

年度科技專案  
(韌性生產系統技術開發計畫(1/4))

## 產線自主補料決策與控制技術

期中報告

(產線自主補料決策與控制技術)

## 期中報告

計畫主持人：賴奎魁 教授兼副校長

研究機構：朝陽科技大學企業管理系

執行期間：自111年03月01日至111年11月30日止

中華民國 111 年 6 月 30 日

## 計畫摘要

本合作研究計畫目標在於利用企業客戶的歷史訂單需求，針對ERP銷售系統當中物料進料與補料的需求，提出一套深度學習的推薦機制，預測未來幾週客戶的訂單量，並將預測值轉成生產線的生產排程及備料需求。若能精準預測客戶之下單量，將可降低庫存壓力、準時交貨及提供多目標補料策略之備料需求量，對於降低企業庫存成本將有非常大的助益。

(關鍵詞)：深度學習、大數據、智慧製造、需求預測

## 目錄

	頁次
計 畫 摘 要 .....	i
目 錄 .....	ii
表 目 錄 .....	iii
圖 目 錄 .....	iv
一 前 言 .....	1
(一) 背景 .....	1
(二) 相關文獻 .....	1
二 研 究 目 的 .....	1
三 研 究 方 法 .....	2
(一) 專利技術手段 .....	2
(二) 專利文獻之檢索 .....	2
(三) 資料前處理 .....	4
(四) 模型訓練結果 .....	7
四 結 論 .....	9
五 參 考 文 獻 .....	9

## 表目錄

表 3-1 專利檢索結果.....	3
表 3-2 資料集之客戶下單數與庫存量狀況.....	4
表 3-3 LSTM 模型-訓練結果.....	8

## 圖目錄

圖 3-1 專利技術手段.....	2
圖 3-2 原始資料集.....	4
圖 3-3 建立 Dataframe 資料.....	4
圖 3-4 調整資料起始時間.....	5
圖 3-5 資料缺失值處理.....	5
圖 3-6 資料中日期的智慧轉換.....	5
圖 3-7 A 品項-在庫量之時間變化 .....	6
圖 3-8 資料維度的轉換.....	6
圖 3-9 數據歸一化處理.....	7
圖 3-10 訓練過程.....	7

## 一、前言

### (一)背景

在工業4.0時代的推動下，各國皆朝向智慧化的營運方式發展，雖然在數位轉型的過程中，可能會有時間與金錢的考量而無法落實，但若能提早進行評估並做出適時的改變，相信可以在之後產生更大的影響與競爭力。而現今產業中，多是以過去的經驗來解決問題、做出決策，但這樣較難被系統化，同時需花費許多的時間，甚至多數的決策都是以感覺為主，而沒有實際數據能支持所做出的決策。因此若能將工作的經驗量化，保存並作為日後分析的依據，就能針對不足的部分進行改善，而達到降低成本、提升生產線效率之目的。

在滿足客戶的前提條件下，於生產工廠的生產效率、庫存管理上，將受到很大的影響，其中針對長期大量下單之客戶，最怕遇到的就是備料不足，生產成品無法準時交貨，造成下單客戶之損失，導致客戶減少訂單或另尋可替代之供應商。因此為了滿足客戶之需求，供應商透過計畫性生產，能夠事先預測需求量，提前備料並超前部屬，於接到客戶訂單前就已完成生產或至少大部分製程，以此來讓產品的交貨期能夠符合客戶的要求。其中關鍵料庫存管理是生產線的源頭，若原料庫存過多會造成資金積累，若庫存不足則會使生產停機待料，因此精準提前預測關鍵料備料，是廠商需面對的重要課題。

### (二)相關文獻

在過去十年中，機器學習模型的發展，使複雜回歸模型的發展成為可能，出於這個原因，在各行業中已經進行了研究並使用機器學習來設計預測模型。在智慧製造方面，有應用在能源效率[1-5]，設備狀態檢測[6-11]，以及產品缺陷檢測[12-14]等。隨著公司的不斷發展，對預測產品準確性的需求只會不斷增加，因此需要有更適合的預測方法來解決這些問題。在[15][16]中，作者將機器學習應用於電力負載的預測，並證實它有好的運行效果；而Toubeau等人[17]則應用在多元電力市場之調度；其他還有機器學習應用於太陽能發電，並在複雜的自然現象中找到重要的預測結果[18-20]等等。可以看出，目前有許多透過機器學習來預測其他項目之需求，其中也包括在庫存管理中的應用[21-23]，可以幫助公司有效率地降低成本。

