

多變量時間序列分析

-以 Horizon Zero Dawn 遊戲為例

109354003 統研所二 吳書恆

109354018 統研所二 吳承臻

June 15, 2022

目錄

第一節	研究動機與目的	2
一、	研究動機	2
二、	研究目的	2
第二節	資料介紹	2
第三節	資料分析	3
一、	初步分析	3
二、	VAR 模型	5
三、	VARMA 模型	8
第四節	預測表現	11
第五節	結論與建議	12
附錄		14

第一節 研究動機與目的

一、研究動機

遊玩遊戲時，畫面流暢度是影響遊戲體驗的因素之一。幀率 (Frame Per Second，簡稱 FPS) 為衡量遊戲流暢度的度量，定義為一秒鐘畫面含有靜態圖片的張數，FPS 越高畫面流暢度也會越高。大多數電影的 FPS 皆大於均 24，但對射擊遊戲或競速遊戲來說，FPS 若少於每秒 30 張遊戲就會顯得不連貫，某些高畫質的遊戲甚至會犧牲流暢的 60FPS 來達到更佳的畫面。這些應用說明 FPS 與電腦硬體有關，當中央處理器 (Central Processing Unit，簡稱 CPU) 或是圖形處理器 (Graphics Processing Unit，簡稱 GPU) 的效能增加，FPS 也會跟著增加。有鑑於此，我們考慮 FPS、CPU 和 GPU 相關變數對彼此進行預測。

二、研究目的

為了找出 FPS、CPU 與 GPU 相關變數之間的關係並進行預測，我們建立三個變數的多變量時間序列分析，並且考慮多變量 AR 模型 (Vector Autoregressive model，簡稱 VAR) 與多變量 ARMA 模型 (Vector Autoregressive Moving-average Model，簡稱 VARMA)。兩種方式的建模皆會列舉出可能的候選模型來檢查其殘差，最後在對殘差診斷較佳的模型進行預測能力表現的評估。

第二節 資料介紹

資料來源為遊玩 2K 畫質的《地平線：期待黎明》時，透過監測軟體 HWinfo 記錄特效全開的 FPS、CPU 時脈與 GPU 時脈，其中 CPU 與 GPU 時脈分別反映 CPU 與 GPU 的效能，單位為百萬赫茲 (MHz)，後續分別用 CPU 與 GPU 作為簡稱。資料時間為每 2 秒紀錄一次，共有 3193 期。在後續分析中，僅會採用前 3183 筆資料進行訓練與建模，後 10 筆資料則用來進行測試與預測。

第三節 資料分析

一、初步分析

首先觀察 FPS、CPU 與 GPU 三個變數各自的時間序列圖，如圖 1，可發現 CPU 與 GPU 在某些時刻的數值與整體平均差距甚遠，反映兩變數存在離群值，尤其 GPU 整體變動不大，但其離群值都落於整體趨勢的下方(遠低於整體平均)。然而 CPU 與 GPU 離群值的時間點附近有部分重疊，說明這些離群值的交叉相關性(cross correlation)仍舊存在，雖然這些離群值

