108cbc-經1（委託研究報告）

運用大數據之機器學習方法
預測臺灣經濟成長率

受委託單位：國立臺灣師範大學

研究主持人：何宗武 教授 (國立臺灣師範大學全球經營與策略所)

兼任研究員：葉國俊 教授 (國立臺灣大學國家發展所)

牟萬馨 副教授 (中原大學財務金融學系)

林雅淇 助理教授 (逢甲大學經濟學系)

中央銀行 委託研究

中華民國 109 年 8 月

中央銀行委託研究計畫編號
108cbc-經1

運用大數據之機器學習方法
預測臺灣經濟成長率

受委託單位：國立臺灣師範大學

研究主持人：何宗武 教授 (國立臺灣師範大學全球經營與策略所)

兼任研究員：葉國俊 教授 (國立臺灣大學國家發展所)

牟萬馨 副教授 (中原大學財務金融學系)

林雅淇 助理教授 (逢甲大學經濟學系)

研 究 期 程：中華民國108年9月至109年8月

研 究 經 費：新臺幣599,000元

中央銀行 委託研究

中華民國 109 年 8 月

..

目 錄

表目錄

圖 目 錄

1 緒論

1.1 研究目的

本計畫研究機器學習模式是否能對臺灣經濟成長率的時間序列預測，做出較佳的貢獻。 我們透過比較兩個方法對臺灣經濟成長率的預測績效，來進行優劣評估：一是機器學習方法，另一則是時間序列模型方法。前者直接分析從數據資料所萃取的特徵，並且透過訓練模式，提升了預測表現；後者無訓練機制，主要將預先設定的模型套用於數據資料，並漸進地尋找解決方案。然而對於機器學習方法是否在預測能力上領先時間序列模型，仍處於研究階段。較早的文獻如 Makridakis and Hibon (2000) 和 Morlidge (2014)，並不認為精密的機器學習方法預測績效優於傳統計量模型。但是近幾年的發展，則已使學者對於機器學習方法轉趨樂觀 (Baek and Kim, 2018; Beyca *et al*., 2019; Bolhuis and Rayner, 2020; Chakraborty and Joseph, 2017; Chatzis *et al*., 2018; Kong *et al*., 2017; Richardson and Mulder, 2018; Tiffin, 2019; Torres *et al*., 2018; Zhao *et al*., 2017a)。

近期機器學習模式，諸如支援向量機器 (support vector machine, SVM)、自動化機器學習 (automatic machine learning, autoML) 和深度學習長短期記憶 (long-short term memory, LSTM) 的循環類神經網絡 (recurrent neural network, RNN) 等聲譽鵲起。首先是被視為淺層機器學習的 SVM，該方法以最小化度量法求解參數之估計值，相較於最小平方法，不易受離群值偏離整體數據資料趨勢的影響；其次是 autoML，該方法是一個自動機器學習和大數據的公開資源解決方案，依照數據資料的時間序列結構產生大量特徵，再將這些特徵作為解釋變數進行演算，學者可在該環境下建立自動化演算法包含類神經網路 (neutral network, NN) 或

隨機森林 (random forest, RF)，自動執行機器學習工作流程，利於執行建模相關的任務；第三是 LSTM 的 RNN，該方法擅長趨勢識別 (pattern-recognition)，得以處理長期的過去訊息，將之前的網絡做連結，彌補 NN 無法克服的問題。

Chakraborty and Joseph (2017) 為英格蘭銀行所做的機器學習研究報告，可說是開風氣之先，在金融機構的資產負債表檢測示警、通貨膨脹率、以及科技創投公司的籌資模式等領域影響深遠，其結論指出 RNN 與微調的 SVM，預測績效超越時間序列模型。SVM 方法亦在預測總體經濟數據、金融危機事件和能源類消費方面有卓越的表現。部分學者則認可深度學習的 autoML 架構在預測電力消費或經濟成長率的表現，因為該方法預測誤差較線性模型低。也有一些研究將 RF 應用在預測經濟成長率、美國消費者物價指數通貨膨脹率、或是金融資產的風險溢酬。至於 LSTM 方法，部分研究亦非常肯定 LSTM 方法應用在住宅道路、交通及股市指數預測方面的表現。上述成就將在第 2 章文獻回顧中詳述，也是激發我們以臺灣經濟成長率為題進行嘗試的主因。

過去文獻偏重以計量方法預測臺灣經濟成長率，以機器學習方法進行者尚付之闕如，本研究欲彌補這部分之不足，因此我們將採用 SVM、autoML 和 LSTM 等方法，也特別將注意力放在 LSTM。至於相應的計量研究方法，本文依據 Hyndman *et al*. (2018) 時間序列預測模型集合 (pool) 挑選以下模型，這些曾受學界肯定的方法包括：(1) Autoregressive Integrated Moving Average (ARIMA. Ediger and Akar, 2007) ，(2) Self-Exciting Threshold Autoregressive (SETAR. Feng and Liu, 2003)，(3) logistic STAR (LSTAR. Stock and Watson, 1998)，(4) Generalized Additive Models (GAMs. Serinaldi, 2011)，(5) Bootstrapping aggregation (bagged. Zhao *et al*., 2017b)，以及 (6) Neural network (NN. 常被運用於匯率預測)。就評估預測績效而言，本文使用多種模型設定，包括趨勢模型、線性模型、非線性模型、加入固定趨勢的線性模型、加入季節性虛擬變數的線性模型、以及機器學習模型等等。

如圖一所示，本文採用支出面以 2016 年為基期的實質 GDP 成長率，樣本期間為 1962 年第一季至 2019 年第四季，並將之分成兩個子樣本期間，一個是 1962 年第一季至 2016 年第四季，以及 2017 年第一季至 2019 年第四季，分別以時間序列模型和機器學習方法進行預測。至於預測方法，本文執行單步 (one-step) 以及多步預測 (multi-step prediction)。雖然多數時間序列模型使用最大概似估計法 (maximum likelihood estimation) 進行單步預測之表現優異，尤其是樣本內數據 (in-sample data，亦即訓練數據 training data)；但實務上樣本外數據 (out-of-sample data，亦即測試數據 test data) 的預測必須是多步預測。由於臺灣經濟成長率為季資料，單步預測僅在低頻樣本內歷史數據有良好績效。不同於單步預測，多步預測使得政策制定者得以展望未來且評估政策。如文獻所述 (Hamzacebi *et al*., 2009; Kline, 2004)，遞迴方法 (recursive) 是最具直覺性的預測方法，即先以模型預測下一期的值，依據該期的值再預測下下一期的值，再使用解釋變數真實值的落後期，從預測樣本的第一期，計算樣本外預測 (如 Saad *et al*., 1998)。再者，面對季資料樣本數不足的問題，前述 Charkraborty and Joseph (2017) 以 60 季開始，用逐次增 1 的方法遞迴更新 (recursively update)，樣本外只採用 8 季。因此在執行多步預測方面，本研究亦使用遞迴和直接 (direct) 預測方法。