## 二、研究目的

以往面對備料的需求，是由生管人員人工預測的方式，決定訂單的送料需求量，但若預估值太大會造成成品庫存過多，導致倉儲成本增加而影響毛利。而預估值太小，則會造成庫存不足，延長交貨期，影響後續客戶之下單量。因此要如何降低庫存不足之週數及減少庫存最大量為本計畫研究之目的，以達到更精準的方式評估所需要之每週生產量。因此我們計畫以一套深度學習的模型，針對 ERP 銷售系統當中物料進料與補料的需求，進行預測未來幾週客戶的訂單量，以客戶需求面來說，假設客戶要求的交期有 2 週，但實際製程須 6 週，就必須提前備料及產品庫存，以因應生產能趕上交期。

### 三、研究方法

資料面會先透過ERP取得計畫性生產產品之客戶下單歷史資料，進行數據前處理並建立比較之基準。現階段是先使用LSTM模型來預測品項A之在庫量，做一個初步的嘗試，了解資料前處理後的可行性。以下為較詳細之實作階段性過程。未來會再嘗試使用不同的方法做實驗並比較優缺點與差異，從中找出最合適的架構，以及哪些參數會影響預估客戶下單量的準確性。

#### (一)專利技術手段

時間序列預測問題之困難點在於它的高度不穩定和非線性的特性，使得我們依靠單純的線性模型(SVM、ARIMA)或是基礎的機器學習模型(LSTM、CNN\_LSTM)通常難以捕捉到資料在時間上的關聯性和變化，導致預測失誤，因此本計畫所提出之專利技術，希望能找出Target data和Non-target data的細部特徵，以及找出它們在時間上的關聯性和變化，達到細化萃取特徵的效果，提升模型預測的準確性。其中Target data為過去一段時間的產品銷售量，而Non-target data除了過去的產品銷售量以外，其他還可包含季節、淡旺季、目前需求量、未來3個月需求量、3個月後之訂貨週期、當前庫存量、每件商品採購成本、庫存損失等。

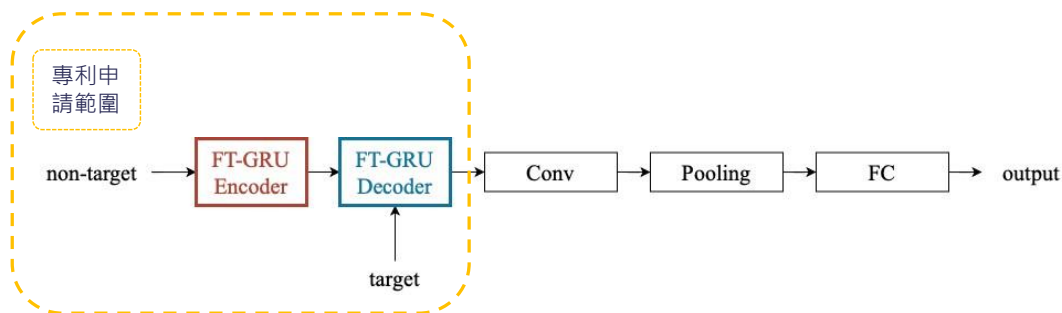


圖3-1 專利技術手段

這裡舉風力發電產量預測問題為例，環境的溫度、濕度、可見度、氣壓、風向及風速等作為Non-target data，發電產量作為Target-data。Non-target data會進入FT-GRU Encoder&Decoder去萃取其特徵，包含特徵之間的關聯以及時序間的關聯性，例如：溫度和濕度之間可能存在某種關聯、風速和可見度之間可能有關聯以及連續的時間內溫度之前會有關聯等。Target Data會經過多層的GRU去抓取長短時間內的特徵變化，例如：前一天的風力發電產量可能和當天的風力發電產量有關、去年夏天可能和今年夏天有關等。最後將Non-target data及Target Data的結果合併進入CNN去抓取重要的特徵，以輸出預測得到最終發電產量結果。