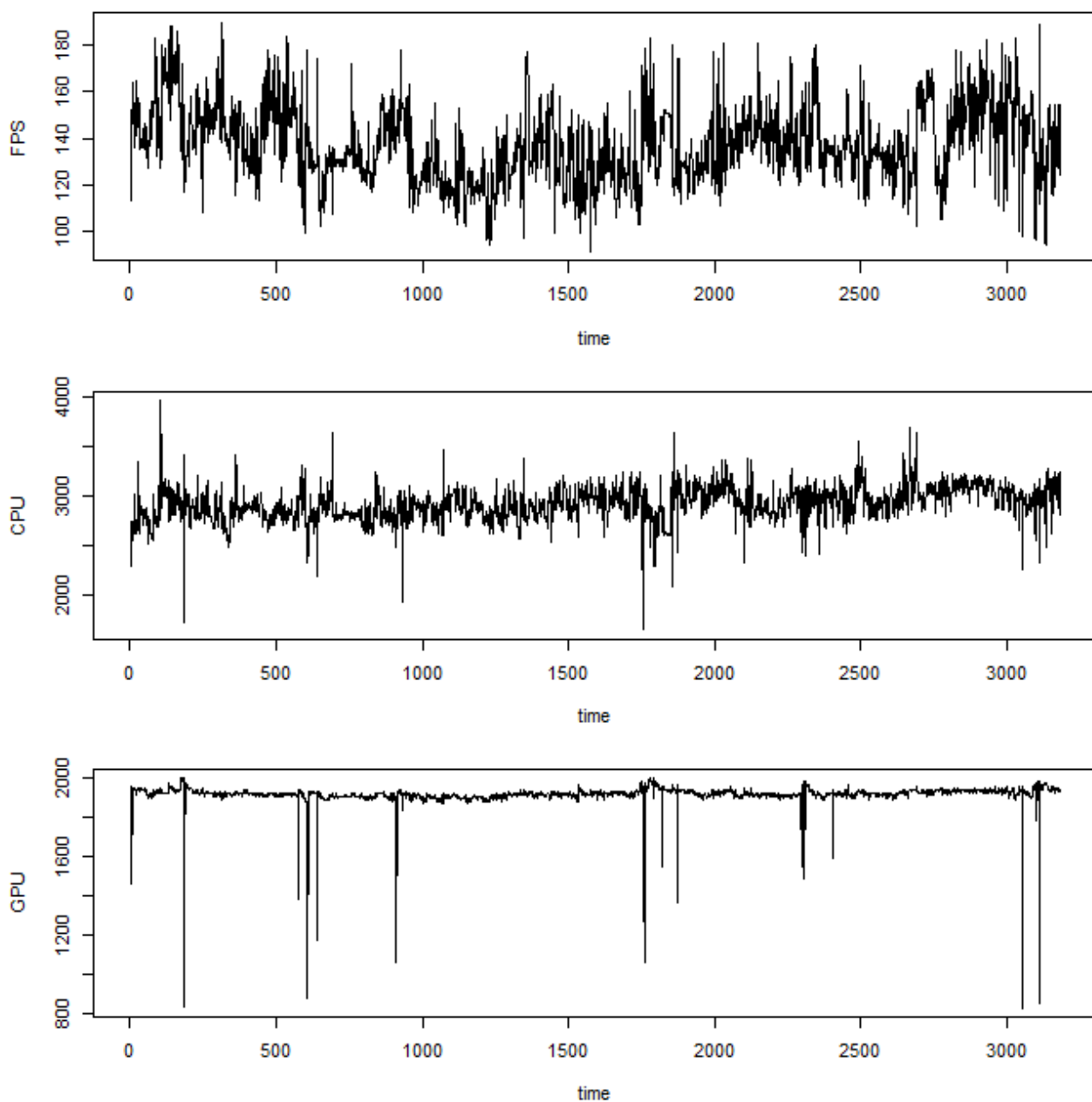


圖 1: FPS、CPU 和 GPU 的時間序列圖

槓桿作用較大，但可能並不是敏感點，因此後續分析仍會納入這些離群值。

接著觀察這三個變數的交叉相關性函數圖 (Cross-Correlation Function，簡稱 CCF)，CCF 為測量時間序列資料的線性動態相依性 (dynamic dependence)。此圖的說明可分為斜對角與非斜對角部分，斜對角由左上到右下分別為 FPS、CPU 與 GPU 各自的 ACF 圖，發現 FPS 與 CPU 下降速度緩慢，GPU 則在滯後 (後續值皆稱 lag) 為 3 以後下降至不顯著的範圍，圖 3 為三變數各自的 PACF 圖，發現三變數大多在前幾個 lag 比較顯著，因此我們考慮使用多變量的 AR 模型來描述此現象；倘若我們考慮 FPS 與 CPU 的 PACF 沒有顯著的斷點，那麼我們考慮使用多變量的 VARMA 模型來描述此現象。另外，由於三個變數的迪基福勒檢定 (Augmented Dickey-Fuller test，簡稱 ADF 檢定) 皆有通過，因此不再做差分。

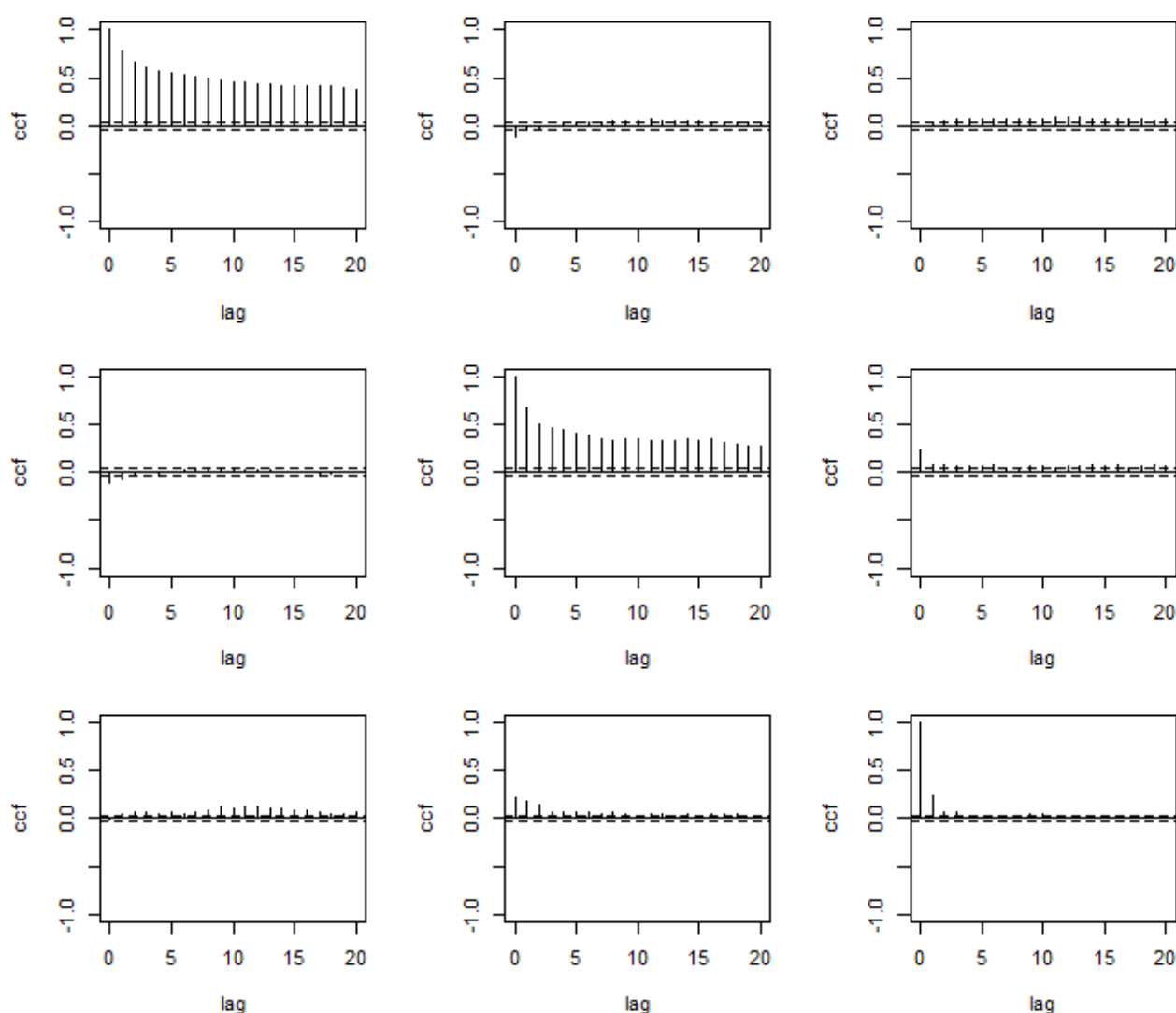


圖 2: FPS、CPU 和 GPU 的交叉相關性矩陣圖

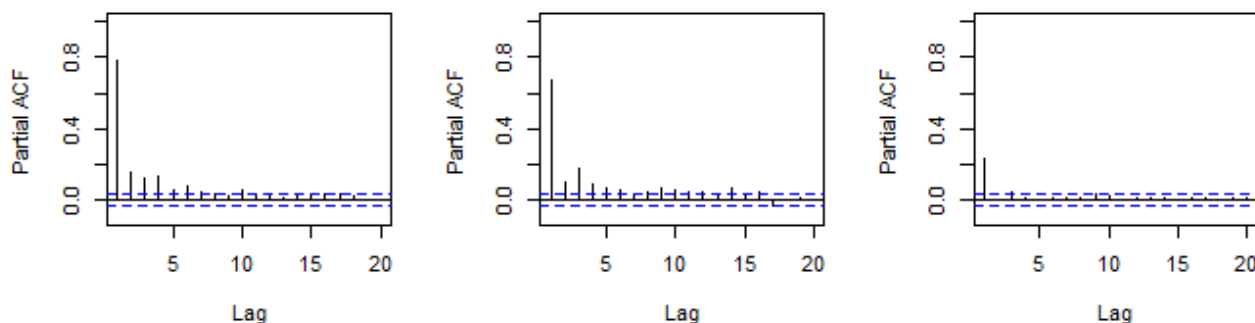


圖 3: FPS、CPU 和 GPU 各自的 PACF

非斜對角部分為兩兩變數之間的 CCF，第一列中間與右邊的圖分別描述當期的 FPS 與不同 lag 的 CPU、當期的 FPS 與不同 lag 的 GPU 的 CCF，其餘以此類推。以右上角的圖為例，可看到 CCF 大多都在虛線之上 (虛線為 CCF 信賴區間)，說明當期的 FPS 與不同 lag 的 GPU 具有正向動態相關，而且是顯著的，右邊中間的圖也反映當期的 CPU 與不同 lag 的 GPU 有同樣的情形；左下角的圖則反映當期的 GPU 與不同 lag 之下的 FPS 具有顯著的正向動態相關，其餘 CCF 圖則沒那麼顯著。