雖然部分研究認可機器學習方法應用在時間序列預測的績效，但本文結論則與上述文獻不盡相同，茲將本研究實證結果重點臚列如下：

首先，就單步預測而言，統計時間序列方法，尤其是 ARIMA 和 bagged 的模型表現很好，傳統時間序列與機器學習方法在此處不分上下。

其次，採行多步動態遞迴預測評估，以樣本外預測績效而言，H_2O 表現最佳。以整體時間序列表現來說，機器學習中 SVM 的 4 種設定優於 H_2O 和以 RNN 深

度學習的 LSTM 模式。因此，在多步動態遞迴預測方面，機器學習方法優於時間序列方法。

其三，在混合落後結構和添加其他變數時，三個機器學習模型均有相當優異的預測績效，且以 autoML 為最佳。而在整體時間序列結果上， LSTM 難以有效捕捉樣本內高峰或低谷，autoML 在某些情形下表現良好，SVM 則相對有穩定良好表現。

最後，以機器學習方法改善時間序列模型預測雖屬可行，但仍有改善空間。本研究只是嘗試設計一個作法，初步判讀其可行性。若就資訊科學的角度思考，只要硬體演算能力夠強大，用 autoML 對資料直接進行大規模計算，找出最佳資料結構，再使用預測平均法，應是改善時間序列預測更為有效的方式。

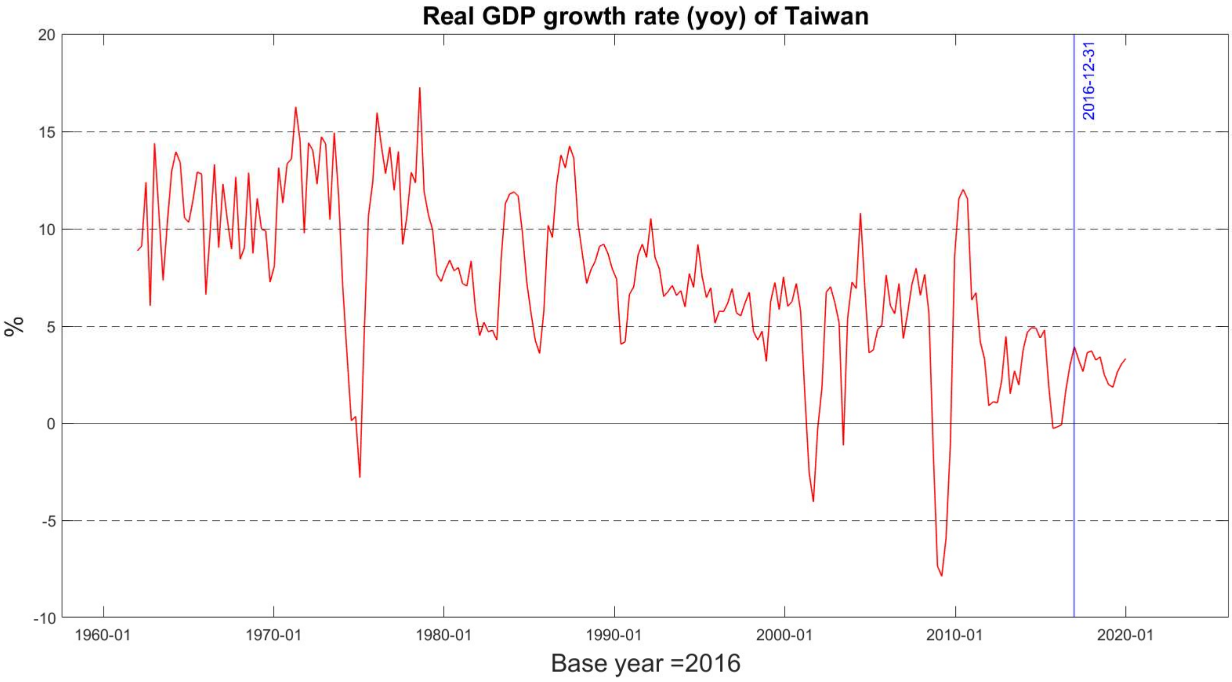
本研究將對這些問題作系統性的呈現，然後提出預測模式的建議。

1.2 研究流程

本計畫分為兩大部分，一為相關資料整理；二是分別以傳統時間序列分析和機器學習分析，預測我國經濟成長率。本文使用多個模型設定，考量包括趨勢模型、線性模型、非線性模型、包含固定趨勢的線性 (機器學習) 模型，與包含季節性虛擬變數的線性 (機器學習) 模型。根據既定研究計畫，完成以下工作項目：

1. 綜整臺灣經濟成長率預測相關文獻，並對模型設定進行討論。
2. 整理臺灣經濟成長率季資料。
3. 建立多個模型設定，據以預測臺灣經濟成長率。
4. 執行樣本單步和多步預測與績效評估。

5. 建立機器學習對臺灣經濟成長率的預測模型。



資料來源：中華民國統計資訊網。

圖一、臺灣經濟成長率時間序列

2 文獻回顧

2.1 預測臺灣經濟成長率之文獻回顧

GDP 與 GDP 成長率是最常用的經濟發展衡量指標，一個合適的預測模型對於政策制定將有極大助益。林向愷等 (1998) 和 Huang (1999) 曾使用單變量馬可夫轉換模型 (Markov switching model) 預測臺灣景氣循環動態，Chen and Lin (2000) 則以多變量馬可夫轉換模型研究相關議題。即使這些模型在預測方面表現良好，但未將其他總體資料數據對提升預測力的可能性納入考量。徐士勛等 (2005) 以 Stock and Watson (2002) 為基礎，建立考量商品市場、貨幣市場及勞動市場等變數的擴散指標預測模型，顯示這些市場訊息得以掌握總體面之表現，進而改善預測能力。

