

比特幣價格預測

On the Prediction of Bitcoin Prices

系所班別：統計所碩二

姓名學號：李承志 (109354005)

吳承臻 (109354017)

劉得心 (109354020)

陳麒仲 (109354022)

報告日期：2022/1/12

摘要

近年來，隨著比特幣價格大漲，比特幣也逐漸成為大家其中一種投資工具，而預測比特幣價格也成為大家的工作之一。在此期末專案中，我們結合時間序列以及 **LSTM** 進行比特幣價格的預測，發現預測結果在未來的前三天半內有良好的效果，然而隨著時間變長在後三天半有偏離真實價格的情形。另外，我們使用 **ICA**、**Cmeans** 以及類神經網路 (**FNN**)進行比特幣價格趨勢的預測，達到結合迴歸與分類問題在比特幣價格方面的應用。

章節目錄

1. 導論	1
2. 流程圖	2
3. 迴歸分析	3
3.1. 時間序列模型.....	3
3.2. 自我相關(ACF)、偏自我相關(PACF)與 DF 檢定.....	3
3.3. 長短期記憶(Long Short-Term Memory; LSTM).....	4
4. 走勢分群	5
4.1. 獨立成分分析(Independent Component Analysis; ICA).....	5
4.2. Fuzzy Cmeans 分群法	6
4.3. 前回饋類神經網路(Feedforward Neural Network; FNN).....	8
5. 比特幣預測的實際應用	8
5.1. 比特幣價格預測.....	8
5.2. 比特幣價格趨勢預測.....	13
6. 結論與建議	15

圖目錄

圖 1 單一個 LSTM cell 運算流程.....	4
圖 2 many-to-many LSTM 示意圖.....	5
圖 3 Cmeans 分群流程圖.....	7
圖 4 FNN 神經網路示意圖.....	8
圖 5 比特幣價格在取對數後的趨勢、季節以及短期循環.....	9
圖 6 log(bitcoin)的剩餘趨勢的 ACF 與 PACF 圖.....	9
圖 7 模型變數 X 與比特幣價格趨勢圖（以 NVIDIA、TSLA、乙太幣為例）	10
圖 8 LSTM 模型架構（units = 500, 300）.....	11
圖 9 Model 3 資料變數 X 與反應變數 Y 的對應關係圖.....	11
圖 10 比特幣預測結果範例.....	12
圖 11 最終比特幣預測結果示意圖.....	12
圖 12 ICA 分解後的 7 個獨立訊號波段.....	13
圖 13 比特幣非常樂觀走勢.....	14
圖 14 比特幣非常悲觀走勢.....	14
圖 15 類神經網路架構.....	15
圖 16 比特幣價格真實預測（第一、二次）.....	16
圖 17 比特幣價格真實預測（第三、四次）.....	16
圖 18 比特幣價格真實預測（第五、六次）.....	17
圖 19 比特幣價格真實預測（第七、八次）.....	17

1. 導論

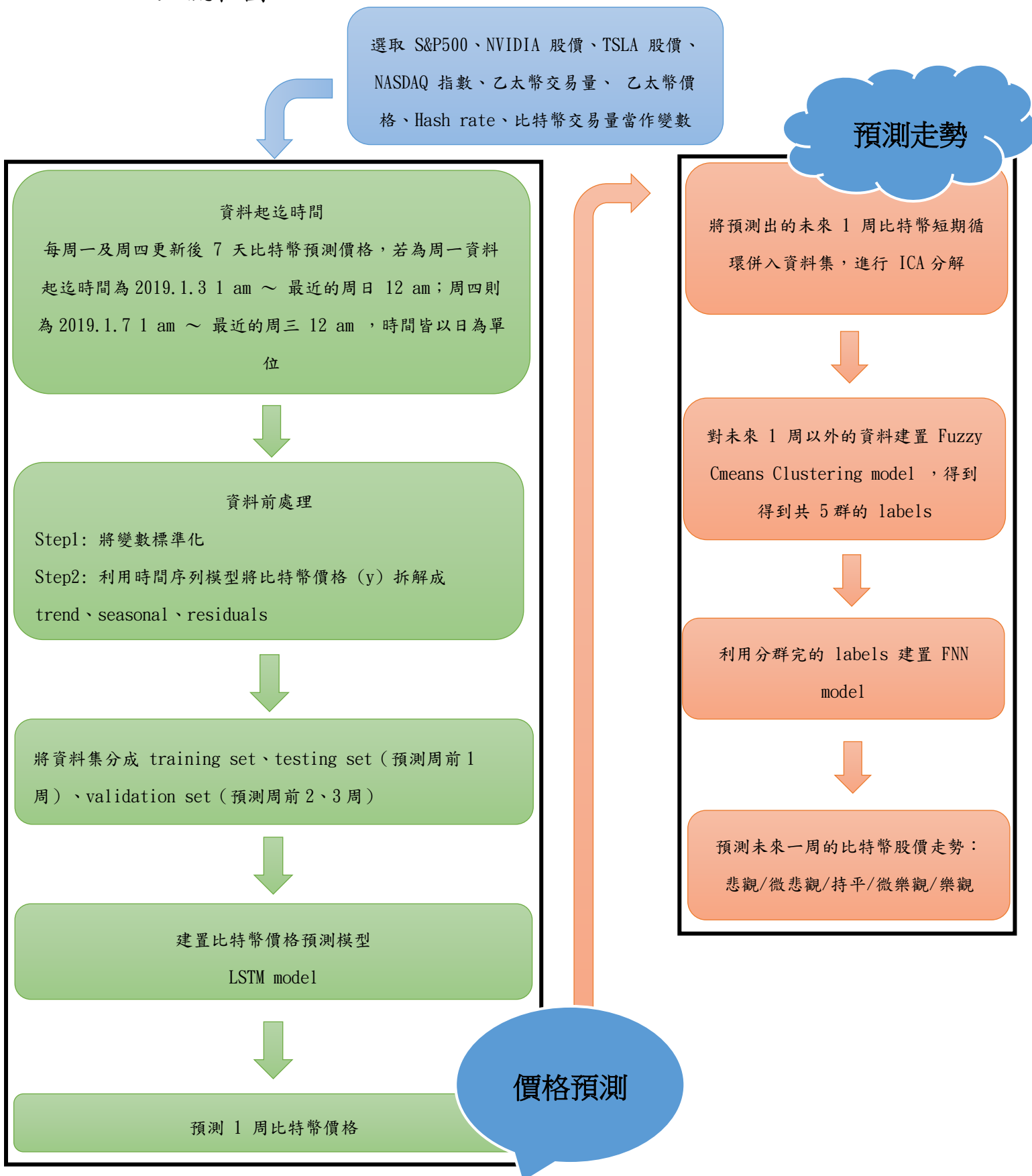
比特幣是區塊鏈中最知名的應用，其主要採用 P2P 的技術。而這個電子加密貨幣最大的特色就是去中心化，跟以往需要仰賴中央銀行的貨幣交易形式，這個應用代表每個人可以算是一家銀行，進行直接的交易，且每筆交易都具有匿名性，可以防止中央銀行不良政策所導致的通貨膨脹。因為比特幣不是由某家銀行統籌大量交易，所以被認為是一個安全系統也是其優點之一。