#### (二)專利文獻之檢索

在庫存最佳化的自主補料技術中分為三大部分：物料管理、線邊庫存管理以及庫存管理。

物料管理是於進行入料/入庫動作之前，視實際需求情況，進行物料點料動作。在這方面相關的專利如TW202046195A[1]適用於發放存放於倉庫所需材料，TW202008243A[2]利用機器手臂進行物料出/入庫。在清單處理的方面，CN114239537A[3]提供一種物料清單快速比對



的方法。CN114118921A[4]是基於ERP系統實現物料調撥一體化，CN113379353A[5]則可以提高物料存取效率。

線邊庫存管理是生產線邊上的暫存庫管理。TW202008242A[6]不使用人工作業出庫到工單料線邊會上備料區，CN108898278A[7]打通裝配和測試系統間的信息斷層，提高整個生產線的生產效率。US20220067622A1[8]預測產品吞吐量，並向實體介面發出警報和/或推薦操作。

庫存管理又分為庫存物料作、庫存儲位以及庫存盤點整理。TW201918943A[9]利用行動化即時IoT掃描裝置以及條碼標籤達到庫存管理。TW202141371A[10]增加空間使用率的庫存管理方法。CN113487177A[11]藉由倉庫的物料使用壽命和量整個倉庫的物料質量和管理質量。US20220058577A1[12]追蹤和核算部分使用的庫存，並基於調查結果產生供需預測或庫存優化建議。US20210398064A1[13]構建庫存項目分佈的三維模型與庫存物品分佈的數據庫比較差異。

表3-1 專利檢索結果

	公開號	專利名稱
1	TW202046195A	物料控管方法及物料控管系統
2	TW202008243A	物料管理系統及其方法
3	CN114239537A	服務器物料清單快速比對方法、系統、終端及存儲介質
4	CN114118921A	一種基於ERP系統實現物料調撥一體化的系統及方法
5	CN113379353A	一種智能物料管理系統
6	TW202008242A	倉儲管理系統及其方法
7	CN108898278A	一種生產線數據自動採集系統及方法
8	US20220067622A1	SYSTEMS AND METHODS FOR AUTOMATING PRODUCTION INTELLIGENCE ACROSS VALUE STREAMS USING INTERCONNECTED MACHINE-LEARNING MODELS
9	TW201918943A	庫存收貨管理系統
10	TW202141371A	用於庫存管理之電子裝置及其操作方法
11	CN113487177A	一种物料管理系统及管理方法
12	US20220058577A1	To inventory management systems and related methods
13	US20210398064A1	SYSTEM AND METHOD FOR DYNAMIC INVENTORY MANAGEMENT

### (三)資料前處理

資料集中每個品項都有各自的客戶下單數與出入庫、在庫量，目前這是「客戶下單量與庫存量統計表」之資料，在經過與賴先生討論後得知，在實際的ERP銷售系統中會生成的其實是「歷史出入庫資料」，故後續的研究中，會改採用歷史出入庫資料作為我們的訓練資料。

表3-2 資料集之客戶下單數與庫存量狀況

名稱	資料更新週期	單位
客戶下單數	一週一筆 (非每周都有下單)	週數 (2020/2/9~2021底)
庫存量狀況 (包含入庫、出庫、在庫量)	2020：7天一筆 2021：2天一筆/5天一筆	日期 (2020/3/26~2021/12/30)