陳宜廷等 (2011) 在模型設置時納入了消費、貨幣和國外部門的歷史訊息，發現可以改善主計處發布的一季前臺灣經濟成長率之預測能力，惟仍未能解決景氣循環時對訊息過度反應而產生的預測誤差。葉錦徽等 (2012) 應用 Huang *et al.* (1998) 的經驗解構法 (empirical mode decomposition，EMD) 預測臺灣經濟成長率，解構序列並將所有成分內涵置於模型進行預測，結果呈現良好的多步預測績效。吳易樺等 (2014) 使用要素模型進行預測，並比較該模型與時間序列線性模型的預測力，以主成分分析法 (principal component analysis) 挑選用來預測 GDP 的要素，結果顯現該模型多步預測力優於時間序列模型。

2.2 時間序列的機器學習方法應用

近幾年，機器學習模型大量應用於時間序列預測，在總體變數預測方面，Ahmed *et al.* (2010) 提出多層感知機 (multilayer perceptron)，貝式類神經網絡 (Bayes neural networks)，徑向基函數 (radial basis functions)，一般化迴歸類神經網絡 (generalized regression neural networks，GRNN)，K-近鄰演算法 (K-nearest neighbor，KNN)，分類和迴歸決策樹 (CART regression trees)，使用 SVM 作迴歸模型 (support vector regression，SVR)，高斯過程 (Gaussian processes) 等八個機器學習方法，預測低頻的貨幣供給 M3，樣本為約一千筆的 M3 月資料，該研究結果顯示，多層感知機與高斯過程的預測力是最好的。在研究通貨膨脹率方面，Medeiros *et al*. (2019) 和 Ülke *et al*. (2018) 以月頻率數據資料，分別為美國消費者物價指數通貨膨脹率和核心個人消費支出通貨膨脹率建立預測模型，並比較機器學習方法及時間序列模型，前者使用機器學習的隨機森林，後者使用的機器學習方法包括 KNN、人工類神經網絡 (artificial neural network，ANN) 和 SVR，兩篇論文實證結果皆顯示機器學習方法之預測準確性較高。在預測失業率方面，由於該資料具有週期性，傳統時間序列模型不易捕捉該資料之轉折點，因此預測表現差，Hall (2018) 使用機器學習的彈性網路 (Elastic Net) 方法預測美國的失業率，發現該方法可以辨識數據資料之轉折點，相較於時間序列模型不僅降低預測之誤差，亦提升預測之準確性。在預測土耳其伊斯坦堡的天然氣消費方面，Beyca *et al*. (2019) 的論文以複回歸、ANN 和 SVM 等方法建立模型，結果指出機器學習的 SVM 方法之預測誤差最小且準確性最高。

在預測 GDP 表現和經濟成長率方面，Gogas *et al*. (2015) 以 1976Q3-2007Q2 的美國殖利率曲線和實質 GDP 資料數據，預測 2007Q3-2011Q4 美國經濟衰退 (以實質 GDP 偏離長期趨勢衡量)，結果顯示相較於多元概率比回歸 (probit) 和

對數機率 (logit) 模型，使用 SVM 方法之預測正確性更高。Feng and Zhang (2014) 使用 ANN 和灰預測 GM (1,1) 等方法，[1]以 1971-1994 中國浙江省第三產業 (tertiary sector) 對該省整體經濟成長的貢獻比率作為訓練樣本，預測該數據資料在 1995-1997 期間每年之表現，研究結果呈現 ANN 之預測相較於 GM (1,1) 更接近真實數據而且預測誤差較小。Tiffin (2016) 預測黎巴嫩之經濟成長率，該國之資料有結構變動的特質且官方機構發布數據有時間落差，因此對研究者而言預測該國 GDP 非常困難。他以 1996-2010 期間月頻率總體數據資料，為黎巴嫩 2011-2014 的季經濟成長率進行預測，研究指出機器學習方法的 Elastic net regression 和 RF 不僅克服以上問題，[2]也提升了預測績效，使得政策制定者得以檢視目前的政策方針。

近兩年學者使用機器學習模型預測經濟成長率亦有突破，例如 Richardson and Mulder (2018) 使用 550 個國內外總體指標，使用 1995Q1-2008Q4 之數據為紐西蘭 2009Q1-2018Q1 之經濟成長率執行即時預測 (nowcasting)，發現機器學習方法如 SVM 和 NN，比傳統時間序列模型的自我迴歸、因子模型 (factor model)

---

[1] 灰預測 GM(1,1)主要功能在系統模型訊息不完全、影響變數紛雜、行為模式不確定的狀況下，進行系統的關聯分析、模型建構及預測。主要優點為所需的數據不用太多且數學基礎簡單，但缺點是模型參數採最小平方法推估，在系統受干擾時會產生偏誤估計。

[2] Elastic Net regression 方法結合了 Ridge Penalty 及 Lasso Penalty 之優勢，前者達到有效正規化優勢 (亦即降低模型過度配適情形和樣本外誤差)，後者則是進行變數挑選優勢。RF 是使用多個決策樹組合建立而成的總和預測演算法，決策樹將分類數據資料分成兩類，此分類程序會重複直到將預測變數之誤差降到最低。在訓練樣本內，RF 採用 bagged 方法將個別決策樹建立在觀察值的單一隨機樣本，在學習過程的每次候選分裂中選擇隨機子樣本集合，產生大量無相關的決策樹，從而大幅降低預測誤差。