為什麼「挖礦」、「礦工」、「顯示卡」這些關鍵字與比特幣息息相關呢？原因是比特幣每個節點的交易中區塊鏈協議都需要驗證交易，而驗證交易這個動作需要電腦向網路提供算力去確認交易以保持系統網路正常運行，因此這些自願提供算力的電腦供應者可以獲得比特幣當作獎勵，所以這些電腦被稱作「礦工」。大家為了獲得比特幣進行「挖礦」，當算力越強則越能獲得更多獎勵，這導致需要提升「顯示卡」的效能。

近年來，比特幣的呼聲水漲船高。2017 年時，比特幣在 5 個月內暴漲 740%，導致很多人趨之若鶩的跟上「挖礦列車」，高級顯示卡也在一夕之間變成很搶手，民眾也不惜金錢砸下重本大量購買組成「礦場」。2021 年 3 月時，Tesla 執行長 Elon Musk 宣布，消費者能以比特幣購買其公司的電動汽車，比特幣網路聲量又達到高峰。然而，如此被看好的前景在近期發生巨大的變化。由於美國聯準會(FED)收緊政策的立場比預期更激進，以及全球第二大礦場國家哈薩克內部緊張情勢升溫，投資者紛紛拋售加密貨幣並且退場，導致比特幣價格跌破 41,000 美金，造成比特幣一大利空。

基於如此大幅度震盪的情勢，買賣比特幣就如同投資股票起起伏伏。為了可以提早進場與退場，投資者也會對於比特幣價格進行預測。然而不同於股票的是，比特幣不像股票受政府控制（如漲停板、跌停板）也沒有明顯的時間週期。因此，類似股票的分析方法可能不適用於比特幣上。在這次的計畫中，我們提出結合時間序列與深度學習的方法進行預測：第二章節我們將詳細介紹本次在價格預測與走勢的流程圖；第三、四章節分別介紹在實際應用上所使用的迴歸與預測的方法；第五章節為比特幣實際資料的應用與預測，在最後的第六章節我們也會對於我們的模型進行評估與建議。我們希望不只是使用單純深度學習的方法進行比特幣價格的預測，也希望藉由傳統時間序列的方法給予協助。

2. 流程圖



3. 迴歸分析

3.1. 時間序列模型

在統計上，時間序列被應用在存在趨勢與循環的資料中，且下一個資料點受過往的資料影響。通常來說，時間序列包含 (1)趨勢 (2)季節 (3)短期循環 (4)隨機現象。在此我們考慮時間加成模型 (Additive Decomposition):

$$X_t = m_t + s_t + c_t + \epsilon_t; \quad t = 1, 2, 3, \dots$$

其中:

X_t : 觀測值

m_t : 趨勢

s_t : 季節循環

c_t : 短期循環

ϵ_t : 隨機現象

一般來說，趨勢現象可以使用一次線性迴歸或者二次迴歸等做簡單的估計。舉例: 二次趨勢線性模型可以寫成

$$m_t = a_0 + a_1 t + a_2 t^2; \quad t = 1, 2, 3, \dots$$

a_0, a_1, a_2 為迴歸估計參數， t 為時間點。季節循環(s_t)代表資料存在一個明顯的區間效應，簡單的例子為四季變化。假設季節循環長度為 d ，則

$$s_t = s_{t-d}$$

且常見的季節循環(s_t)估計為:

$$s_t = a_0 + \sum_{j=1}^k (a_j \cos(\lambda_j t) + b_j \sin(\lambda_j t))$$

a_j, b_j 為預估計參數， λ_j 為 \sin 與 \cos 函數的週期長度。這裡可以注意的是，趨勢(m_t)與季節循環(s_t)通常會造成時間數列的不平穩(non-stationary)。在傳統的時間序列模型像 AR, MA, ARMA 都假設在平穩(stationary)時間序列模型假設。故我們會至少去除趨勢與季節循環後再使用以上模型對於剩餘的時間效應進行估計。

3.2. 自我相關(ACF)、偏自我相關(PACF)與 DF 檢定

假設在前一節已將資料的趨勢與季節循環移除，接著可以觀察剩餘趨勢 (1) 是否為隨機現象 (2) 若不為隨機現象，資料是否為穩定? 假設剩餘趨勢為隨機現象，我們可以稱資料只存在趨勢與季節效應，分析可以到此結束。若仍保有趨

勢，則需要檢定剩餘趨勢是否穩定。而簡單檢查(1)的方法就是藉由 ACF 與 PACF 圖觀察資料的趨勢。若 ACF 或 PACF 在兩期後(lag = 2)有顯著情形，則可以判定剩餘趨勢至少有 ARMA(p, q)效應。至於如何確認平穩性，我們可以藉由 ADF 檢定判斷。ADF 檢定過程如下：

考慮一個已經移除趨勢的差分模型：

$$\Delta Y_t = \widehat{\beta}_0 + \widehat{\gamma} Y_{t-1} + \sum_{i=1}^{p-1} \widehat{\gamma}_i \Delta Y_{t-i}$$

其中， $\widehat{\beta}_0, \widehat{\gamma}$ 為估計參數， $\Delta Y_{t-i} = Y_{t-i} - Y_{t-i-1}$ 即差分值。則假設檢定可以寫成

$$H_0: \gamma = \beta_1 + \beta_2 + \cdots + \beta_{p-1} = 0 \text{ (有單根存在)}$$

$$H_1: \gamma < 0 \text{ (無單根存在)}$$

$$\text{檢定統計量: } d = \frac{\widehat{\gamma}}{s.e.(\widehat{\gamma})} \sim \tau \text{ (DF distribution)}$$

檢定規則： $p - value = P(DF < d) < \alpha = 0.05$ 則拒絕 H_0 。

3.3. 長短期記憶(Long Short-Term Memory; LSTM)

長短期記憶(LSTM)是一種 RNN 模型，最早由 Sepp Hochreiter 和 Jurgen Schmidhuber 在 1997 年提出。LSTM 適合使用在時間序列且存在延遲的資料，其說明如下：

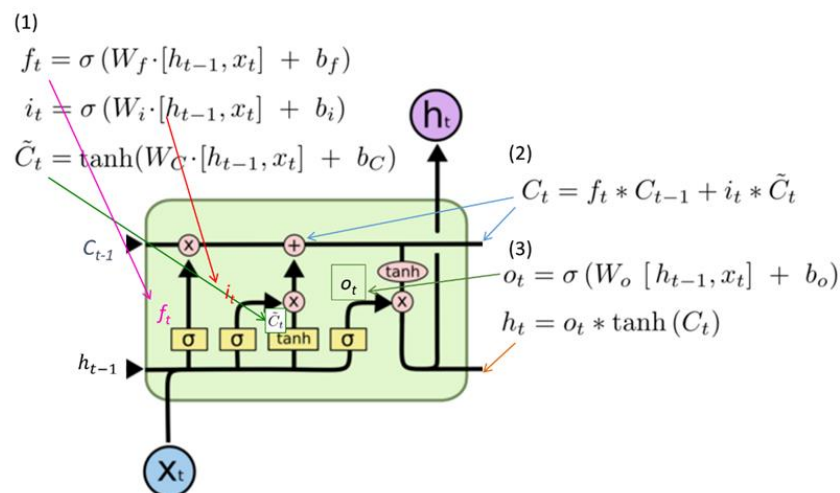


圖 1 單一個 LSTM cell 運算流程

(來源: [Understanding LSTM Networks -- colah's blog](https://colah.github.io/posts/Understanding-LSTMs/))

