datetime.datetime.now()

    return ip, create_time

def save_data_to_sql_server(ip, create_time):
    connection = pymysql.connect(server, user, password, database)
    cursor = connection.cursor()

    # 檢查是否已存在該 siteno 的資料
    select_sql = "SELECT COUNT(*) FROM solaredgesitenoipstatus WHERE siteno = %s"
    cursor.execute(select_sql, (siteno,))
    result = cursor.fetchone()
    count = result[0]

    if count > 0:
        # 刪除原有該 siteno 的資料
        delete_sql = "DELETE FROM solaredgesitenoipstatus WHERE siteno = %s"
        cursor.execute(delete_sql, (siteno,))

    # 儲存 local 伺服器的資料
    insert_sql = "INSERT INTO solaredgesitenoipstatus (siteno, ip, datetime, local
    | | | | VALUES (%s, %s, %s, %s)"
    params = (siteno, ip, create_time, 1)
    print(siteno,ip)
    cursor.execute(insert_sql, params)

    connection.commit()
    connection.close()

if __name__ == "__main__":
    ip, create_time = visit_server()
    save_data_to_sql_server(ip, create_time)
```

中。

圖 3-11、localcommunication.py 腳本檔主函數程式碼

8. 檔案八：Soladmod.bat 批次檔

用於執行上述多個 Python 腳本以及在特定時間範圍內執行，其主要功能如下：

- 執行 autofilerenew.py 和 localcommunication.py 兩個 Python 腳本，確認檔案保持在最新狀態及通訊正常。
- 檢查當前時間是否介於凌晨 4 點至晚上 8 點之間，若介於此時間段，則執行 solaredgemodbus.py 腳本，以儲存逆變器資訊。

```
@echo off
chcp 65001

:LOOP
echo %TIME%
set HOUR=%TIME:~0,2%
echo HOUR:%hour%

REM Check if current time is within the executable range (4:00 to 20:00)
call conda activate modbus
python C:\Users\N000184003\Desktop\solaredge\Solaredgeapplicationfile\autofilerenew.py
python C:\Users\N000184003\Desktop\solaredge\Solaredgeapplicationfile\localcommunication.py

if %hour% geq 4 if %hour% leq 20 (
    echo Starting script...
    REM Execute Python script
    @REM conda activate modbus
    @REM call conda activate modbus
    python C:\Users\N000184003\Desktop\solaredge\Solaredgeapplicationfile\solaredgemodbus.py
)

REM Pause for 3 minutes
timeout /t 180 /nobreak >NUL
@REM timeout /t 180 >NUL
echo %TIME%

REM Repeat execution until program is manually terminated
goto LOOP
```

圖 3-12、Soladmod.bat 批次檔程式碼

9. 檔案九：webinterface.py 腳本檔

用於將逆變器資訊介接至現有之綠電網頁，並將資料 (solaredgeinverterdata 資料表) 經換算後轉存至綠電網頁現有資料表 (SolarEnergyHistoricalData 資料表) 中，其資料表結構將於 3-3 小節說明，下圖 3-13 為 webinterface.py 主程式碼。