及貝式向量自我回歸模型 (Bayesian vector autoregression，BVAR)，在誤差表現方面皆有明顯改善。Diebold and Shin (2019) 提出 partially-egalitarian 最小絕對收斂和選擇算子 (least absolute shrinkage and selection operator，peLASSO) 程序，該程序從多個因子中篩選出最適合預測因子，再為最後僅存之因子配置相同權重進行合併，以 1999Q3-2016Q4 歐洲中央銀行數據資料，預測歐元區樣本外一年的經濟成長率，作者指出 peLASSO 之預測績效優於簡單因子平均合併 (simple average combination) 的方法，而從向前滾動 (rolling forward) 找出的最佳績效平均 (best average) 方法亦在預測表現勝於 simple average combination。Bolhuis and Rayner (2020) 有鑑於土耳其經濟成長率波動幅度過大，傳統計量模型不易預測，以機器學習的 RF、梯度提升決策樹 (Gradient boosted trees，GBT)[3]、SVM 方法、以及傳統計量模型的 OLS，以約 200 個因子為 2012-2019 期間之土耳其經濟成長率進行 nowcasting，實證顯示相較於計量模型，機器學習方法不僅預測誤差減少許多，亦能在捕捉經濟成長率波動性上有所改善。Cepni *et al*. (2020) 使用傳統時間序列 AR 方法和多個 LASSO 方法，樣本期間自 2003 年 1 月至 2018 年 6 月，以超過 80 個國內外經濟政策不確定性和數據非預期因子 (data surprise factors) 建立模型，並為巴西、印尼、墨西哥、南非及土耳其等國，進行 2008 年 7 月至 2018 年 6 月期間經濟成長率的遞迴擬真 (pseudo) 樣本外預測，包括樣本外一期、樣本外兩期、以及 nowcasting，結果指出包含全球因子的貝氏變異 LASSO 方法預測績效，領先傳統時間序列及其他 LASSO 方法。

[3] GBT 是從決策樹弱預測模型學習，進而建立更穩健的模型，因此產生弱預設模型集合，提升預測績效。

除了總體經濟變數，學者亦使用機器學習模型預測金融變數。由於股價高頻資料數據本身具有跳躍式或噪音等問題，導致時間序列模型預測力惡化，序列出現轉折點時亦然。在日資料方面，Patel *et al.* (2015) 以印度 S&P CNX Nifty 指數和孟買敏感指數 (S&P Bombay Stock Exchange Sensitive Index) 十年數據做為樣本，將第一階段的 SVR 模型結果分別帶入第二階段的 ANN，RF 和 SVR 等模型，結果顯示 SVR 與 ANN 預測力最佳，Wei (2016) 提出混合時間序列的適應性類神經模糊推論系統 (adaptive network-based fuzzy inference system, ANFIS)，用來預測臺灣加權股價指數與香港恆生股價指數，實證顯示該模型預測力優於 AR 模型和 SVR 模型。Baek and Kim (2018) 建立一個結合 LSTM 預測模組 (prediction LSTM module) 和 LSTM 預防過度配適模組 (overfitting prevention LSTM module) 的 ModAugNet 模型，為美國的 S&P500 指數和南韓的 KOSPI200 指數進行預測，與只有 LSTM 預測模組的 SingleNet 模型相較，ModAugNet 模型降低預測誤差且改進預測績效。至於在低頻的股價預測方面，Hafezi *et al.* (2015) 提出以 bat-neural network multi-agent system (BNNMAS)、遺傳演算法類神經網絡 (genetic algorithm neural network， GANN)、GRNN 等方法，使用德國股價指數 DAX 季資料預測未來八年走勢，比較上述方法在預測 2007-08 年全球金融危機下 DAX 的績效後，認為 BNNMAS 是正確性最高最可靠的模型。Gu *et al*. (2020) 以月頻率動能指標 (momentum)、流動性和波動性當作預測值，用來預測股市指數和個別股票的風險溢酬，使用傳統計量的最小平方法以及機器學習的 RF 和 NN 方法，實證指出以 RF 和 NN 方法可帶來最佳的預測績效。在預測美國股票溢酬方面，Iworiso and Vrontos (2020) 使用二元 probit、分類和迴歸樹以及懲罰的二元 probit (結合 LASSO、脊 ridge 或彈性網的二元 probit) 等方法，實證顯示懲罰的二元 probit 預測正確性優於二元 probit 方法。

機器學習方法近年來亦應用在其他金融指數、資產和股市崩盤事件，De Stefani *et al*. (2017) 採用機器學習模型預測恐慌指數，而此前曾有文獻使用非線性 AR (nonlinear AR，NAR) 模型預測單一指數，作者另提出外部輸入非線性自動 (nonlinear AR with exogenous input，NARX) 迴歸方法，亦即結合兩個指數的訊息後預測其中一個指數，同時兼採 ANN，KNN 和 SVR 等方法，結果顯示在 NARX 架構下，另一個指數的補充訊息有助於降低 NAR 架構下的預測誤差，進而提升指數之預測力。Chatzis *et al*. (2018) 建立一個含括 39 個國家之股市、債券、和匯率的架構，以金融資產的日報酬率和波動性預測股市崩盤，使用 RF、SVM、NN 和深度學習前饋網絡 (deep learning feedforward network) 等機器學習方法，實證指出深度學習前饋網絡方法在樣本外(out-of-sample) 預測績效領先其他機器學習方法，該模型不僅提升金融危機分類正確性，同時也為政策制定者提供一個為全球金融市場預警金融危機的參考工具。Risse (2019) 以消費者物價指數、匯率、政府公債殖利率、期間利差、S&P500 指數、GSCI 商品指數、和已實現波動性等作為預測因子，結合不連續小波轉換 (discrete wavelet transformation) 和 SVR 的方法，預測黃金價格動態，作者將預測因子分類為經濟變數和財金變數，發現經濟變數因子預測準確性較高。

附帶一提，近期 Montero-Manso *et al*. (2020) 與 Smyl (2020) 分別提出解決方案：前者是一種混和解決模式 (hybrid approach)，結合多個統計方法 (如 Naïve、隨機漫步、theta 方法、自動化 ARIMA 演算法、自動化指數平滑演算法、BATS、以局部加權散步平滑 Loess 將季節性趨勢性分解的 AR 模型、以及 RR 等等) 及一個機器學習方法；後者則是混和指數平滑方法和 LSTM 的 RNN 方法。相較於單一機器學習如自動化 ANN 方法，前述二個解決方案不論在區間預測 (prediction interval) 還是在點預測 (point forecast)，皆有所改善 (Gilliland, 2020)。