其中:

x_t 為 t 時間點送入的向量

h_t 為 t 時間點的狀態值

C_t 為 t 時間點 memory cell 的儲存值

- (1) LSTM 的第一步決定要從單元狀態丟棄什麼訊息，這個決定由 f_t 做出，輸出值介於 0 和 1，0 表示完全忘記，1 表示完全記住。
- (2) 第二步決定要在單元狀態中儲存什麼新訊息。它有兩部分：首先， i_t 的 sigmoid 決定更新哪些值，接下來， \tanh 建立新的候選向量，在下一個步驟中，把兩者結合起來對 memory cell 更新。
- (3) 最後計算輸出值(o_t)和更新狀態(h_t)

若將多個 LSTM 組合起來，則 many-to-many 的 LSTM 的示意圖如圖 2：

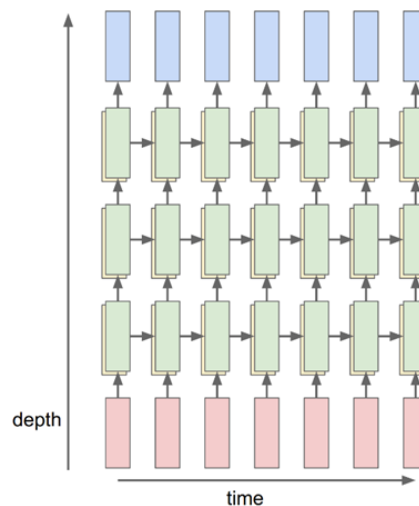


圖 2 many-to-many LSTM 示意圖

(來源: cs231n.stanford.edu/slides/2021/lecture_10.pdf)

4. 走勢分群

4.1. 獨立成分分析(Independent Component Analysis; ICA)

獨立成分分析(ICA)主要是希望可以把資料區分出不同的成份來源，因此時常被用來處理混和訊號。不同於主成分分析(PCA)，PCA 是從資料中結合特徵而 ICA 是分離特徵。因此，兩者在假設上也有所不同。假設模型可以寫成 ICA 的形式：

$$x = As + \xi$$

ICA 的目標是根據 observed data x 去找到 mixing matrix A 和因子 s ，且在模型中假設所有因子 s 皆是統計獨立且服從常態分布。如果 s 服從非常態，則希望找到 w 從而 $s = w^T x$ ，使得 s 相互獨立就可分離出來源資料 s 。接著，根據 Darmois - Skitovitch Theorem:

$$x_1 = \beta_{11}s_1 + \cdots + \beta_{1n}s_n$$

$$x_2 = \beta_{21}s_1 + \cdots + \beta_{2n}s_n$$

假設 s_1, s_2, \dots, s_n 是相互獨立的來源，對於兩隨機變量 x_1, x_2 若獨立，則對任意 $\beta_{1i}\beta_{2i} \neq 0$ ，其 s_i 服從常態分佈。所以，ICA 求解系數的流程為：

Step1: 預處理：

中心化(Centering)：將 x 轉換成以 0 為中心。

白化(Whitening)： $\tilde{x} = ED^{-\frac{1}{2}}E^T x = ED^{-\frac{1}{2}}E^T A s = \tilde{A} s$

$$\tilde{x}\tilde{x}^T = ED^{-\frac{1}{2}}E^T A s s^T A^T E D^{-\frac{1}{2}}E^T = I$$

Step2: 迭代算法：

經過白化後，只要找到最優方向 w 使得改方向的非常態性最大 $\max J(w^T x)$ ，

其中 $J(y) = H(y_{\text{gauss}}) - H(y)$ 又 $J(y) \approx \frac{1}{12} E(y^3)^2 + \frac{1}{48} \text{kurt}(y)^2$ ，而在

fastICA 中使用的是 $J_G(w) = \left[E(G(w^T x)) - E(G(v)) \right]^2$ 。

透過限制 $\|w\|^2 = 1$ 求導數

$$\frac{\partial}{\partial w} (J_G(w) - \beta(\|w\|^2 - 1)) = 2E(xG'(w^T x)) - 2\beta w = 0$$

$$\Rightarrow \beta = E(w_0^T x G'(w_0^T x))$$

利用牛頓法求

$$w_{n+1} = w_n - \frac{E(xG'(w_n^T x)) - \beta w_n}{E(G''(w_n^T x)) - \beta I}$$

4.2. Fuzzy Cmeans 分群法

Fuzzy Cmeans Clustering 利用隸屬值 membership value 與 fuzzier 來進行模糊化，描述了樣本屬於哪個類別與樣本的隸屬程度，類似機率的概念，分群流程如圖 3，其中 u_{ij} 為第 j 個觀測值對第 i 個群心的 membership value， m 為

fuzzier 用來調控對於群心計算的影響力， K 為分群個數， N 為觀測值個數， J 則為目標函數，詳細步驟如下：

Step1：隨機產生 $i \times j$ 個 u_{ij} ，並滿足 $\sum_{i=1}^K u_{ij} = 1$

Step2：利用這些 u_{ij} 計算初始的 K 個群心 C_i

Step3：透過將目標函數及限制式 $\sum_{i=1}^K u_{ij} = 1$ 代入拉格朗日乘數計算出新的 u_{ij} ：

$$u_{ij} = \frac{\frac{1}{\|X_j - C_i\|^{\frac{2}{m-1}}}}{\sum_{i=1}^K \frac{1}{\|X_j - C_i\|^{\frac{2}{m-1}}}}$$

Step4：將新的 u_{ij} 代入目標函數計算出 $J(k)$

Step5：判斷每次迭代所計算出的 $J(k)$ 與前一次 $J(k-1)$ 的差若小於我們所設定的門檻，則停止迭代，反之則重新計算 K 個群心 C_i 。

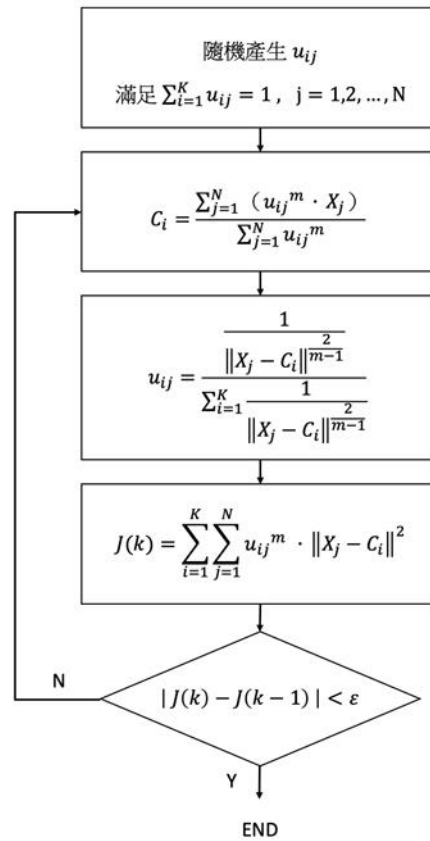


圖 3 Cmeans 分群流程圖

4.3. 前回饋類神經網路(Feedforward Neural Network; FNN)

在神經網路中，神經元可以接收來自其他神經元的輸出或外部數據，經過 **activation function** 計算後產生輸出，而神經網路中的權重表示輸出神經元與接收神經元之間聯繫的強弱，會藉由訓練過程自動調整趨於穩定，也是神經網路主要的訓練對象。前饋神經網路（FNN）是最簡單的神經網路模型，每一層計算完成的輸出，只會單方向往前傳遞給下一層的神經元，也就是說輸入與輸出是獨立沒有關係的，主要結構如圖 4。