客戶下單量與庫存量統計表之原始資料如圖3-2，其中上一次的在庫量加上入庫量減去出庫量，就會得到當前的在庫量。

入出/在	年分																														2020年						
	日期	1-2	1-9	1-16	1-23	1-30	2-6	2-13	2-20	2-27	3-5	3-12	3-19	3-26	4-2	4-9	4-16	4-23	4-30	5-7	5-14	5-21	5-28	6-4	6-11	6-18	6-25	6-30	7-7								
入庫量														1,170	0	1,205	1,000	1,000	1,000	1,000	1,000	1,250	1,250	750	750	750	750	750	0								
出庫量														1,170	0	1,205	955	610	278	610	1,862	1,842	0	375	470	0	0	0	0								
在庫量	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	435	1,157	1,547	685	93	1,343	1,718	1,998	2,748	3,498	4,248	4,248								
入庫量														710	0	318	850	850	940	850	850	850	1,000	700	700	700	700	700									
出庫量														710	0	318	715	985	940	170	1,372	0	1,077	375	350	0	0	0									
在庫量	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	135	0	0	680	158	1,008	931	1,256	1,606	2,306	3,006	3,706	3,706								
入庫量														1,905	635	2,430	2,760	2,635	2,635	1,800	1,800	3,600	3,500	2,100	2,100	2,100	2,100	2,100									
出庫量														1,905	635	2,340	2,775	1,270	2,748	1,600	3,005	3,907	2,880	1,455	1,635	255	0	0									
在庫量	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	75	1,440	1,327	1,527	322	15	635	1,280	1,745	3,590	5,690	7,790	7,790								
入庫量														1,590	530	1,750	1,200	2,150	2,860	1,600	1,600	4,130	3,500	2,100	2,100	2,100	2,100	1,500									
出庫量														1,590	530	1,680	1,060	1,940	2,465	2,215	1,798	4,070	3,532	1,859	1,940	2,110	0	530	0								
在庫量	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	210	420	815	200	2	62	30	271	431	421	2,521	4,091	5,591								

圖3-2 原始資料集

圖3-3首先是使用Python程式語言中，用於數據操縱和分析的Pandas軟體庫來將Excel檔案的資料作抓取，並保留我們想要的部分，並透過Pandas軟體庫提供之Dataframe，建立Dataframe物件儲存雙維度資料。

日期	2020-1-2	2020-1-9	2020-1-16	2020-1-23	2020-1-30	2020-2-6	\
在庫量	0	0	0	0	0	0	
2020-2-13	2020-2-20	2020-2-27	...	2021-11-30	2021-12-2	2021-12-7	\
	0	0	0	...	3422	3922	3922
2021-12-9	2021-12-14	2021-12-16	2021-12-21	2021-12-23	2021-12-28	\	
	4482	4482	5982	5982	6282	6224	
2021-12-30							
	7124						

圖3-3 建立Dataframe資料

接著將前面0的部分去除，從資料有更新的地方開始作為我們後續訓練模型的起始時間如圖3-4。

2020-4-16	2020-4-23	2020-4-30	2020-5-7	2020-5-14	2020-5-21	2020-5-28	\
45	435	1157	1547	685	93	1343	
2020-6-4	2020-6-11	2020-6-18	...	2021-11-30	2021-12-2	2021-12-7	\
1718	1998	2748	...	3422	3922	3922	
2021-12-9	2021-12-14	2021-12-16	2021-12-21	2021-12-23	2021-12-28	\	
4482	4482	5982	5982	6282	6224		
2021-12-30							
7124							

圖3-4 調整資料起始時間

再來檢查是否有缺失值並做資料的處理，防止後續訓練時資料有問題的情況發生，圖3-5為找到其中一筆缺失值並作處理之例子。

```

缺失值總數: 1
      2021-4-15  2021-4-22  2021-4-27  2021-4-29
0      2563      3118      NaN      4618
      2021-4-15  2021-4-22  2021-4-27  2021-4-29
0      False      False      True      False
刪除後缺失值總數: 0
      2021-4-15  2021-4-22  2021-4-29  2021-5-4
0      False      False      False      False

```

圖3-5 資料缺失值處理

再來將日期做智慧轉換如圖3-6，將Dataframe列轉換為Python日期時間的方法，使其自動判斷為日期格式，以利後續在可視化預測結果時，顯示的圖表呈現正確的時間軸。

```

數據=
  2020-4-16      45
2020-4-23      435
2020-4-30      1157
2020-5-7       1547
2020-5-14      685
dtype: int64
日期=
Index(['2020-4-16', '2020-4-23', '2020-4-30', '2020-5-7', '2020-5-14',
       '2020-5-21', '2020-5-28', '2020-6-4', '2020-6-11', '2020-6-18',
       ...,
       '2021-11-30', '2021-12-2', '2021-12-7', '2021-12-9', '2021-12-14',
       '2021-12-16', '2021-12-21', '2021-12-23', '2021-12-28', '2021-12-30'],
      dtype='object', length=139)
日期(智慧轉換)= DatetimeIndex(['2020-04-16', '2020-04-23', '2020-04-30', '2020-05-07',
                                '2020-05-14', '2020-05-21', '2020-05-28', '2020-06-04',
                                '2020-06-11', '2020-06-18',
                                ...,
                                '2021-11-30', '2021-12-02', '2021-12-07', '2021-12-09',
                                '2021-12-14', '2021-12-16', '2021-12-21', '2021-12-23',
                                '2021-12-28', '2021-12-30'],
                                dtype='datetime64[ns]', length=139, freq=None)