```

# 檢查記錄是否存於資料庫中
query = '''
SELECT * FROM SolarEnergyHistoricalData WHERE Roof = %(Roof)s AND DateTime = %(DateTime)s
'''
cursor.execute(query, newroof_data[f"fc{sitenos}"])
existing_record = cursor.fetchone()
if existing_record is not None:
    print('已有資料, 改寫')
    # 更新資料庫中的記錄
    query = '''
UPDATE SolarEnergyHistoricalData
SET DateTime = %(DateTime)s,
    Roof = %(Roof)s,
    RoofName = %(RoofName)s,
    Factory = %(Factory)s,
    [Case] = %(Case)s,
    DateParallel = %(DateParallel)s,
    InstalledCapacity = %(InstalledCapacity)s,
    EachUnitElectricityProduction = %(EachUnitElectricityProduction)s,
    ElectricityProduction = %(ElectricityProduction)s,
    ElectricityProductionThisMonth = %(ElectricityProductionThisMonth)s,
    ElectricityProductionThisYear = %(ElectricityProductionThisYear)s,
    SolarIrradiance = %(SolarIrradiance)s,
    PR = %(PR)s,
    CapacityFactorThisMonth = %(CapacityFactorThisMonth)s,
    CapacityFactorThisYear = %(CapacityFactorThisYear)s
WHERE Roof = %(Roof)s AND DateTime = %(DateTime)s
'''
else:
    print('無資料')
    # 插入新的記錄到資料庫中
    query = '''
INSERT INTO SolarEnergyHistoricalData ({fields})
VALUES ({values})
'''
    .format(
        fields=", ".join(newroof_data[f"fc{sitenos}"].keys()),
        values=", ".join("{} {}".format(k, newroof_data[f"fc{sitenos}"][k]) for k in newroof_data[f"fc{sitenos}"].keys())
    )

```

圖 3-13、webinterface.py 腳本檔主函數程式碼

3-5 綠電網頁介接

上述現場逆變器之資料存於，solaredgeinverterdata 資料表中，在經過 3-2 小節 webinterface.py 腳本進行資料轉換後，即可存於綠電網頁現有資料表 SolarEnergyHistoricalData 中，其資料表格式如下表 3-6 所示。

列名	DateTime	Roof	RoofName	Factory	DateParallel
資料型態	datetime	Nvarchar(50)	Nvarchar(50)	Nvarchar(20)	datetime
中英對照	資料創建時間	案場編號	案場名稱	現場工作站狀態	案場上線時間
列名	InstalledCapacity	EachUnit ElectricityProduction	ElectricityProduction	ElectricityProduction ThisMonth	ElectricityProduction ThisYear
資料型態	float	float	float	float	float
中英對照	裝置容量	單位發電量	日發電量	月發電量	年發電量
列名	SolarIrradiance	PR	CapacityFactor ThisMonth	CapacityFactor ThisYear	[Case]
資料型態	float	float	float	float	float
中英對照	日照	PR值	月容量因素	年容量因素	網頁識別編號

表 3-6、SolarEnergyHistoricalData 資料表結構

網頁顯示資訊主要有，上表中的 單位元發月容量因素
電量、日發電量、月發電量、年發電量、日照量、PR 值、月
容量因素、年容量因素等等，如下圖 3-14 所示。

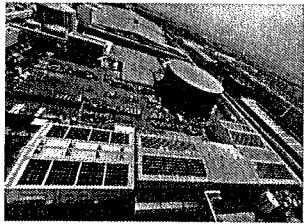


台化新港廠光電案場

首頁 / 新港廠 / 發電廠 / 儲能廠 / 彰化廠 / 公司外 / 水電

更新時間：2023-05-30 11:10:00

案場3 新港公用廠房屋頂



裝置容量(kWp) 347 併聯日期：2022/10/24	案場位置 <u>施工場所及保養廠房屋頂</u> 苛化廠房屋頂	即時功率(kW) 261.8
逆變器異常台數(即時) 0 (總共：4台)	維護聯繫窗口 工務部-新港公用廠 上次清洗：	今日等效發電量(kWh/kWp) 1.82
今日發電量(kWh) 633	本月累計發電量(kWh) 40,908	今年累計發電量(kWh) 183,505
發電效能比(今日) 84.3 %	本月/今年容量因數 16.4 % / 14.7 %	監控系統連結 盛齊綠能(監控連結)

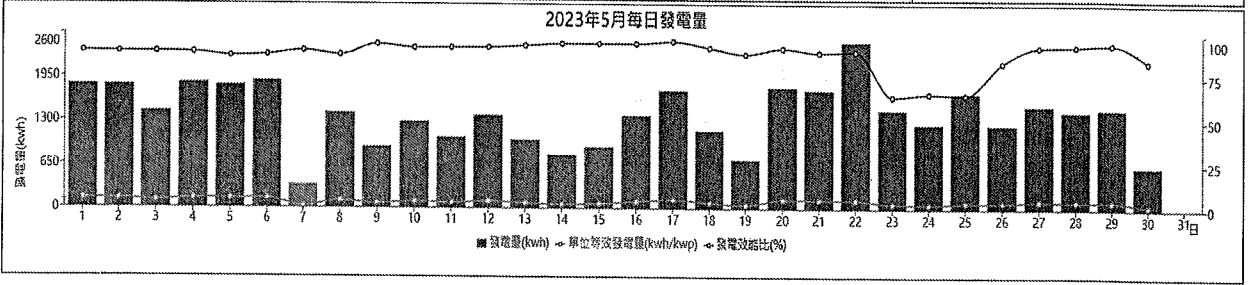


圖 3-14 現有綠電網頁示意圖

四、工作心得