正如 Makridakis et al. (2020) 所言，精細的研究方法即使增加了運算時間，卻得以更好地捕捉時間序列趨勢，提升預測力並使預測績效有所進步。

3 時間序列預測工具箱簡介：統計方法與機器學習

令 y_t 為經濟成長率的時間序列，如式 (1) 為簡單帶有漂移項 (drift) 的隨機漫步模型 (random walk model)：

$$y_t = \bar{c} + y_{t-1} + \varepsilon_t , \tag{1}$$

其中 ε_t 呈常態分配。季節性的隨機漫步模型如式 (2)：

$$y_t = \bar{c} + y_{t-m} + \varepsilon_t , \tag{2}$$

其中 m 表示季節性的落後項。

h-期預測則為 $y_{t+h} = \bar{c} \cdot h + y_t$ 。

3.1 統計方法

(1) 差分整合移動平均自我迴歸模型 (autoregressive integrated moving average model, ARIMA)

ARIMA 模型依據自我相關，當前預期價格與該前期價格是線性相關的，ARIMA 模型的限制在於選擇 AR 項數和 MA 項數的適當值時，需要仔細觀察 ACF 和 PACF 圖，也需要更深入理解 ARIMA 模型在統計學實作的概念。只要滿足了選擇 AR 項數和 MA 項數適當值的條件，ARIMA 可視為一個良好的模型，詳見 Tsay (2010) 第二章的說明。

我們採用 AIC 的自動選擇程序，以最大落後階次為 5，最大差分階次為 2 的條件，搜尋最適 ARIMA 結構，標示符號為 auto.arima。在這個架構之下，我們也額外添加外生變數：確定趨勢 (Trend)、季節虛擬變數 (sDummy) 和捕捉頻率無關的週期傅立業 (Fourier)。週期傅立業項取 2 做傅立業週期函數展開。

另外，我們也添加分量自我迴歸 (Quantile AutoRegression, QAR)，採取 Koenker and Xiao (2006) 的作法，標示符號為 QAR(p)。

(2) 非線性移轉模型：SETAR 和 LSTAR

我們使用兩個移轉模型：SETAR 和 LSTAR，前者是自我激勵門檻自我回歸模型 (self-exciting threshold autoregression) 的簡稱，例如 SETAR(p) 模型如式 (3)：

$$y_t = \begin{cases} \mu_1 + \rho_{1,1} y_{t-1} + \cdots + \rho_{1,p1} y_{t-p1} + \varepsilon_t & \text{if} \quad x_{t-d} \le \theta_{m-1} \\ \mu_2 + \rho_{2,1} y_{t-1} + \cdots + \rho_{2,p2} y_{t-p2} + \varepsilon_t & \text{if} \quad \theta_{m-1} < x_{t-d} \le \theta_{m-2} \\ \vdots & \\ \mu_m + \rho_{m,1} y_{t-1} + \cdots + \rho_{m,pm} y_{t-pm} + \varepsilon_t & \text{if} \quad \theta_1 < x_{t-d} \end{cases}, \tag{3}$$

其中參數如次：

m: 狀態的數量

$\mu_1 \ldots \mu_m$: 每個狀態的截距

$p_{j,1} \ldots p_{j,m-1}$: 在狀態 j 的落後期數量

$\theta_1 \ldots \theta_{m-1}$: 門檻值

d: 移轉變數的延遲項數

x_{t-d}: 門檻 (移轉) 變數

該模型稱為 SETAR 在於 $x_{t-d} = y_{t-d}$, 亦即 AR 項本身是門檻變數。我們將理論上無限的狀態數量限制為 2 或 3，如高、中、低等三個狀態，因此該模型可套用如式 (4) 的簡易型式：

$$y_t = \begin{cases} \mu_L + \rho_{L,1} y_{t-1} + \cdots + \rho_{L,pL} y_{t-pL} + \varepsilon_t & \text{if} \quad \theta_L \le x_{t-d} \\ \mu_M + \rho_{M,1} y_{t-1} + \cdots + \rho_{M,pM} y_{t-pM} + \varepsilon_t & \text{if} \quad \theta_L < x_{t-d} \le \theta_H \\ \mu_H + \rho_{H,1} y_{t-1} + \cdots + \rho_{H,pH} y_{t-pH} + \varepsilon_t & \text{if} \quad \theta_H < x_{t-d} \end{cases}, \tag{4}$$

LSTAR 是羅吉斯平滑門檻自我回歸模型 (logistic smoothing threshold autoregression)，納入一個羅吉斯平滑 AR(p) 建置

$$\begin{aligned} y_t = & (\mu_L + \phi_{L,1} y_{t-1} + \cdots + \phi_{L,p} y_{t-p} + \varepsilon_t) \cdot G(z_t, m, \gamma) \\ & + (\mu_H + \phi_{H,1} y_{t-1} + \cdots + \phi_{H,p} y_{t-p} + \varepsilon_t) \cdot (1 - G(z_t, m, \gamma)) \end{aligned},$$

其中 $G(\cdot)$ 為羅吉斯函數，而 z_t 為門檻變數，以 x_t 加上延遲項數 (delay order) m，也就是 $z_t = x_{t-m}$ 的設定。

(3) GAMs:

一般化的加成模型 (generalized additive models， GAMs) 依據線性平滑模型做修正。依照迴歸設定，GAMs 具有以下型式：

$$E(y_t \mid x_{1t}, x_{2t}, \ldots. x_{pt}) = \alpha + f_1(x_{1t}) + f_2(x_{2t}) \cdots + f_p(x_{pt}), \tag{5}$$

其中 f_j 是未指定的平滑 (非參數性) 函數。如欲展開基底函數建立模型，則結果由最小平方法算出。GAMs 則是每個函數由散佈點平滑法配置，如三次樣條函數 (cubic spline) 或核平滑函數 (kernel smoothing function)，通過這種方式，GAMs 具有充分彈性包含任何線性和非線性成分的組合。在時間序列應用方面，Wood and Augustin (2002) 延伸懲罰迴歸樣條方法 (penalty regression splines approach) 以便更進一步利用一般化樣條平滑法 (generalized spline smoothness, GSS) 的好

處。本計畫參照 Wood and Augustin (2002) 的方法採用一般化的廣義加成 AR(p) 模型預測經濟成長率。

(4) BATS (**B**ox-Cox transform，**A**RMA，**T**rend & **S**easonality)