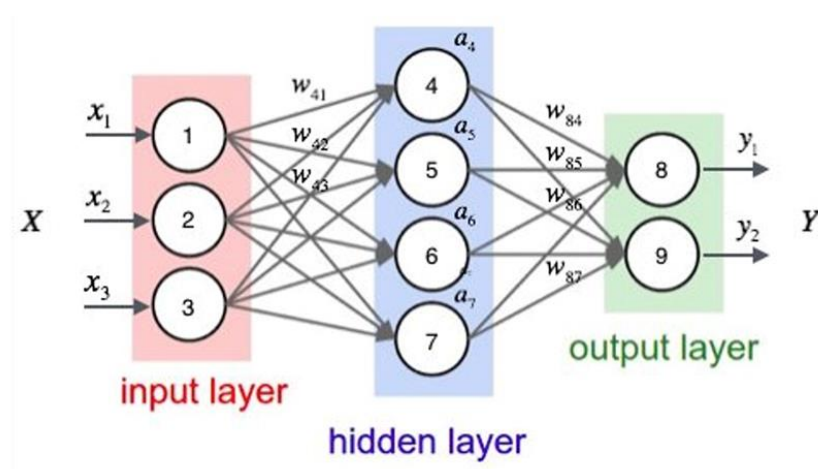


圖 4 FNN 神經網路示意圖

5. 比特幣預測的實際應用

5.1. 比特幣價格預測

首先，對於一星期的兩次預測週期我們設置為：(1) 週四 01:00 a.m. ~ 下周四 00:00 a.m. (起訖時間：2019.1.3 1 am ~ 週四 00:00 a.m.) (2) 週一 01:00 a.m. ~ 下週一 00:00 a.m. (起訖時間：2019.1.7 1 am ~ 週一 00:00 a.m.)，皆為美國時間。

(1)對於反映變數 Y (即：比特幣價格)

我們首先使用 `statsmodels.tsa` 進行時間序列的處理，對於在 3.1 的時間序列我們在比特幣價格取過對數後，其趨勢、季節循環與短期循環結果可見圖 5。而在移除趨勢及季節循環後，對於剩餘的趨勢(短期循環+隨機現象)我們可以畫 3.2 的 ACF 與 PACF 圖(如圖 6)，發現在 ACF 的部分呈現指數遞減且 ACF 與 PACF 皆在 $\text{lag} = 2$ 之後顯著，故可以斷定剩餘趨勢至少仍存在 ARMA 效果。最後，再藉由 ADF 檢定檢查其是否為穩定時間序列。而 ADF 檢定後可以得檢

定的 $p - value < 2e - 16 < \alpha = 0.05$ ，因此拒絕虛無假設，確定此為一定態時間序列。

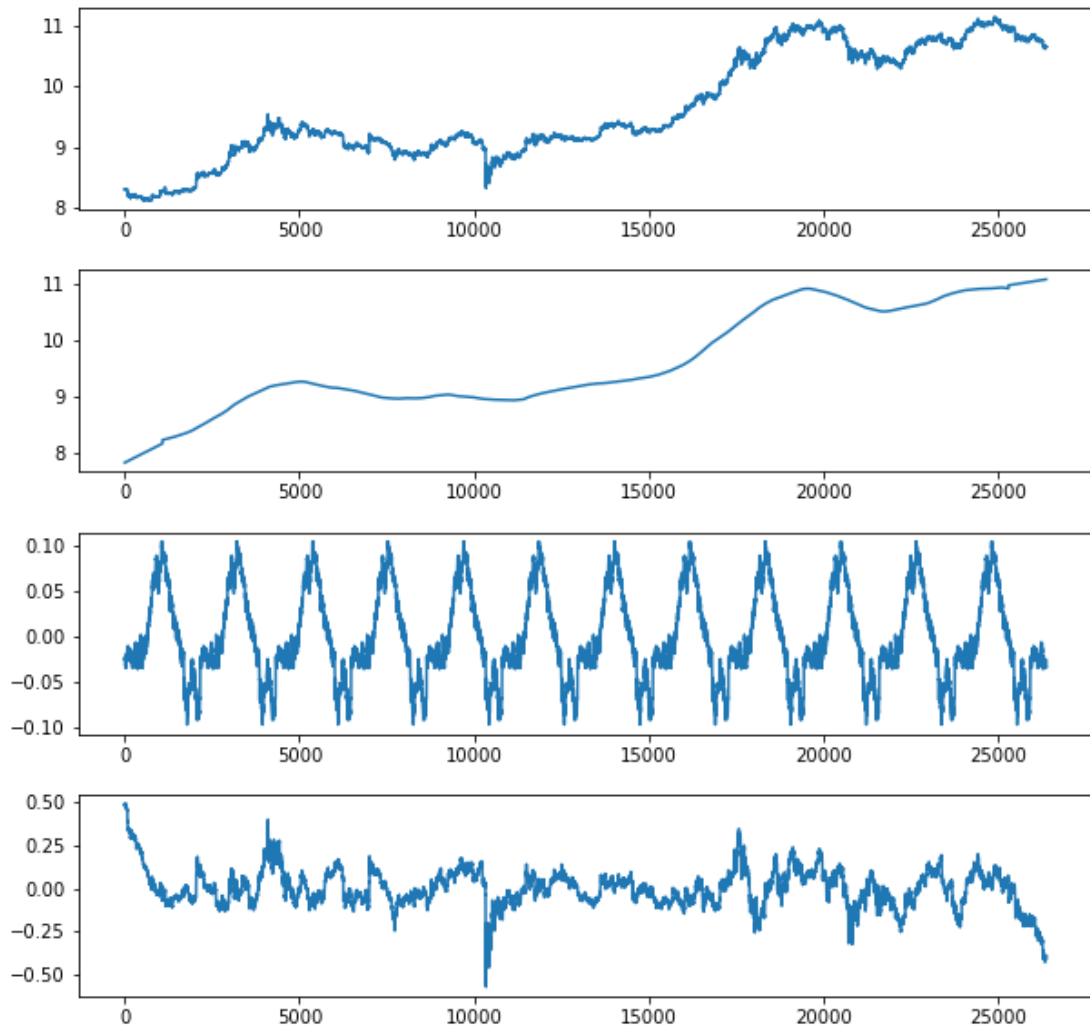


圖 5 比特幣價格在取對數後的趨勢、季節以及短期循環

圖片依序為 $\log(\text{bitcoin})$ 、趨勢、季節循環與短期循環

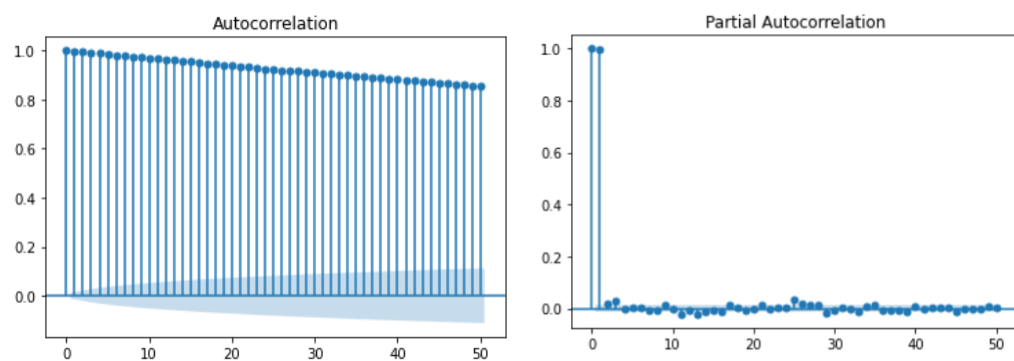


圖 6 $\log(\text{bitcoin})$ 的剩餘趨勢的 ACF 與 PACF 圖

(2)對於模型變數 X (即：基本面、市場面、消息面)