轉眼間，來到自控處已經過九個月。期間承接新港公用廠 SK4 淨發電量 AI 模型修模」及「太陽能數據監視系統自行開發工程案」兩案，第一案主要在找出冷卻水塔中「風車」及「泵浦」的開台數最佳操作組合，進一步預測汽機的淨發電量，過程中有遇到資料不完整有缺漏，或是資料數值有極端值的狀況，所幸在與現場溝通的情況下，均有得到解答，藉由 AI 人員與現場 domain 不斷的溝通，最終找出最佳的模型，並提供相應的操作建議，從而更好地管理和控制淨發電量的變化。

而第二案「太陽能數據監視系統自行開發工程案」則是為配合本公司太陽光電案場陸續增加，各案場發電監視系統擬採自行開發，可節省各光電案場長達 20 年監控平台租賃費用，本案執行太陽能案廠自行開發工程之軟體設計，但也包含了泰山機房資料庫及相關軟體的請購部分，由於是第一次請購發現流程有許多不清楚的地方，也很感謝前輩們耐心的教導，讓我能順利請購。

目前總共開發了 7 隻腳本程式，但由於現在程式開發都在辦公室電腦模擬運行，因此待資料庫電腦建置完成後還需再更進一步上線測試，另外，諸如伺服器設定、資料庫設定

及路由器設定等等都還需要在此案完結後續持續進行。

而目前遇到最棘手的問題是，現場工作站電腦使用的是浮動 IP 4G 網路，若需要進行軟體更新或維護則無法使用遠端連線的方式進行，但現場的盤體均放置於露天開放式的情況下，若遇下雨或天候不佳等因素，人為現場維護是難度極高的，因此在與陳重昇高專討論後，擬定需要購買具有轉址功能的 4G 路由器，透過路由器管理介面，創建一個新的靜態 IP 位址分配規則，指定工作站電腦的 MAC 地址和要分配的固定 IP 地址，這樣即可透過分配的固定 IP 地址遠端進行維護，配合上文所提到的 localcommunication.py 腳本隨時更新現場浮動 IP 的位置，即可透過在遠端連線介面輸入{浮動 IP}:{PORT}的方法遠端連線至工作站電腦，同時讓工作站電腦保有連線至公共網路的功能，這樣才能向遠端資料庫儲存資料。

這幾個月體會到自控處是軟硬體分不開的，諸如串列通訊、防火牆設定、路由器設定、等等都需要一定的硬體觀念，因此還需要向自控處的各位前輩多多學習，精進自身能力。

在此段時間也上了人工智慧學校的課程，專題內容為營建處的鋼構重量最佳化，我負責的部分為資料處理、AI 建模訓練及後續的基因演算法開發等部分，由於此案尚在開發中，將於下篇報告中詳細討論。

五、附錄

位址	基址	大小	名稱	類型	單位	說明
40069	40070	1	C_SunSpec_DID	uint16	N/A	SunSpec 設備識別碼
40070	40071	1	C_SunSpec_Length	uint16	N/A	模型區塊長度
40071	40072	1	I_AC_Current	uint16	安培 (A)	AC 總電流值
40072	40073	1	I_AC_CurrentA	uint16	安培 (A)	AC 的 A 相電流值
40073	40074	1	I_AC_CurrentB	uint16	安培 (A)	AC 的 B 相電流值
40074	40075	1	I_AC_CurrentC	uint16	安培 (A)	AC 的 C 相電流值
40075	40076	1	I_AC_Current_SF	int16	N/A	AC 電流比例因數
40076	40077 (1)	1	I_AC_VoltageAB	uint16	伏特 (V)	AC 的 AB 相電壓值
40077	40078 (1)	1	I_AC_VoltageBC	uint16	伏特 (V)	AC 的 BC 相電壓值
40078	40079 (1)	1	I_AC_VoltageCA	uint16	伏特 (V)	AC 的 CA 相電壓值
40079	40080 (1)	1	I_AC_VoltageAN1	uint16	伏特 (V)	AC 的 A 相對 N 電壓值
40080	40081	1	I_AC_VoltageBN1	uint16	伏特 (V)	AC 的 B 相對 N 電壓值
40081	40082	2	I_AC_VoltageCN1	uint16	伏特	AC 的 C 相對 N 電壓值
40082	40083	1	I_AC_Voltage_SF	int16		AC 電壓比例因數
40083	40084	1	I_AC_Power	int16	瓦特	AC 功率值
40084	40085	1	I_AC_Power_SF	int16		AC 功率比例因數
40085	40086	1	I_AC_Frequency	uint16	赫茲	AC 頻率值
40086	40087	1	I_AC_Frequency_SF	int16		比例因數
40087	40088	1	I_AC_VA	int16	VA	視在功率
40088	40089	1	I_AC_VA_SF	int16		比例因數
40089	40090	1	I_AC_VAR	int16	VAR	虛功率
40090	40091	1	I_AC_VAR_SF	int16		比例因數
40091	40092	1	I_AC_PF	int16	%	功率因數
40092	40093	1	I_AC_PF_SF	int16		比例因數
40093	40094	2	I_AC_Energy_WH	acc32	瓦時	AC 總累積的發電量
40095	40096	1	I_AC_Energy_WH_SF	uint16		比例因數
40096	40097	1	I_DC_Current	uint16	安培	DC 電流值
40097	40098	1	I_DC_Current_SF	int16		比例因數
40098	40099	1	I_DC_Voltage	uint16	伏特	DC 電壓值
40099	40100	1	I_DC_Voltage_SF	int16	比例因數	直流 (DC) 電壓比例因數
40100	40101	1	I_DC_Power	int16	瓦特	直流 (DC) 功率值
40101	40102	1	I_DC_Power_SF	int16	比例因數	直流 (DC) 功率比例因數
40103	40104	1	I_Temp_Sink	int16	攝氏度	散熱片溫度
40106	40107	1	I_Temp_SF	int16	比例因數	溫度比例因數
40107	40108	1	I_Status	uint16	-	操作狀態
40108	40109	1	I_Status_Vendor	uint16	-	廠商定義的操作狀態及錯誤代碼。 錯誤說明、意義及故障排除請參閱 SolarEdge 安裝指南。

表 5-1 solaredgeinverterdata 對照表