二、VAR 模型

首先資料分析考慮 VAR 模型，根據 Tsay (2013) 對 VAR 模型次數的選擇，可逕行採用 AIC、BIC 與 HQ 選模統計量來判斷。圖 4 為三種選模統計量在不同 VAR 模型次數之下的數值，AIC 在次數為 12 時有最小值，BIC 與 HQ 則在次數為 4 時有最小值，不過圖中可發現，三種選模統計量在次數為 2 時幾乎與較大次數的數值差不多。由於次數 12 稍嫌過大，因此我們僅考慮 VAR(2) 與 VAR(4) 的模型來比較。

移除部分不顯著係數的 VAR(2) 模型配適結果為

$$\mathbf{z}_t = \begin{bmatrix} -8.428 \\ 1108. \\ 1336. \end{bmatrix} + \begin{bmatrix} 0.668 & 0.003 & 0.006 \\ -0.646 & 0.615 & -0.256 \\ 0.131 & 0.037 & 0.198 \end{bmatrix} \mathbf{z}_{t-1} + \begin{bmatrix} 0.146 & 0 & 0.006 \\ 0.899 & 0.110 & 0.077 \\ 0.149 & 0.017 & 0 \end{bmatrix} \mathbf{z}_{t-2} + \hat{\mathbf{a}}_t。$$

其中 $\mathbf{z}_t = (z_{1t}, z_{2t}, z_{3t})'$ ， z_{1t} 為 FPS 在 lag 為 t 時的數值， z_{2t} 為 CPU 在 lag 為 t 時的數值， z_{3t} 為 GPU 在 lag 為 t 時的數值， $\hat{\mathbf{a}}_t$ 為多維度的殘差，係數為 0 部分為移除的不顯著係數。VAR(2) 模型殘差的 CCF 檢定與多變量 Ljung-Box 檢定如圖 5 所示，由 CCF 圖可以發現殘差在大多數

的 lag 之下存在交叉相關性；多變量 Ljung-Box 檢定除了在 lag 為 1 之外，其他也都在顯著水準 0.05 以下，因此此模型在殘差上的表現不佳。

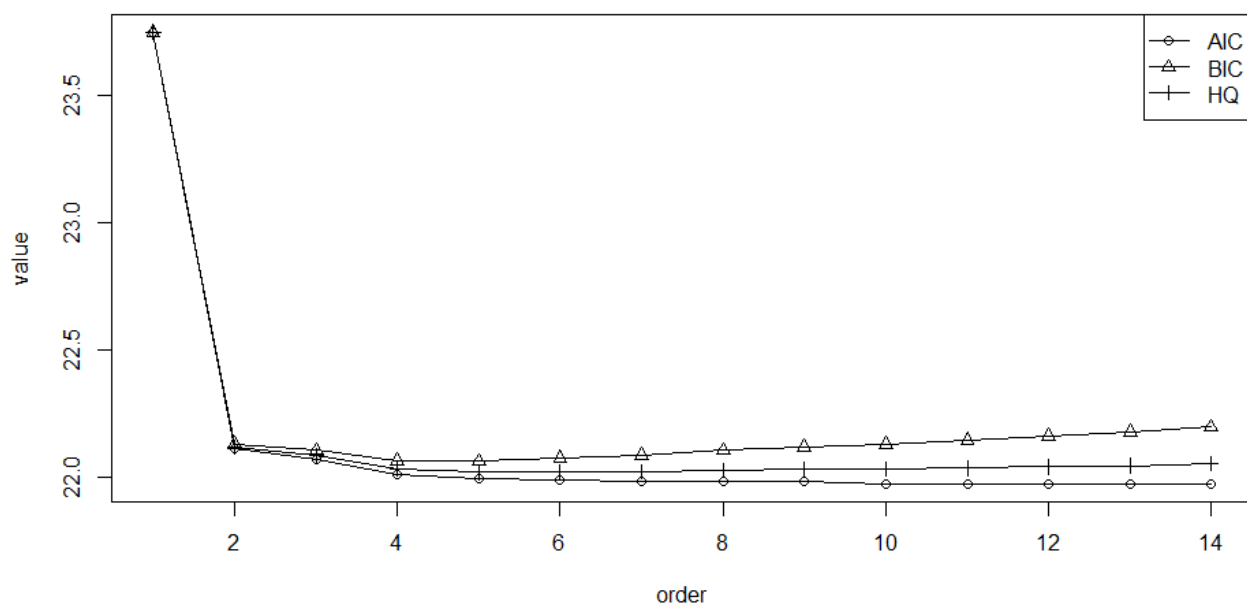
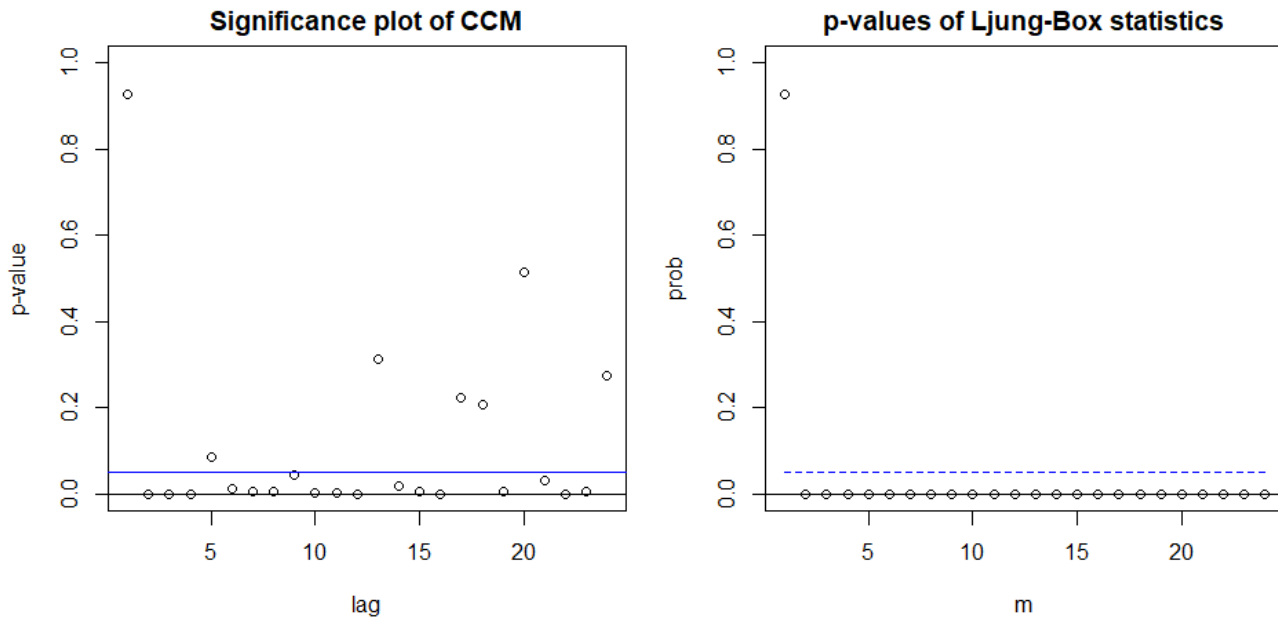