我們分別在基本面、市場面與消息面的三大面向分別挑選變數，而變數來源可能為與比特幣價格走勢類似(例如：股票股價，見圖 7)或者與比特幣息息相關(例如：以太坊自身或以太坊的供需)。變數 X 分別為：

1. 基本面：Hash rate, 比特幣交易量 (Bitcoin Volume)
2. 市場面：NVIDIA 股價、TSLA 股價、NASDAQ 指數、S&P500、以太坊交易量 (Ethereum Volume)、以太坊價格 (Ethereum Price)
3. 消息面：Google Trend

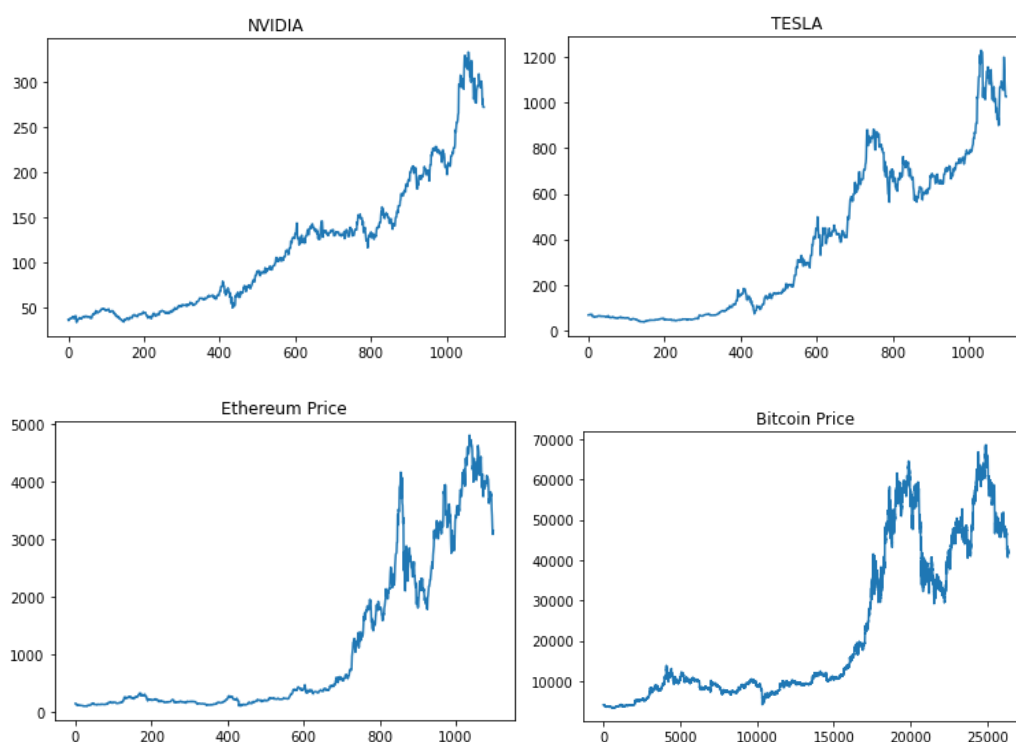


圖 7 模型變數 X 與比特幣價格趨勢圖 (以 NVIDIA、TESLA、以太坊為例)

在將變數各自標準化後 (Z-score normalization)，我們使用標準化變數 X 對時間序列剩餘趨勢進行預測，也就是說，在趨勢與季節循環移除後，模型只需對於取對數後比特幣價格的短期循環進行預測即可，而我們發現這樣可以有效的訓練模型並且減輕訓練方面的負擔。對於預測模型的選用，我們選擇常用於預測時間序列資料的 RNN，且 RNN 模型我們架設兩層雙向的 LSTM (Bidirectional LSTM)。LSTM 的 hidden units 約落在 [800, 500, 300, 100] 不等，取決於當週微調後的最佳組合；LSTM 模型架構可以參考圖 8。此外，針對 LSTM 的輸入 (Input) 我們也設計四種不同的架構，分別為：

Model 1: X non-overlapped; 1 day to 1 day; Y uses residuals training

Model 2: X non-overlapped; 1 week to 1 week; Y uses residuals training

Model 3: X overlapped; 2 weeks to 1 week; Y uses residuals training

Model 4: X non-overlapped; 1 day to 1 day; Y uses log price

```
model3.summary()
```

Model: "sequential_27"

Layer (type)	Output Shape	Param #
bidirectional_52 (Bidirectio	(None, 14, 1000)	2040000
bidirectional_53 (Bidirectio	(None, 600)	3124800
dense_48 (Dense)	(None, 168)	100968

Total params: 5,265,768
Trainable params: 5,265,768
Non-trainable params: 0

圖 8 LSTM 模型架構 (units = 500, 300)

而四者之中相對於 model 4 使用的 Y 為比特幣對數價格，其餘皆為對數後的剩餘趨勢 (residual)。Model 1 及 model 2 的差異為天對天以及禮拜對禮拜，model 2 及 model 3 的差異在 model 3 使用兩個禮拜的資料而 model 2 只使用一個禮拜。因我們比較常使用 model 3，故 model 3 的示意圖可參考圖 9，這邊可以注意藍色與橘色為不同資料點各自對應的關係，並非藍色對齊橘色。

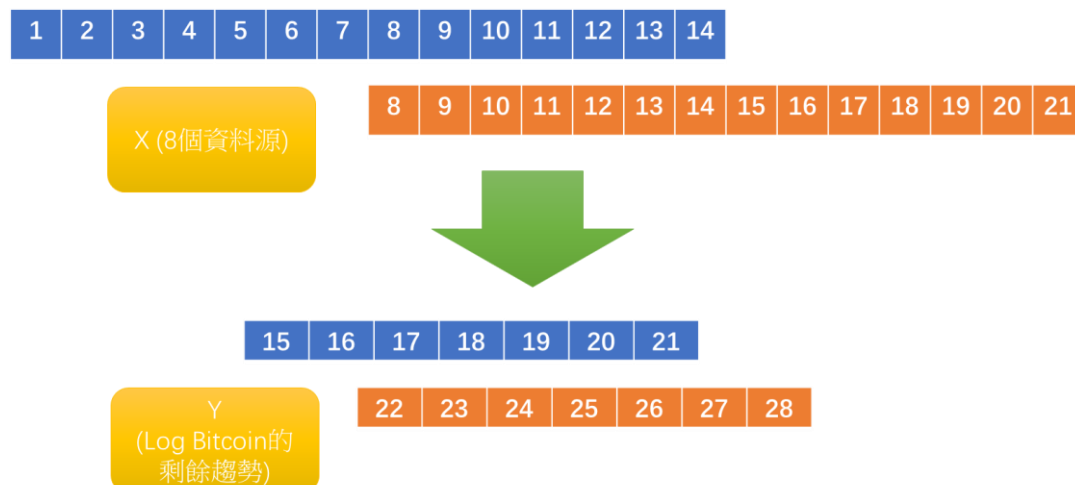


圖 9 Model 3 資料變數 X 與反應變數 Y 的對應關係圖

(3) 模型預測後價格的還原以及其他處理

如同先前提到，我們使用 LSTM 去預測比特幣價格取對數後剩餘短期循環並非原始價格，故我們需要還原價格作為真正的比特幣價格預測。觀察我們施行

的步驟為：取 \log → 移除趨勢、季節 → LSTM 預測剩餘趨勢，則反向運作變成：LSTM 預測剩餘趨勢 → 加回趨勢與季節循環 → 取 \exp 。參考的預測結果可見圖 10。然而我們可以藉此發現兩個問題，分別為價格的最終預測有很大的震盪以