Taylor (2003) 延伸 Hold-Winters 的線性版本，並且包含二個季節性成分的模型如式 (6)

$$\begin{aligned} & y_t = L_{t-1} + b_{t-1} + s_t^{(1)} + s_t^{(2)} + e_t, \qquad i = 1,2 \cdots T \\ & L_t = L_{t-1} + b_{t-1} + \alpha \cdot e_t \\ & b_t = b_{t-1} + \beta \cdot e_t \\ & s_t^{(1)} = s_{t-m_1}^{(1)} + \gamma_1 \cdot e_t \\ & s_t^{(2)} = s_{t-m_2}^{(2)} + \gamma_2 \cdot e_t \end{aligned} , \tag{6}$$

其中 m_1 和 m_2 為季節性周期，e_t 是代表干擾項的白噪音變數。L_{t-1} 和 b_{t-1} 分別代表在時間 t 的 y_t 截距和趨勢，s_t 是季節性成分，係數 α，β，γ_1 和 γ_2 則是平滑參數，$\{s_{1-m_1}^{(1)}, \ldots, s_0^{(1)}\}$ 和 $\{s_{1-m_1}^{(2)}, \ldots, s_0^{(2)}\}$ 代表起始狀態變數。

上述模型有幾個缺點，一是不穩定的非線性，另一是尚未被認可的複雜季節性走勢。為了解決這些問題，De Livera *et al*. (2011) 主張採用 BATS 方法預測具有複雜季節性走勢的時間序列，該方法由其四個成分命名，分別為 **B**ox-Cox 轉置，**A**RMA，趨勢 (**T**rend) 和季節性 (**S**easonality)。Box-Cox 轉置是一個廣泛被接納的轉換法，因為它可以穩定時間序列的變異數。BATS 定義如式 (7)

$$y_t^{(\lambda)} = L_{t-1} + \phi b_{t-1} + \sum_{i=1}^{T} s_{t-m_i}^{(i)} + e_t \tag{7}$$

$$
\begin{aligned}
&y_t^{(\lambda)}=\begin{cases}\dfrac{y_t^{\lambda}-1}{\lambda},\lambda\neq 0\\ \log(y_t),\lambda=0\end{cases}\\
&L_t=L_{t-1}+\phi b_{t-1}+\alpha\cdot e_t\\
&b_t=(1-\phi)b_{t-1}+\phi b_{t-1}+\beta\cdot e_t\\
&s_t^{(i)}=s_{t-m_i}^{(i)}+\gamma_i\cdot e_t
\end{aligned}
\quad ,
$$

其中 e_t 符合 ARMA(p, q) 型態。

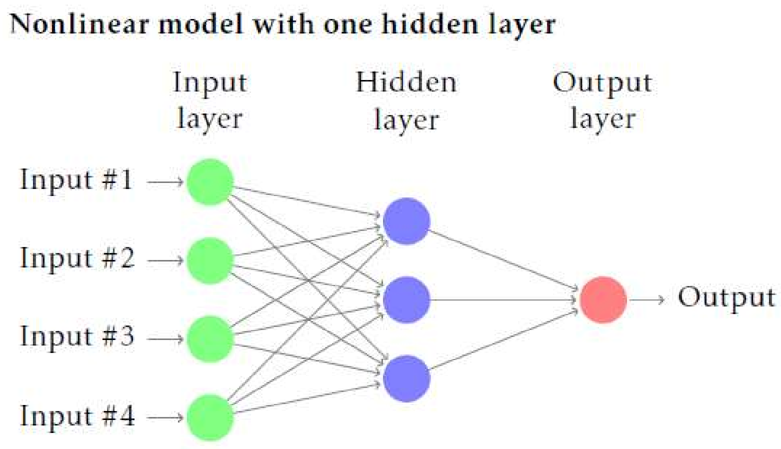
(5) BAGGED (Bootstrap AGGregation)

與 BATS 相似，Bergmeir *et al.* (2016) 為指數平滑的 bootstrap aggregation 提出一個技術方法，因此取部份字母簡稱為 bagged，並為預測績效帶來顯著改進。BATS 依照 STL 分解使用 Box-Cox 轉換法，將時間序列分成三個部分：趨勢、季節性以及殘餘項。

指數平滑的概念，在於相較於較早的觀察值，以最近的觀察值做預測是更適宜的，這表示最近的觀察值應該賦予更高的權重。在本文實證中，ETS 和 ARIMA(*p,d,q*) 自動最佳階次演算 (auto.arima) 皆可用 bagged 方法強化。此案 BAGGED.ETS 指用 bootstrap aggregation 強化的 ETS，BAGGED.autoArima 指用 bootstrap aggregation 強化的 auto.arima。ETS 在文獻上有兩種通用名稱：一指 Error, Trend, Seasonal；二是 ExponenTial Smoothing。有關 ETS 細節，請參考 Hyndman and Athanasopoulos (2018) 第 7 章。

(6) 簡易人工類神經網路 (Simple Artificial Neural Network)

類神經網絡方法本身是包括隱藏層 (hidden layers) 和規模節點 (size nodes) 的非線性設置 (以下簡稱 NNETAR)。該方法是前饋 (feed-forward) 類神經網絡且使用時間序列落後值作投入。此法細節可參考 Hyndman and Athanasopoulos (2018) 的第 11 章，簡述如圖二：



資料來源：Hyndman & Athanasopoulos (2018). *Forecasting: principles and practice*. 2nd edition, OTexts.com/fpp2. 第 11 章。

圖二、類神經網絡模型架構

多層類神經網絡可以想像成一大串的合成函數 (composite functions)

$$x \to f(x) \to g(f(x)) \to h(g(f(x))) \to \cdots\cdots \to y$$

也就是說，函數 *f, g, h* 是由演算法依據資料點對點型態，模擬神經系統運作所產生。整個認知網路的產生很複雜，在類神經網路的演算法中，函數也就由一串一串神經元節點 (neural nodes) 的認知 (感知) 系統所生成，也稱為 layer function。

類神經網絡方法依圖形所示，有三項特色：

(i) 在一個多層前饋網絡中，節點的每一層接收前一層的投入；

(ii) 結合使用線性組合將投入置於每個節點；

(iii) 在輸出前，以非線性函數修改結果。

以線性結合投入至隱藏神經元 (neuron)

$$z_j = b_j + \sum_{i=1}^{4} w_{i,j} y_i \,, \tag{8}$$

使用非線性函數修改後的類神經網絡激活函數 (sigmoid)

$$s(z) = \frac{1}{1+e^{-z}} \,, \tag{9}$$

使用非線性函數傾向減少極端投入值，使得網絡不受離群值影響且仍是穩健的，茲將人工類神經網路特性整理如下：

- 權重以隨機值開始，接著隨觀察值更新；
- 預測中包含隨機，所以網絡由不同之隨機起始點訓練，結果是經由平均而來；
- 必須在事前將隱含層以及每個隱含層的節點數設定好；