```

圖3-6 資料中日期的智慧轉換

使用Matplotlib資料視覺化工具，方便我們理解數據的變化過程，透過一堆數字我們無法直接看出結果，但透過圖形、圖表資料的呈現，就可以讓我們明確的知道資料的走向和結果。

圖3-7為A品項庫存量之時間變化，載入的資料集為2020年4月到2021年12月，總共有139筆資料。

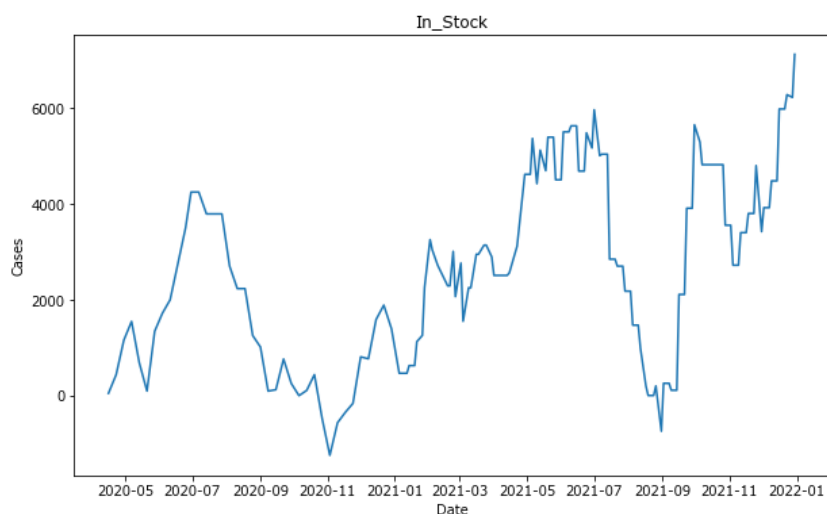


圖3-7 A品項-在庫量之時間變化

接著要做資料餵入模型的準備，所以需要將資料轉換成陣列的型式，並且一列只能有一個值，使輸入的資料型態能順利餵入LSTM模型。圖3-8是使用Numpy函式庫將Dataframe格式之資料轉成二維陣列的內容值，再接續後面的運算。

```
# DataFrame(467行) -> values內容值(467列,1行)
train_data = np.expand_dims(train_data, axis=1) # (467列,1列存放1個數值)
val_data = np.expand_dims(val_data, axis=1) # axis=1: (467) -> (467,1)
```

圖3-8 資料維度的轉換

很多演算法對數據範圍非常的敏感。因此為了要讓模型訓練的更強大，通常的做法是對特徵進行調節，使得數據更適合這些演算法。這裡將資料數據做歸一化，使用的套件是Scikit-learn中的MinMaxScaler。

其中使用的MinMaxScaler方法會把數據壓縮到0與1之間，好處是可以消除base unit對最終結果的影響，使不同變數也具有可比性。其中要注意的是，後續在做驗證時，訓練集做的transform必須與驗證集的transform一致，才不會導致有兩種歸一化的標準。圖3-9為資料前5筆數據轉換之前後差異，到這裡即完成資料的前處理，準備進入模型訓練。

```

轉換前:
[[ 45]
 [ 435]
 [1157]
 [1547]
 [ 685]]
train_data shape= (79, 1)

轉換後:
[[0.19149882]
 [0.24925948]
 [0.35619076]
 [0.41395142]
 [0.28628555]]