圖 4: AIC、BIC 與 HQ 在不同 VAR 模型次數下的變動



(a) CCF 檢定

(b) 多變量 Ljung-Box 檢定

圖 5: VAR(2) 模型的殘差檢定

接著考慮 VAR(4) 模型，移除部分不顯著係數的模型配適結果為

$$\mathbf{z}_t = \begin{bmatrix} -17.24 \\ 1047. \\ 1291. \end{bmatrix} + \begin{bmatrix} 0.627 & 0 & 0.007 \\ -0.686 & 0.581 & -0.249 \\ 0.137 & 0.038 & 0.202 \end{bmatrix} \mathbf{z}_{t-1} + \begin{bmatrix} 0.056 & 0 & 0 \\ 0.683 & 0 & 0.064 \\ 0.209 & 0.041 & -0.018 \end{bmatrix} \mathbf{z}_{t-2} \\ + \begin{bmatrix} 0.042 & 0 & 0.007 \\ 0.481 & 0.140 & -0.074 \\ 0.000 & -0.035 & 0.055 \end{bmatrix} \mathbf{z}_{t-3} + \begin{bmatrix} 0.129 & 0.003 & 0 \\ -0.274 & 0.079 & 0 \\ -0.092 & 0 & 0 \end{bmatrix} \mathbf{z}_{t-4} + \hat{\mathbf{a}}_t。$$

VAR(4) 模型殘差的 CCF 檢定與多變量 Ljung-Box 檢定如圖 6 所示，由 CCF 圖可以發現殘差在大多數的 lag 之下已不存在交叉相關性；多變量 Ljung-Box 檢定的表現也有所改善，在 lag 為 11 以前都在顯著水準 0.05 以上，在較大的 lag 之下則沒有，因此此模型在殘差在較低的 lag 上表現尚可，在較大的 lag 上的表現不佳。

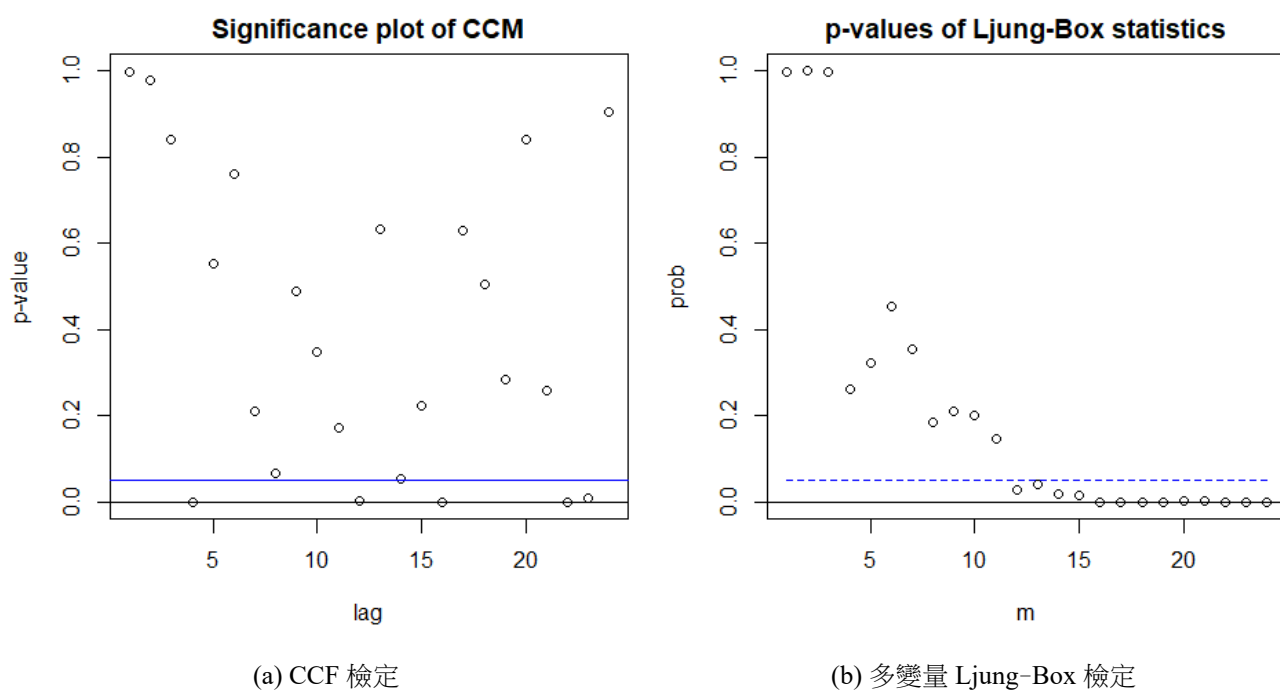


圖 6: VAR(4) 模型的殘差檢定

三、VARMA 模型

接著資料分析考慮較為複雜的 VARMA 模型，根據 Tsay (2013) 對 VARMA 模型次數的選擇，可使用延伸交叉相關性矩陣 (Extended Cross-Correlation Matrices，簡稱 ECCM) 的 p 值是否大於顯著水準 0.05 來判斷。表 1 ECCM 在不同 AR 與 MA 係數之下的 p 值，VARMA(3, 1)、VARMA(2, 2) 與 VARMA(1, 3) 為可能合適的候選模型，但由於 VARMA(3, 1) 在計算反矩陣時無法得到唯一解 (singular)，因此以下僅列出 VARMA(2, 2) 與 VARMA(1, 3) 配適的結果。