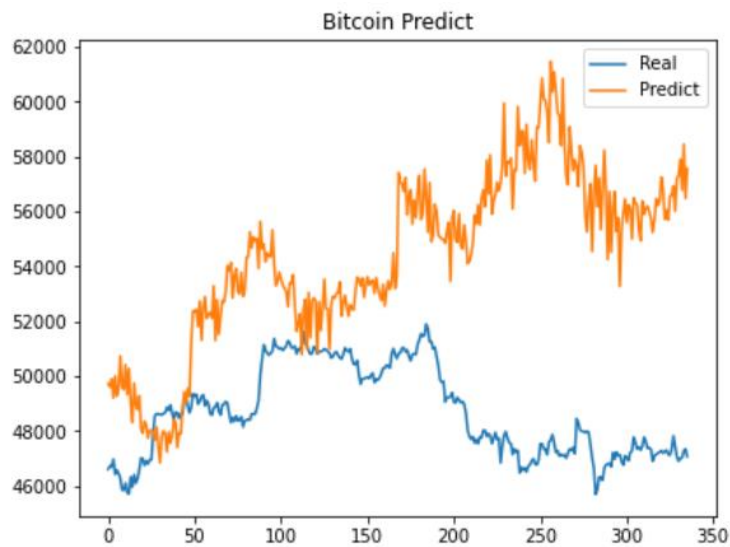


圖 10 比特幣預測結果範例

及第一筆預測值無法對齊第一筆真實價格的窘境。因此，對於價格過於震盪我們做平滑 (smoothing) 以及對齊的部份我們假設比特幣在一分鐘內不會有太大的震盪，因此我們將第一筆預測值對齊 12:59 a.m. 的真實價格。最終的結果可見圖 11。

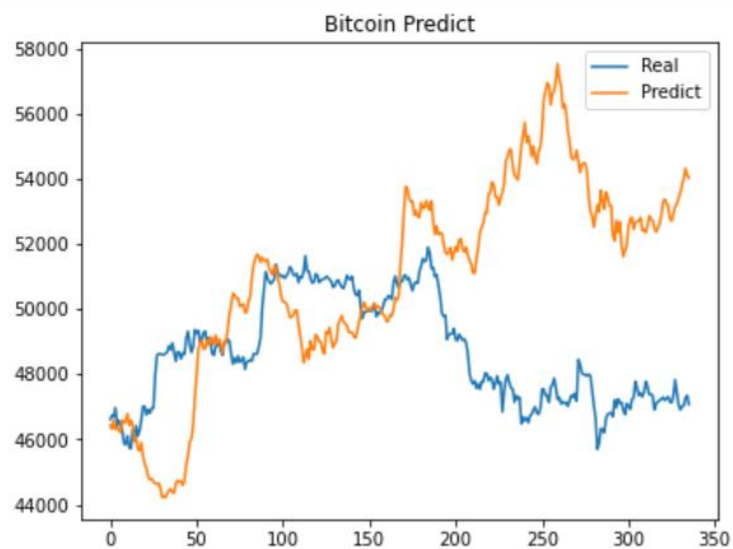


圖 11 最終比特幣預測結果示意圖

5.2. 比特幣價格趨勢預測

對於比特幣價格趨勢的預測，我們先將剩餘趨勢以 7 天共 168 小時為一筆觀測值進行 4.1 節的獨立成分分析 (ICA)。我們假設比特幣一週的趨勢是由 7 個獨立成分組成 (如圖 12)。

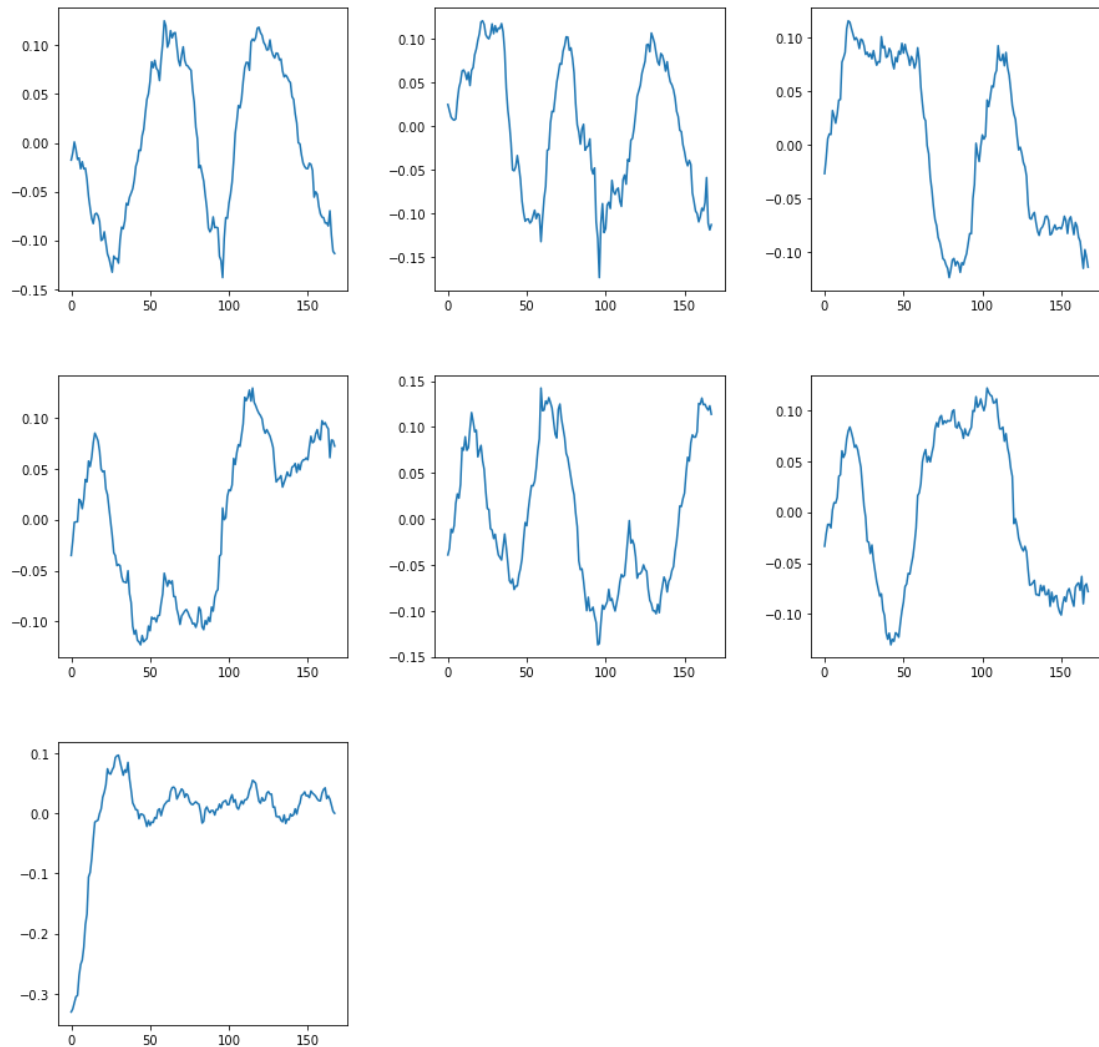


圖 12 ICA 分解後的 7 個獨立訊號波段

在 ICA 分解後可以得到混合係數組成的混合係數矩陣 (mixing matrix)，此矩陣代表對於每個波段的組成係數。假設波段 Y 為上述的 7 個獨立成分組成，則可以寫成：

$$Y = a_1ICA_1 + a_2ICA_2 + \cdots + a_7ICA_7$$

則稱 a_1, a_2, \dots, a_7 為混合係數，且我們將使用於後續的分群動作。相較於 Kmeans 分群，我們反而使用 4.2 節的 Cmeans 分群法因為我們覺得波段彼此相似故需要考慮混淆程度的軟性分群 (fuzzy clustering)。將剩餘趨勢分群後對應回原始比特幣價格的分群範例可見圖 13 與圖 14。這裡需要注意的是我們是將剩餘趨勢分