```

圖3-9 數據歸一化處理

#### (四)模型訓練結果

將上述處理好之資料使用LSTM模型做訓練，訓練過程如圖3-10，總共跑了500回合，並得到如表3-2之實驗結果。

```

查看餵入的數據shape是否正確:
x = torch.Size([51, 28, 1]) (batch_size, time_step, input_size)

Epoch 0  train loss: 17.2390  val loss: 21.5095  最佳評估指標 MSE= 0.6722 (best_epoch=0)
  best_epoch 0  val loss: 21.5095
  best_epoch 1  val loss: 14.8579
  best_epoch 2  val loss: 7.5404
  best_epoch 3  val loss: 1.8468
  best_epoch 16  val loss: 1.8158

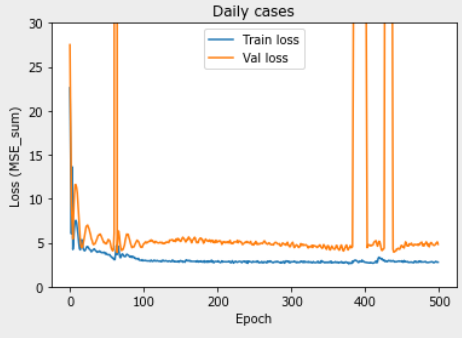
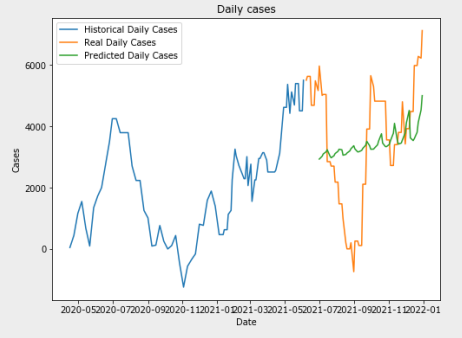
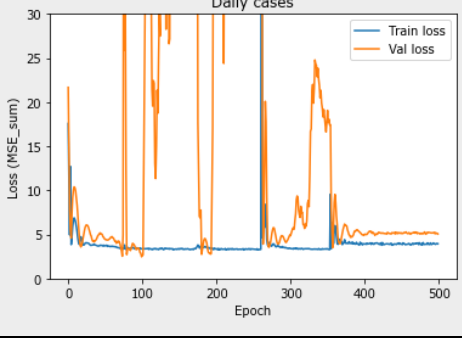
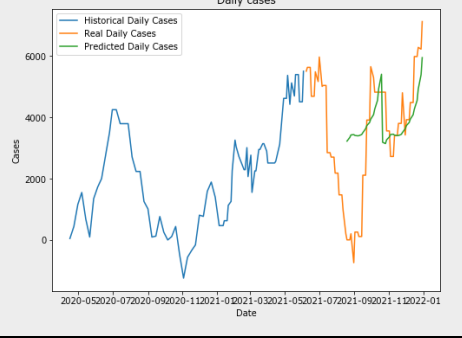
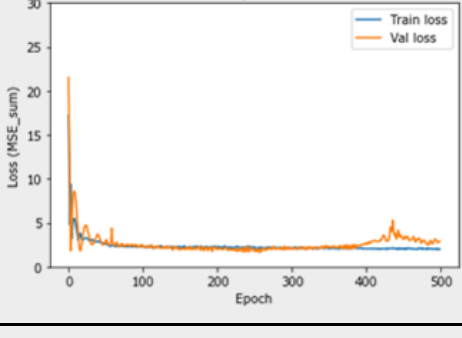
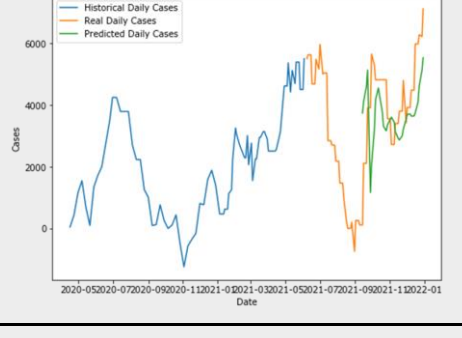
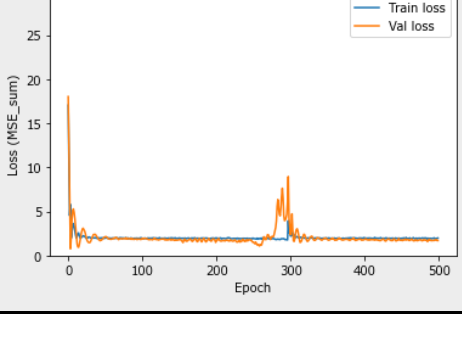
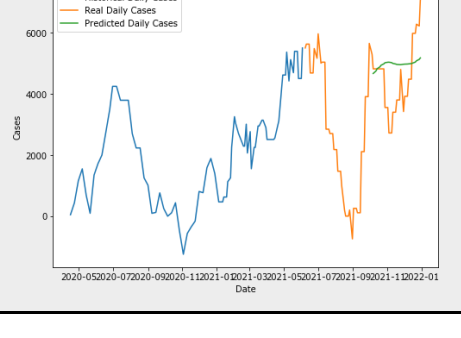
Epoch 20  train loss: 3.1091  val loss: 3.8941  最佳評估指標 MSE= 0.0567 (best_epoch=16)