移除部分不顯著係數的 VARMA(2, 2) 模型配適結果為

$$\mathbf{z}_t = \begin{bmatrix} -37.50 \\ 1195. \\ 807.2 \end{bmatrix} + \begin{bmatrix} 1.182 & 0 & -0.008 \\ 0.114 & 0.901 & -0.585 \\ 1.553 & 0.107 & 0.511 \end{bmatrix} \mathbf{z}_{t-1} + \begin{bmatrix} -0.234 & 0 & 0.031 \\ 0.339 & 0.064 & -0.018 \\ -1.245 & -0.088 & 0.016 \end{bmatrix} \mathbf{z}_{t-2} + \mathbf{a}_t$$

$$- \begin{bmatrix} 0.574 & -0.003 & -0.013 \\ 0.966 & 0.350 & -0.348 \\ 1.467 & 0.068 & 0.310 \end{bmatrix} \mathbf{a}_{t-1} - \begin{bmatrix} 0.075 & 0.001 & 0.032 \\ 0.049 & 0.266 & -0.251 \\ -0.567 & -0.078 & 0.091 \end{bmatrix} \mathbf{a}_{t-2} \circ$$

VARMA(2, 2) 模型殘差診斷如圖 7 所示，由 CCF 圖可以發現殘差在大多數的 lag 之下並沒有

表 1: 延伸交叉相關性矩陣 p 值表

AR	MA Order: q								
p	0	1	2	3	4	5	6	7	8
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.0010	0.1402	0.1912	0.1533	0.0945	0.0481	0.1667
2	0.0000	0.0000	0.2339	0.2195	0.1692	0.2975	0.7181	0.2538	0.3897
3	0.0000	0.4004	0.5659	0.1196	0.0607	0.1769	0.6706	0.4171	0.7261
4	0.0000	0.9357	0.3538	0.3818	0.5777	0.9311	0.9453	0.9244	0.9970
5	0.0000	0.9638	0.7989	0.8947	0.9423	0.9840	0.9963	0.9999	0.9979
6	0.0000	0.9940	1.0000	1.0000	0.9953	1.0000	1.0000	0.9998	0.9993
7	0.0000	0.9994	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
8	0.0000	0.9995	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9926

顯著的交叉相關性，但在 lag 為 1 的殘差卻顯著相關；多變量 Ljung-Box 檢定的表現並不好，在較小和較大的 lag 之下都為顯著的，因此此模型在殘差在較小與較大的 lag 上的表現不佳。

接著考慮 VARMA(1, 3) 模型，移除部分不顯著係數的模型配適結果為

$$\mathbf{z}_t = \begin{bmatrix} -40.89 \\ 1022. \\ 817.2 \end{bmatrix} + \begin{bmatrix} 0.947 & 0.001 & 0.023 \\ 0.265 & 0.969 & -0.507 \\ 0.215 & 0.020 & 0.526 \end{bmatrix} \mathbf{z}_{t-1} + \hat{\mathbf{a}}_t - \begin{bmatrix} 0.339 & 0 & 0.018 \\ 1.115 & 0.419 & -0.270 \\ 0.129 & -0.018 & 0.324 \end{bmatrix} \hat{\mathbf{a}}_{t-1} \\ - \begin{bmatrix} 0.166 & 0.001 & 0.004 \\ -0.251 & 0.242 & -0.232 \\ -0.061 & -0.038 & 0.091 \end{bmatrix} \hat{\mathbf{a}}_{t-2} - \begin{bmatrix} 0.091 & 0.001 & -0.004 \\ -0.543 & -0.010 & 0.005 \\ 0.081 & 0.027 & -0.032 \end{bmatrix} \hat{\mathbf{a}}_{t-3} .$$

VARMA(1, 3) 模型殘差診斷如圖 8 所示，由 CCF 圖可以發現殘差在大多數的 lag 之下並沒有顯著的交叉相關性，且在 lag 為 7 以前的殘差沒有顯著相關；多變量 Ljung-Box 檢定在 lag 為 20 以前皆不顯著，因此此模型整體殘差診斷的表現上較前三個候選模型佳。

接著我們深入探討 VARMA(1, 3) 模型之下變數之間的關係，這邊不去直接看模型係數，而是繪製三變數的脈衝反映函數圖 (Impulse response functions, 簡稱 IRF)，如圖 9，IRF 為變