尤其是 NNETAR 使用 λ=0.5 的 Box-Cox 轉置，以確保殘差值為近似同質性，該模型可寫成下面型式：

$$y_i = f(y_{t-i}) + \varepsilon_t$$

其中 f 代表在單一層具有四個隱藏節點的類神經網絡，假設殘差項 ε_t 為同質性 (且可為常態分配)，因 ε_t 隨機產生一個值，並反覆模擬未來樣本路徑，無論是從一個常態分配產生或是從歷史值重新採樣。

此外我們也使用了 Franses and Van Dijk (2000) 提出的簡單類神經網絡模型，該模型是以一個隱藏層和線性產出的非線性 AR 呈現：

$$y_{t+s} = \beta_0 + \sum_{j=1}^{D} \beta_j \cdot h(\delta_{0j} + \sum_{i=1}^{m} \delta_{ij} y_{t-(i-1)d}) \,, \tag{10}$$

其中 D 是延遲順序 (delay order) 而 m 是嵌入維度 (embedding dimension)。為求簡化以下該模型簡稱為 NNET。

除了標準的類神經網絡之外，機器學習文獻通常使用循環類神經網絡(RNN)方法，而屬於 RNN 的長短期記憶模型(LSTM)則廣泛應用在預測時間序列。

3.2 機器學習方法

(7) SVM

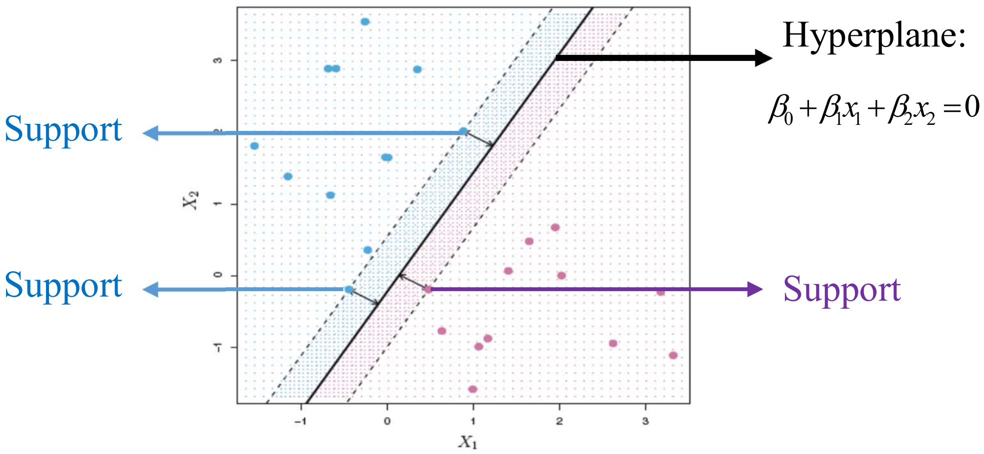
近年來，機器學習方法蓬勃發展，SVM已經被認為是淺層機器學習方法，其角色類似統計方法的線性迴歸，但是其重要性卻不容忽視。SVM的概念由Vapnik (2013)[4] 建立，不同於統計學習方法使用的實證風險最小化原則 (empirical risk minimization，ERM)，將訓練資料數據最小化，SVM體現將結構風險最小化 (structural risk minimization，SRM) 的原理，將一般化誤差的上限最小化，此一差異使得SVM方法較不易受到極端值影響，並可以達成非線性的資料配適與預測，因此具備較高的應用潛力。

SVM是將資料序列模型化，具有強大與充分彈性的研究方法。該方法起源於分類問題，類似線性規劃，找到一個能完美切割資料點的超平面以將樣本資料分類，但亦可以應用在迴歸、預測時間序列等課題。支援向量迴歸(Support Vector Regression, SVR) 隱含的理論建立在SVM分類模型概念上，相較於線性回歸使用最小平方法(least square) 將平方誤差總和(sum of squared errors，SSE) 最小化以尋找迴歸係數估計值（使參數估計值容易受到某一偏離整體數據資料趨勢之觀察值的影響），SVR是嘗試在既定的預測誤差水準與對超過誤差水準的離群值的容忍

[4] 原書首版於 1995。

度下，找到一曲線（或超平面）去配適樣本資料。因為SVR可以調整對離群值的容忍度，並且可以達成非線性的配適，使用SVR可以得到較線性回歸更具彈性的資料配適與預測。

以下從支援向量分類器 (Support Vector Classifier, SVC) 開始，使用圖示與數學模型解釋SVM(或SVR)。如下圖三存在 n 個觀察值的範例，SVC就是找出一個很小的子集合(subset)，做為支撐線性超平面(hyperplane) 的向量。



資料來源： James, *et al*. (2013, 第 9 章)

圖三、SVC 分類示意圖

SVC 的子集合為下述最佳化函數的解得出

$$\underset{\beta_0,\beta_1,\ldots\beta_p}{Max.}\ M$$

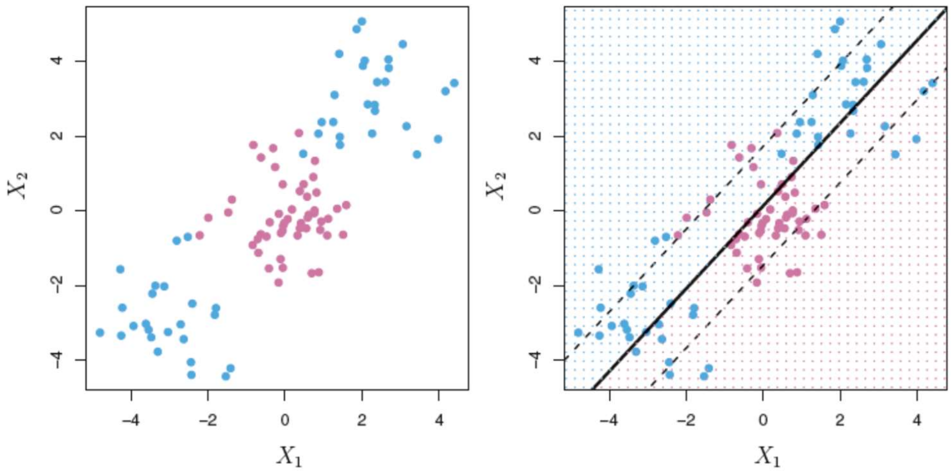
$$s.t.\sum_{j=1}^{p}\beta_j^2=1$$

$$y_i(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip})\geq M(1-\varepsilon_i),\forall i=1,\cdots n,$$