Epoch 40  train loss: 2.8683  val loss: 3.5548  最佳評估指標 MSE= 0.0567 (best_epoch=16)
Epoch 60  train loss: 2.4363  val loss: 2.7188  最佳評估指標 MSE= 0.0567 (best_epoch=16)
Epoch 80  train loss: 2.3093  val loss: 2.5263  最佳評估指標 MSE= 0.0567 (best_epoch=16)
Epoch 100  train loss: 2.2554  val loss: 2.2420  最佳評估指標 MSE= 0.0567 (best_epoch=16)
Epoch 120  train loss: 2.2519  val loss: 2.1987  最佳評估指標 MSE= 0.0567 (best_epoch=16)
Epoch 140  train loss: 2.1873  val loss: 1.9092  最佳評估指標 MSE= 0.0567 (best_epoch=16)
Epoch 160  train loss: 2.1964  val loss: 2.0952  最佳評估指標 MSE= 0.0567 (best_epoch=16)
Epoch 180  train loss: 2.1939  val loss: 2.2985  最佳評估指標 MSE= 0.0567 (best_epoch=16)
  best_epoch 193  val loss: 1.7316
Epoch 200  train loss: 2.1045  val loss: 2.0126  最佳評估指標 MSE= 0.0541 (best_epoch=193)