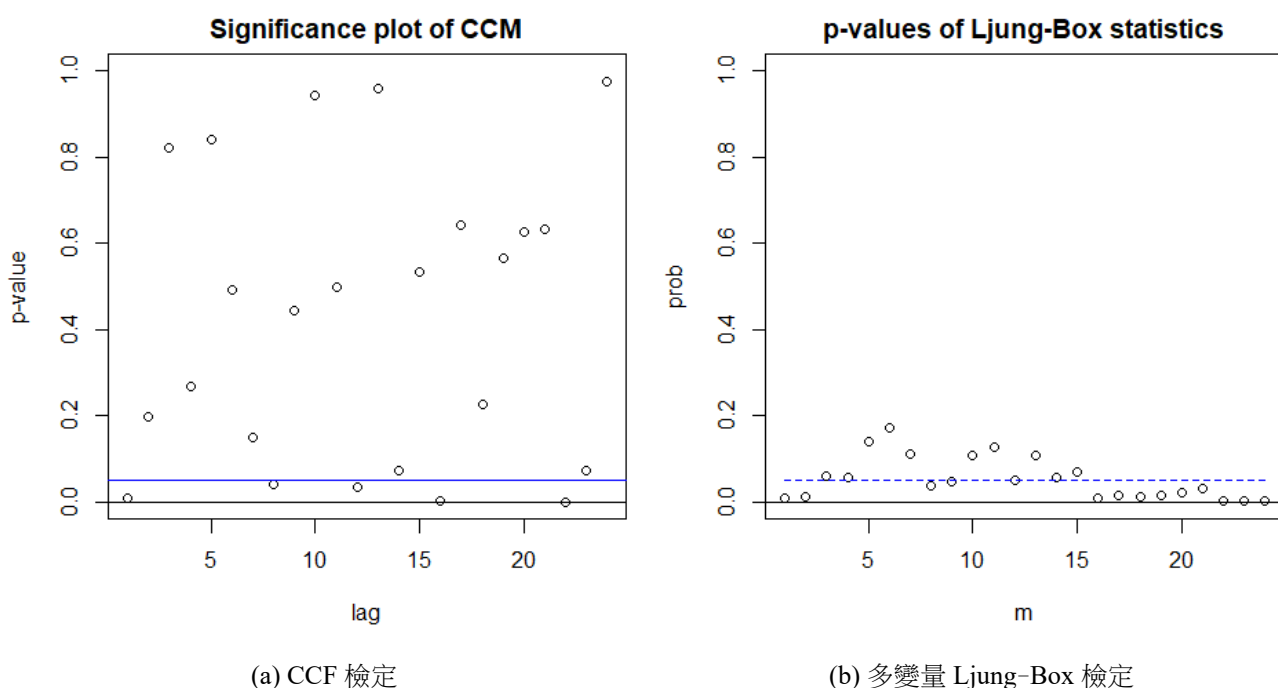


圖 7: VARMA(2, 2) 模型的殘差檢定

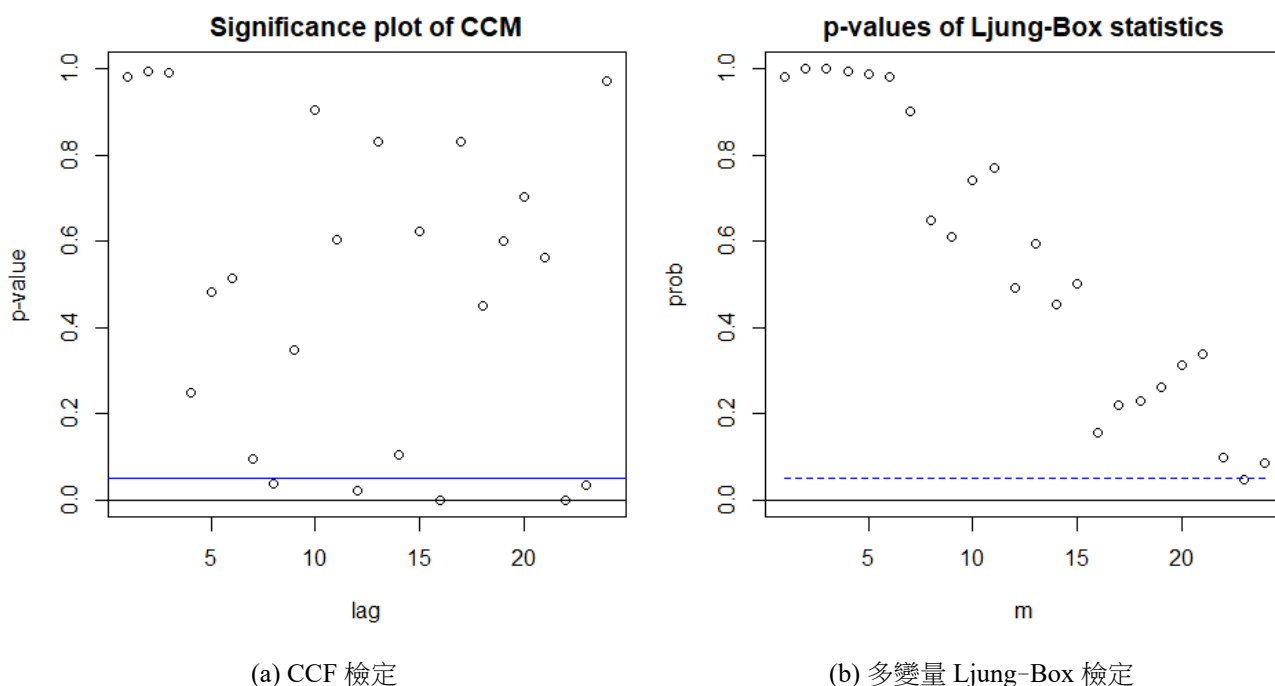


圖 8: VARMA(1, 3) 模型的殘差檢定

數與不同 lag 之下變數之間的作用關係，說明的部分可分為斜對角與非斜對角部分。斜對角部分可發現 FPS、CPU 和 GPU 各自的 lag 對當期各自本身皆是從 1 下降至 0，說明這樣的關係為定態 (stationary) 且正向動態作用。非斜對角部分，由第二列左邊的圖可發現 FPS 的 lag 對當期 CPU 的 IRF 從 0 下降至 -0.8 附近，再漸漸回升至 0，說明此關係為負向動態作用，且這樣的關係比起其餘 IRF 的變化是較明顯的，而第二列右邊的圖也可發現 GPU 的 lag 對當期 CPU 為負向動態作用；第三列的圖說明 FPS 與 CPU 的 lag 對當期 GPU 為正向動態作用，而且此作用從 lag 為 1 持續到 lag 為 2，但中間的圖變化較不明顯；第一列的圖則沒有太明顯的變動，說明 CPU 與 GPU 的 lag 對當期 FPS 作用不大。

VARMA(1, 3) 殘差的相關矩陣為

$$R_0 = \begin{bmatrix} 1.000 & -0.214 & -0.108 \\ -0.214 & 1.000 & 0.189 \\ -0.108 & 0.189 & 1.000 \end{bmatrix}$$

此矩陣為同一期殘差之間的相關性，換句話說為 FPS、CPU 和 GPU 在 VARMA(1, 3) 之下未考慮到的瞬間相關性 (instantaneously correlated)，可發現非對角線上元素的數值絕對值介於 0 到 0.3 之間，說明這些相關性雖然存在但屬於弱相關。因此，VARMA(1, 3) 的脈衝反應函數說明