群，所以不是藉由價格高低進行評斷，解釋方式也為將固定效應(趨勢、季節循環) 移除後剩餘的變化進行比較。假如在固定時間點比特幣價格本身就會大漲，那麼價格上漲的原因就非買賣所導致的，這也因此讓我們覺得直接使用價格進行分群是不對的。

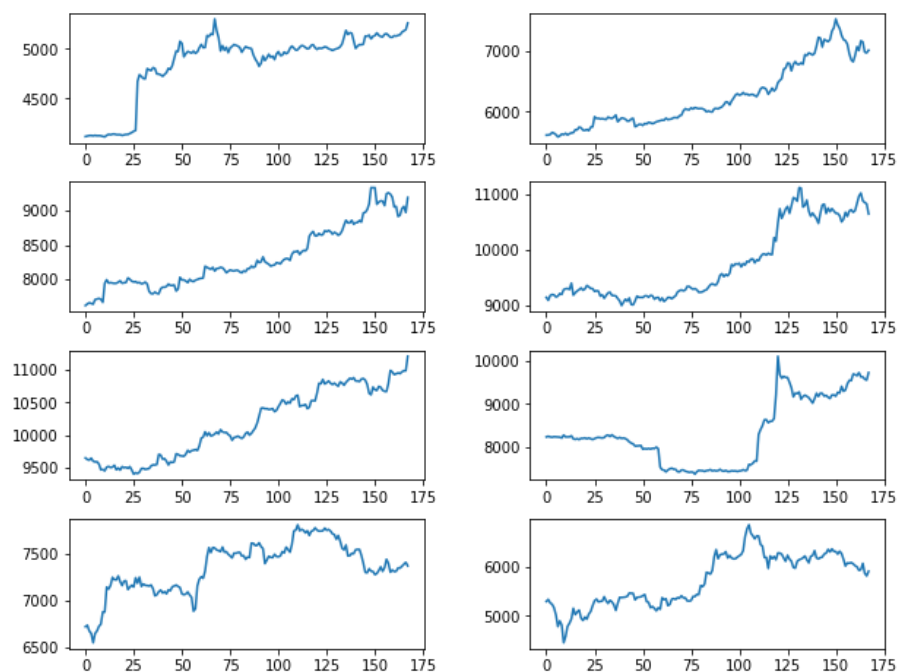


圖 13 比特幣非常樂觀走勢

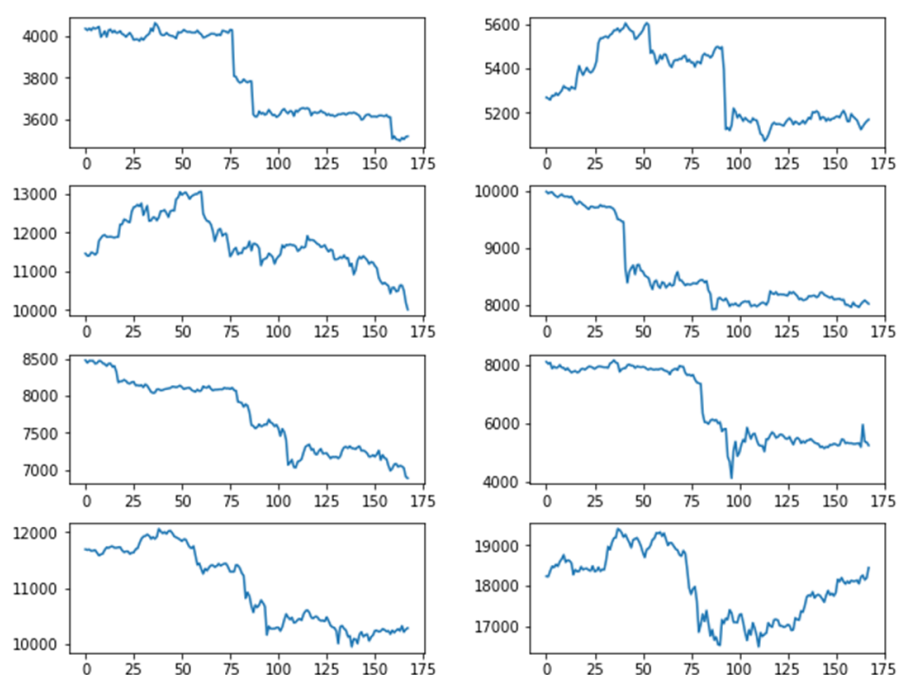


圖 14 比特幣非常悲觀走勢

在 Cmeans 分群後，我們將分群的結果轉為分類的 label 並且建立簡單的類神經網路。類神經網路架構主要為兩層的 hidden layers 且 hidden units 分別為 15 以及 8 個，activation function = relu。架構可以參考圖 15：

Model: "sequential_4"

Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 15)	120
dense_6 (Dense)	(None, 8)	128
dense_7 (Dense)	(None, 5)	45
Total params: 293		
Trainable params: 293		
Non-trainable params: 0		

圖 15 類神經網路架構

最後，將預測結果的 ICA 混和係數放入類神經網路後即可得下週價格走勢的分類。

6. 結論與建議

我們結合了時間序列以及 LSTM 進行比特幣價格的預測，並使用 ICA、Cmeans 與 FNN 進行比特幣趨勢的預測。整合共 8 次的預測結果來看 (只截止至 1/10 12:00 p.m.)，我們發現到在第三、五次預測是不準的，第一、二、四、七次預測在前三天半都有良好的預測結果，但隨著時間越長後三天半的預測有偏離真實價格的情形。第七次後半以及第八次因當時時間未到故沒有真實價格。預測結果與真實價格的比較圖可見圖 16 至圖 19。因此，我們對於模型列了幾個優缺點：

優點：

- (1) Bitcoin 存在許多人為因素在其中，若讓 LSTM 直接預測價格會訓練不起來 (即便取對數)。
- (2) 考量到趨勢與季節循環，即使當禮拜的短期循環波動程度不大，仍可以靠時間序列的成分給予價格上的波動。

缺點：

- (1) Bitcoin 沒有固定季節循環，也就是說如何訂定季節循環長度沒有正確答案 (在此假設 3 個月)。

(2) 預測結果有嚴重震盪與季節循環的估計有關，且因為價格臨時的大跌導致趨勢 (trend) 無法即使反應過來，進而導致有起始點對不齊的現象。

對於缺點我們建議可以使用 residual 對 residual 的方式，也就是將變數 X 移除自身趨勢以及季節循環後的短期循環對 $\log(\text{bitcoin})$ 的短期循環進行訓練，或者假設更複雜的時間序列模型如 ARCH、GARCH 等具有風險貼水模型進行 component 的分割，或許建立多個 RNN 模型分別對各自的時間趨勢進行預測可以有更好的效果。