```

圖3-10 訓練過程

time step意指模型輸入不同筆數的資料，預測下一筆之結果。可以看到預測結果的部分，前半段資料是拿來訓練的資料(藍色線)，後半段資料是拿來做驗證與測試的資料，其中橘色線為Ground Truth，綠色線為預測成果。

表3-3 LSTM模型-訓練結果

time step	loss	預測結果
7		
21		
28		
35		

#### 四、結論

在不同time step(序列長度)下，得到的預測結果有明顯的差異。如果樣本資料夠多，在不超過臨界值的條件下，time step的上升對於預測的準確度是有幫助，然而在樣本資料有限的情況下，就可能需要自行做數據的補強，後續會再針對這塊進一步研究，以及繼續嘗試不同深度學習演算法架構之實作。

#### 五、參考文獻

- [1] L. Fang, K. Ma, R. Li, Z. Wang and H. Shi, "A Statistical Approach to Estimate Imbalance-Induced Energy Losses for Data-Scarce Low Voltage Networks," IEEE Transactions on Power Systems, vol.34, no.4 pp.2825-2835, 2019.
- [2] L. Zhu, C. Lu, Z. Y. Dong and C. Hong, "Imbalance Learning Machine-Based Power System Short-Term Voltage Stability Assessment," IEEE Transactions on Industrial Informatics, vol.13, no.5, pp.2533-2543, 2017.
- [3] C. Tang, S. Wang, C. Zhou, X. Zheng, H. Li and X. Shi, "An Energy-Image Based Multi-Unit Power Load Forecasting System," IEEE International Conference on Industrial Internet (ICII), pp.69-78, 2018.
- [4] B. Bhattacharya and A. Sinha, "Intelligent Fault Analysis in Electrical Power Grids," IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), pp.985-990, 2017.
- [5] X. Wang, T. Zhao, H. Liu and R. He, "Power Consumption Predicting and Anomaly Detection Based on Long Short-Term Memory Neural Network," IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp.487-491, 2019.
- [6] N. Helwig, E. Pignanelli and A. Schütze, "Condition monitoring of a complex hydraulic system using multivariate statistics," IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp.210-215, 2015.
- [7] J. -R. Jiang and C. -K. Kuo, "Enhancing Convolutional Neural Network Deep Learning for Remaining Useful Life Estimation in Smart Factory Applications," International Conference on Information, Communication and Engineering (ICICE), pp.120-123, 2017.
- [8] Z. Chen, Y. Liu and S. Liu, "Mechanical state prediction based on LSTM neural network," 2017 36th Chinese Control Conference (CCC), pp.3876-3881, 2017.
- [9] W. Zhang et al., "LSTM-Based Analysis of Industrial IoT Equipment," IEEE Access, vol. 6, pp.23551-23560, 2018.
- [10] Q. Han, P. Liu, H. Zhang and Z. Cai, "A Wireless Sensor Network for Monitoring Environmental Quality in the Manufacturing Industry," IEEE Access, vol.7, pp.78108-78119, 2019.
- [11] Ji-Hyeong Han and Su-Young Chi, "Consideration of manufacturing data to apply machine learning methods for predictive manufacturing," 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), pp.109-113, 2016.

- [12] T. Lee, K. B. Lee and C. O. Kim, "Performance of Machine Learning Algorithms for Class-Imbalanced Process Fault Detection Problems," *IEEE Transactions on Semiconductor Manufacturing*, vol.29, no.4, pp.436-445, 2016.
- [13] B. M. Haddad, S. Yang, L. J. Karam, J. Ye, N. S. Patel and M. W. Braun, "Multifeature, Sparse-Based Approach for Defects Detection and Classification in Semiconductor Units," *IEEE Transactions on Automation Science and Engineering*, vol.15, no.1, pp.145-159, 2018.
- [14] Y. Luo, J. Qiu and C. Shi, "Fault Detection of Permanent Magnet Synchronous Motor Based on Deep Learning Method," *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, pp.699-703, 2018.
- [15] H. Shi, M. Xu and R. Li, "Deep Learning for Household Load Forecasting—A Novel Pooling Deep RNN," *IEEE Transactions on Smart Grid*, vol.9, no.5, pp.5271-5280, 2018.
- [16] Gan, Dahua, et al. "Enhancing short-term probabilistic residential load forecasting with quantile long-short-term memory." *The Journal of Engineering* 2017, 14, 2622-2627.
- [17] J. Toubeau, J. Bottieau, F. Vallée and Z. De Grève, "Deep Learning-Based Multivariate Probabilistic Forecasting for Short-Term Scheduling in Power Markets," *IEEE Transactions on Power Systems*, vol.34, no.2, pp.1203-1215, 2019.
- [18] N. Tang, S. Mao, Y. Wang and R. M. Nelms, "Solar Power Generation Forecasting With a LASSO-Based Approach," *IEEE Internet of Things Journal*, vol.5, no.2, pp.1090-1099, 2018.
- [19] J. Wu and C. K. Chan, "The Prediction of Monthly Average Solar Radiation with TDNN and ARIMA," *2012 11th International Conference on Machine Learning and Applications*, pp.469-474, 2012.
- [20] Hejase, H.A.; Assi, A.H. Time-series regression model for prediction of mean daily global solar radiation in Al-Ain, UAE. *ISRN Renew. Energy* 2012, 412471.
- [21] Alan, Yasin, George P. Gao, and Vishal Gaur. "Does inventory productivity predict future stock returns? A retailing industry perspective." *Management Science* 60.10, 2416-2434, 2014.
- [22] Wang, Chien-Chih, Chun-Hua Chien, and Amy J.C. Trappey. 2021. "On the Application of ARIMA and LSTM to Predict Order Demand Based on Short Lead Time and On-Time Delivery Requirements" *Processes* 9, no.7, 11571, 2021.
- [23] N. Xue, I. Triguero, G. P. Figueredo and D. Landa-Silva, "Evolving Deep CNN-LSTMs for Inventory Time Series Prediction," *IEEE Congress on Evolutionary Computation (CEC)*, pp.1517-1524, 2019.