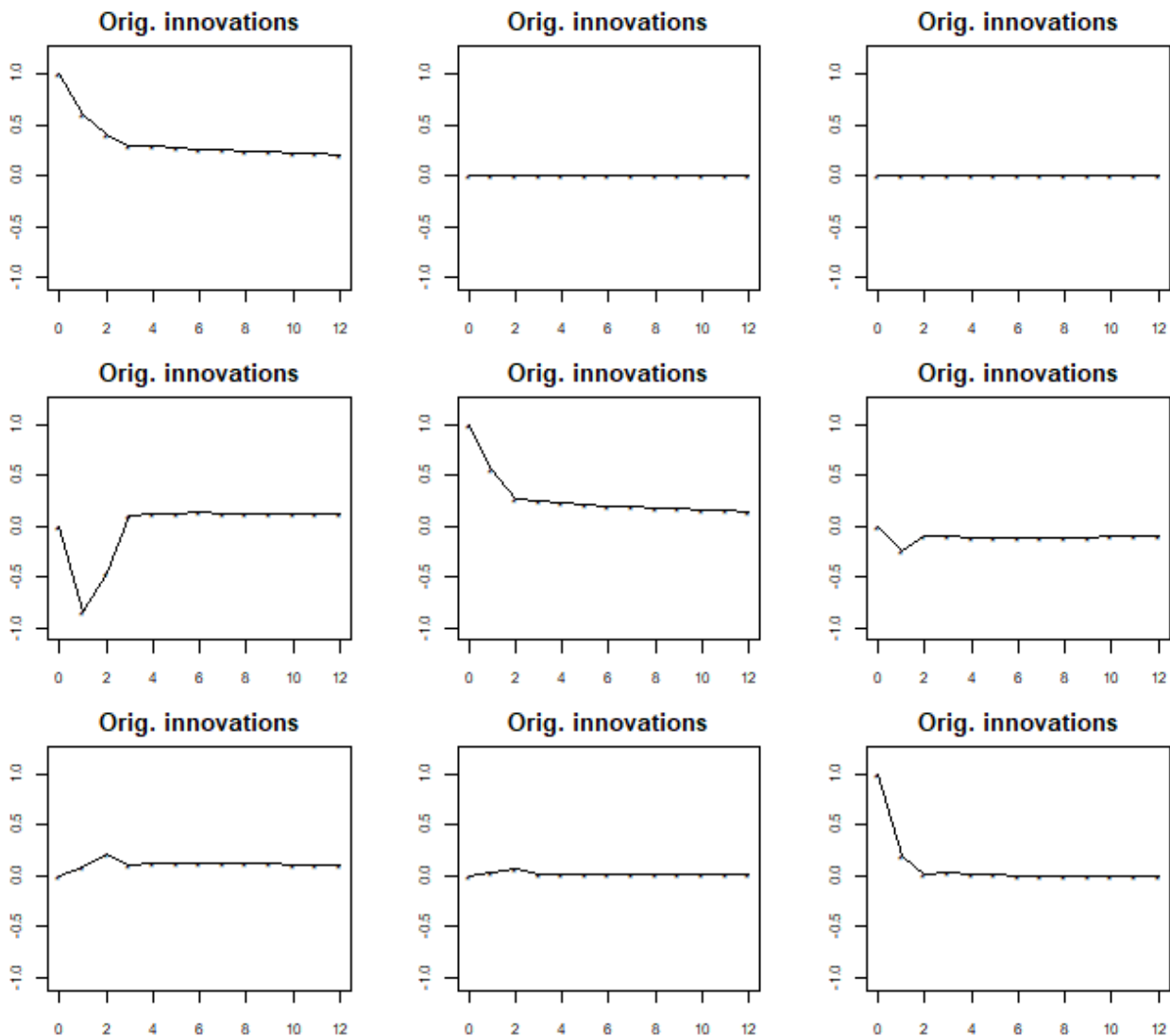


圖 9: VARMA(1, 3) 的脈衝反映函數圖

這三個變數在動態上具有相互影響之外，殘差的相關矩陣也說明其當期也會相互影響，但其影響程度不高。

第四節 預測表現

接著對殘差診斷較佳的 VARMA(1, 3) 進行預測，預測與測試資料比對的結果如圖 10，垂直虛線的左邊為部分的訓練資料，右邊的實線為測試資料，直線加原點為預測結果，兩條虛線包覆的範圍為預測值加減 2 倍預測標準誤的信賴區間。最上面的圖為 FPS 預測的結果，可以看到測試資料都有在 95% 預測信賴區間裏；中間的圖為 CPU 預測的結果，可以看到有兩個測試資料超出信賴區間，但可以明顯看到該兩點偏離整體趨勢，因此屬於離群值；下面的圖