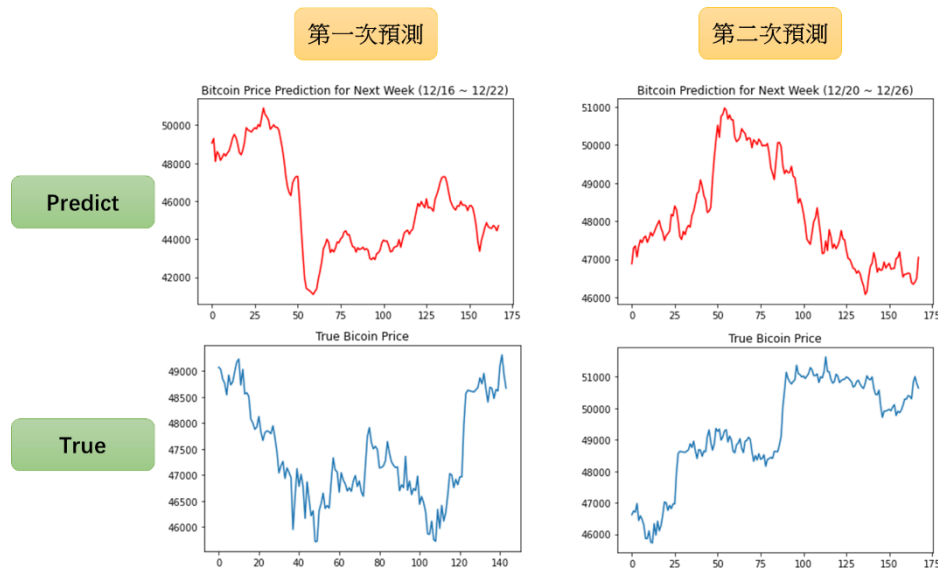


圖 16 比特幣價格真實預測（第一、二次）

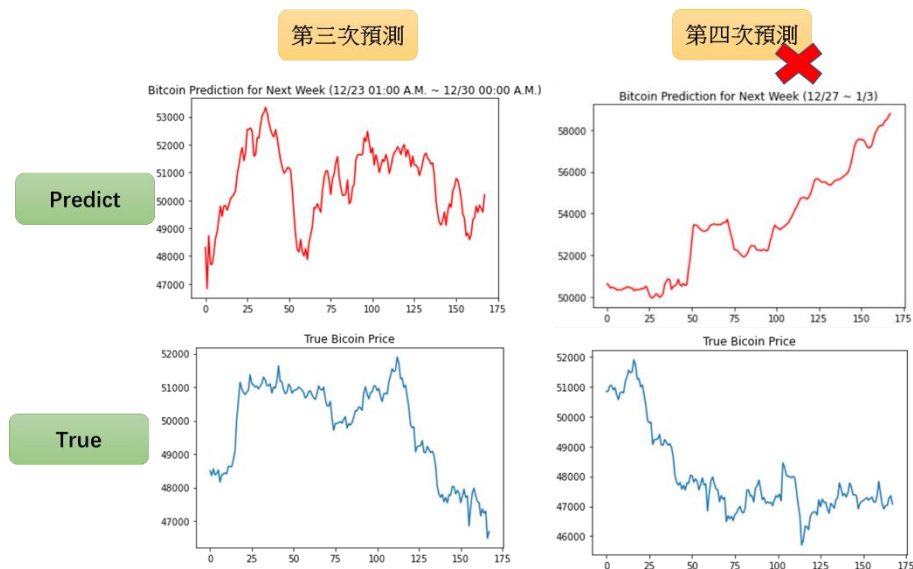


圖 17 比特幣價格真實預測（第三、四次）

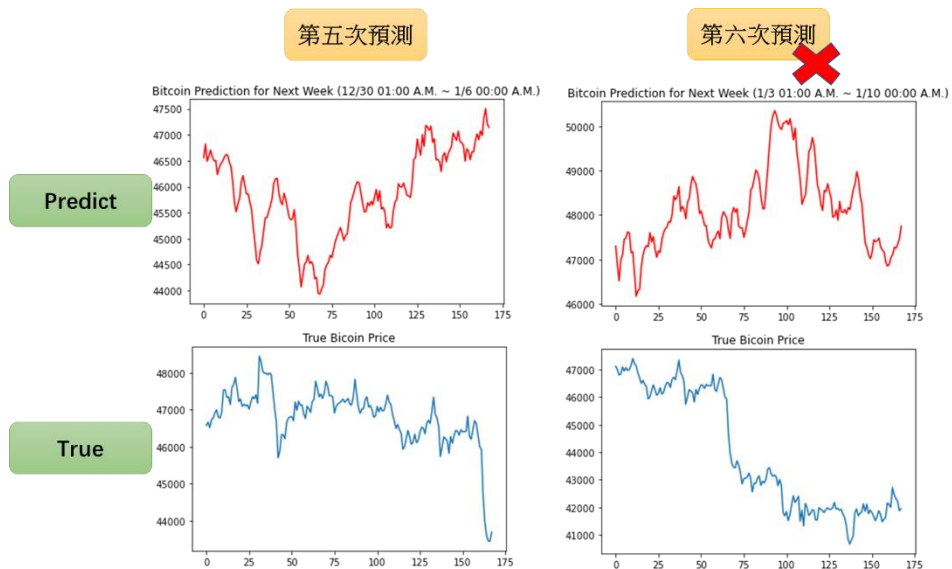


圖 18 比特幣價格真實預測（第五、六次）

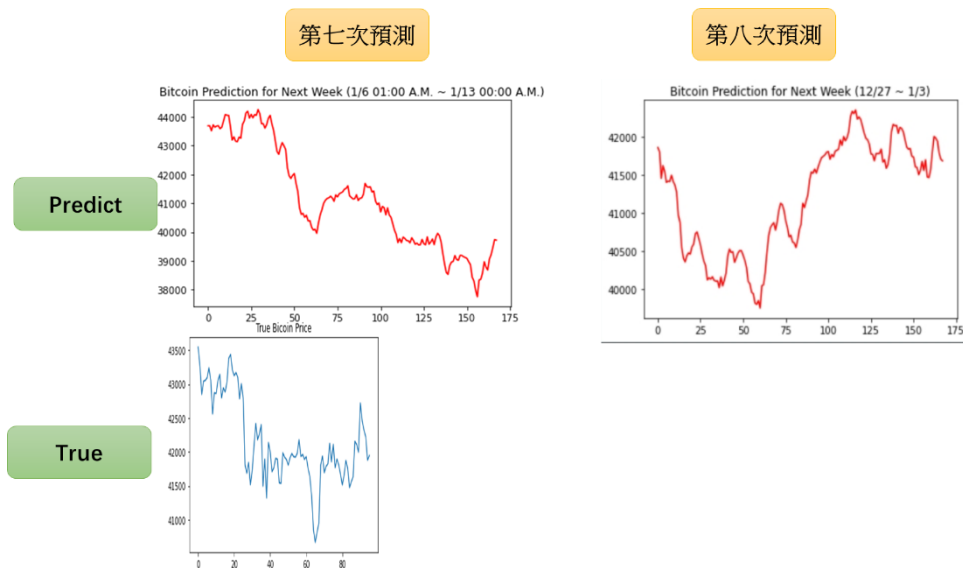


圖 19 比特幣價格真實預測（第七、八次）