為 GPU 預測的結果，可以看到測試資料都有在 95% 預測信賴區間裏。綜合以上，可得除了 CPU 那兩個離群值外，VARMA(1, 3) 的預測結果表現佳。

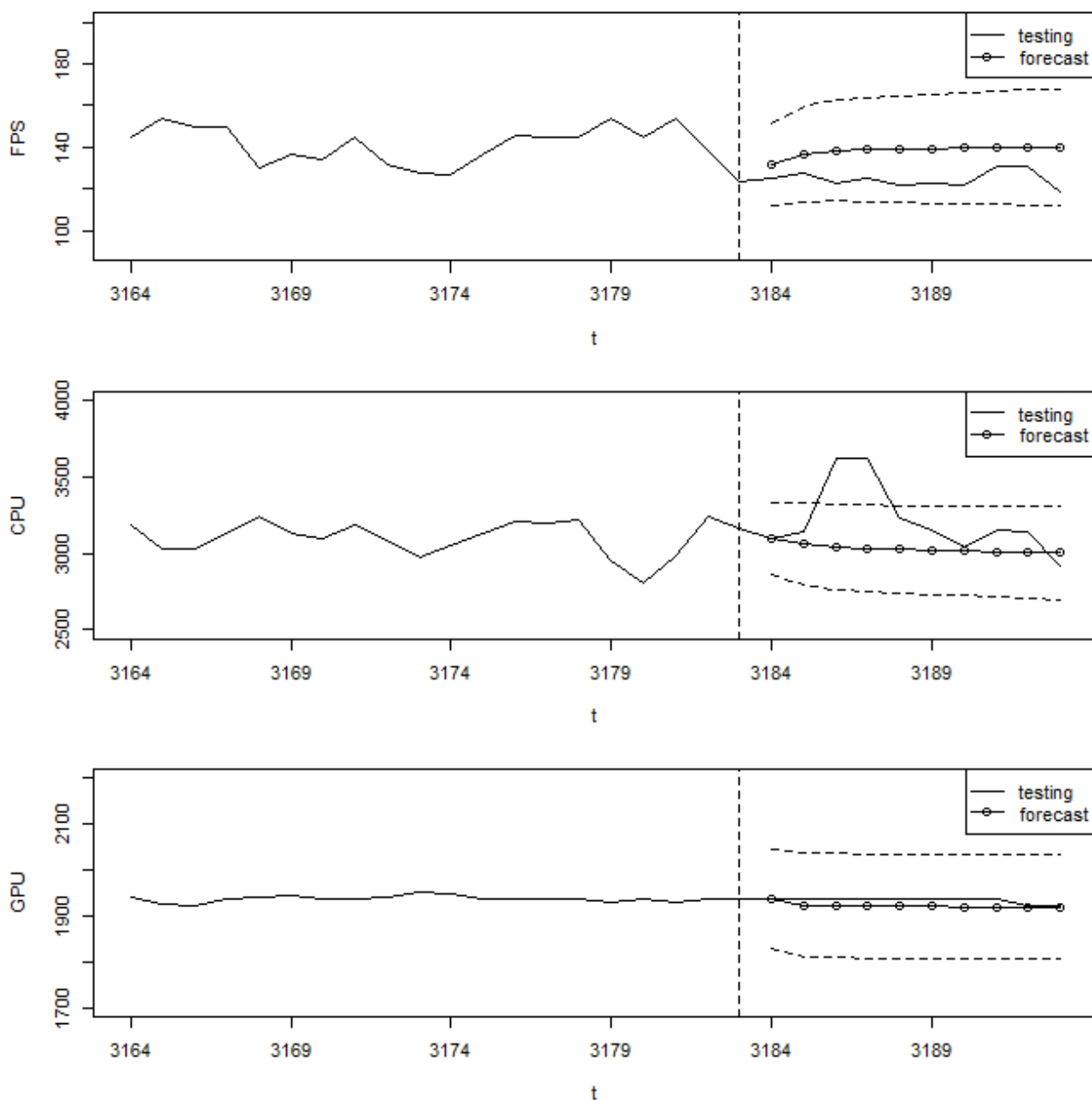


圖 10: VARMA(1, 3) 的預測表現

第五節 結論與建議

為了找出 FPS、CPU 與 GPU 相關變數之間的關係，本研究先透過初步分析，來決定模型是否該考量離群值，以及決定採用 VAR 與 VARMA 來描述變數 CCF 和 PACF 的現象；接著透

過 AIC、BIC 與 HQ 統計量判斷 VAR 的次數，且殘差診斷發現 VAR(4) 仍有些 lag 表現不佳；之後透過 ECCM 來判斷 VARMA 的次數，並透過殘差診斷發現 VARMA(1, 3) 整體表現較其他的模型佳；最後在針對 VARMA(1, 3) 進行預測能力的評估，且其表現不錯。

雖然本研究不針對離群值做處理，但實際從模型的殘差圖仍可看到離群值的存在，倘若要讓模型預測表現更佳，可能要進一步對 GPU 資料的離群值做處理或是屏除。另一方面，由 GPU 變動相對 FPS 和 CPU 不大，可推論本研究對此變數納入分析的決定並不是最好，或許僅 FPS 與 CPU 的多變量時間序列分析，就能滿足此 FPS 的相關性與預測表現，而此正是本研究主要的研究動機。因此，未來研究可進一步探討如何有效地找出與 FPS 有關的變數來進行多變量時間序列分析，但維度也不會過高。

附錄一、參考文獻

Tsay, R. S. (2013). *Multivariate time series analysis: with R and financial applications*. John Wiley & Sons.

附錄二、資料連結

https://github.com/scwu110927/Temporary-repository/blob/main/horizon_3multivariate_final.csv

附錄三、語法

```
library(MTS)

data <- read.csv("horizon_final.csv", head = T)
zt <- data[1:3183, c(14, 1, 12)]
tt <- data[3184:3193, c(14, 1, 12)]
names(zt) <- names(tt) <- c("FPS", "CPU", "GPU")

windows(7, 7); par(mar = c(5, 5, 1, 1))
MTSplot(zt)

windows(7, 6); par(mar = c(5, 5, 1, 1))
ccm(zt, lag = 20)

windows(7, 2); par(mar = c(5, 5, 1, 1), mfrow = c(1, 3))
pacf(zt[, 1], lag = 20, ylim = c(-0.1, 1))
pacf(zt[, 2], lag = 20, ylim = c(-0.1, 1))
pacf(zt[, 3], lag = 20, ylim = c(-0.1, 1))

var <- VARorder(zt)
windows(10, 5); par(mar = c(5, 5, 1, 1))
plot(var$aic, type = "o", pch = 1, xlab = "order", ylab = "value")
lines(var$bic, type = "o", pch = 2)
lines(var$hq, type = "o", pch = 3)
legend("topright", legend = c("AIC", "BIC", "HQ"), lty = 1, pch = 1:3)

m11 <- VAR(zt, 2)
m11a <- refVAR(m11, thres = 1)
windows(5, 5); par(mar = c(5, 4, 2, 1))
MTSdiag(m11a)

m12 <- VAR(zt, 4)
m12a <- refVAR(m12, thres = 1)
windows(5, 5); par(mar = c(5, 4, 2, 1))
MTSdiag(m12a)

Eccm(zt, maxp=8, maxq=8)
```

```

m11 <- VARMA(zt, p=2, q=2) #39; aic=22.10922; bic=22.18354
m11a <- refVARMA(m11, thres=0.5) #37; aic=22.11185; bic=22.18236
windows(5, 5); par(mar = c(5, 4, 2, 1))
MTSdiag(m11a)

m12 <- VARMA(zt, p=1, q=3) #39; aic=22.02045; bic=22.09476
m12a <- refVARMA(m12, thres=0.5) #38; aic=22.02015; bic= 22.09256
windows(5, 5); par(mar = c(5, 4, 2, 1))
MTSdiag(m12a)
MTSdiag(m12a)

windows(7, 6); par(mar = c(5, 5, 1, 1))
VARMAirf(m12$Phi, m12$Theta, m12$Sigma, orth = F)

windows(7, 7); par(mar = c(5, 5, 1, 1), mfrow = c(3, 1))
pred <- VARMApred(m12a, h=10)
plot(data[3164:3193, 14], type = "l", ylim = c(90, 200), ylab = "FPS" ,
      xlab = "t")
lines(x = 21:30, y = pred$pred[, 1], type = "o", lty = 1)
lines(x = 21:30, y = pred$pred[, 1]+2*pred$se[, 1], lty = 2)
lines(x = 21:30, y = pred$pred[, 1]-2*pred$se[, 1], lty = 2)
abline(v = 20, lty = 2)
legend("topright", pch = c(NA, 1), lty = 1, legend = c("testing", "forecast
"))

plot(data[3164:3193, 1], type = "l", ylim = c(2500, 4000), ylab = "CPU" ,
      xlab = "t")
lines(x = 21:30, y = pred$pred[, 2], type = "o", lty = 1)
lines(x = 21:30, y = pred$pred[, 2]+2*pred$se[, 2], lty = 2)
lines(x = 21:30, y = pred$pred[, 2]-2*pred$se[, 2], lty = 2)
abline(v = 20, lty = 2)
legend("topright", pch = c(NA, 1), lty = 1, legend = c("testing", "forecast
"))

plot(data[3164:3193, 12], type = "l", ylim = c(1700, 2200), ylab = "GPU" ,
      xlab = "t")
lines(x = 21:30, y = pred$pred[, 3], type = "o", lty = 1)
lines(x = 21:30, y = pred$pred[, 3]+2*pred$se[, 3], lty = 2)
lines(x = 21:30, y = pred$pred[, 3]-2*pred$se[, 3], lty = 2)
abline(v = 20, lty = 2)
legend("topright", pch = c(NA, 1), lty = 1, legend = c("testing", "forecast
"))

```