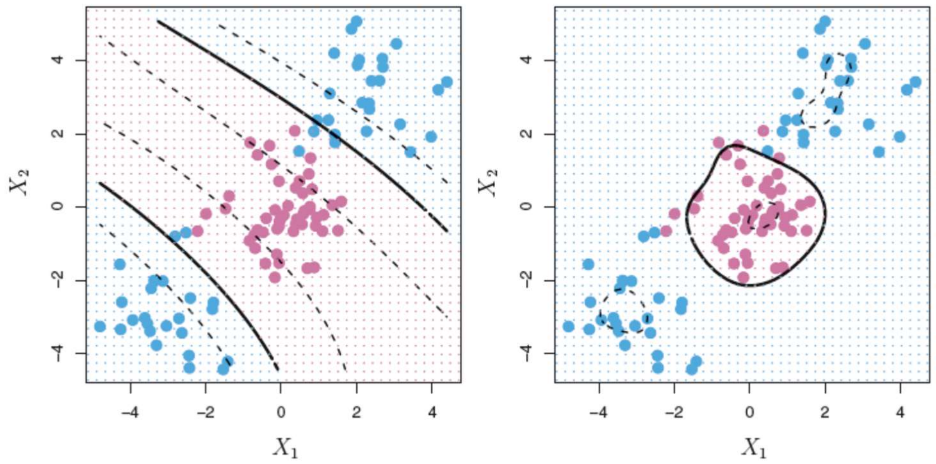
$$\varepsilon_i\geq 0,\sum_{i=1}^{n}\varepsilon_i\leq C$$

$$\forall y_i\in\{-1,1\}:\text{associated class labels}$$

其中 y_i 是分類標籤：當 y_i=1，觀察值為藍色；y_i= −1，觀察值為紅色。此例為雙變數，因超平面的線性組合為 0，若觀察值在此超平面上方(藍色)，$\beta_0+\beta_1x_1+\beta_2x_2$ 即為正數， 且所有觀察值 y_i=1 相乘為正，滿足 M>0 的條件；反之，若觀察值在此超平面下方 (紅色)，$\beta_0+\beta_1x_1+\beta_2x_2$ 為負數，和 y_i = −1 相乘為正，亦滿足 M>0 的條件。SVC 演算法即在解出讓 M 值極大的參數 βs，這個線性分類演算法遇到觀察值不再是線性可分或是非線性時，就出現線性超平面無法將觀察值完全分類的問題。如圖四（A）：



（A）SVM 無法將觀察值分類之示意圖

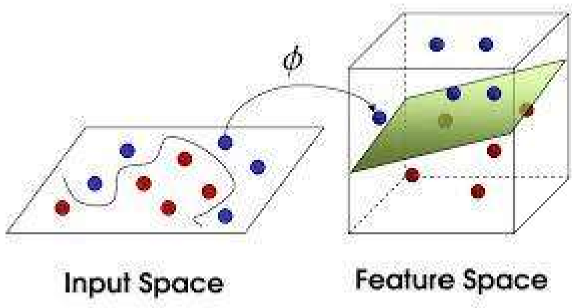


（B）**SVM 用 kernel** 的分類效果

資料來源： James, *et al*. (2013, 第 9 章)

圖四、**SVM** 的分類原理

SVM 就是利用kernel函數，從SVC發展出來的，如圖四（B）。kernel函數簡單地說就是一個投影，把資料從低維度投影到高維度 (超平面)，然後就可以將二維無法分類的資料加以歸類，如圖五。



圖五、**kernel** 投影函數示意圖

就數學規劃，SVM 就是解一個最佳化函數如下：

$$\underset{\beta_0,\beta_1,\cdots\beta_p}{Max.}\ M$$

$$s.t.\sum_{j=1}^{p}\beta_j$$

$$y_i(\beta_0+\sum_{j=1}^{p}\beta_{j1}x_{ij}+\sum_{j=1}^{p}\beta_{j2}x_{ij}^2)\geq M(1-\varepsilon_i),\forall i=1,\cdots n$$

$$\varepsilon_i\geq 0,\sum_{i=1}^{n}\varepsilon_i\leq C,\sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2=1$$

C 是成本函數 (cost function)。

SVM 的差異在於分類器： $f(x)=\beta_0+\sum_{i\in S}\alpha_i\left\langle x,x_i\right\rangle=\beta_0+\sum_{i\in S}K\left\langle x,x_i\right\rangle$

線性 kernel： $K\left\langle x_i,x_j\right\rangle=x_i'x_j$

非線性 kernel 有兩種：

1. d 階多項式 kernel：$K(x_i,x_j)=(r+\gamma\cdot x_i'x_j)^d$。d>0, 一般 r=0，γ為定值(1/n)。

2. Radial kernel：$K(x_i,x_j)=e^{\gamma\cdot x_i'x_j}$ 。此為最廣泛使用，也是實務上較成功的函數。

由上面方程式，SVM 不僅使用 ε 損失函數也考量懲罰項，一個簡易表示法為：SVM 迴歸係數將以下式「成本及懲罰項」方程式最小化

$$Cost\cdot\sum_{i=1}^{n}L_\varepsilon(y_i-\hat{y}_i)+\sum_{j=1}^{P}\beta_j^2\ ,$$

其中 $L_\varepsilon(\cdot)$ 是 ε 的不敏感方程式，$Cost$ 參數是成本懲罰，是用來懲罰大的殘差值。上式的懲罰以脊迴歸 (ridge regression) 的倒數或類神經網絡 (neural networks) 的權值衰減 (weight decay) 表示，因為懲罰附屬於殘差而不是參數。為簡化考量，我們使用 SVM 的穩健性技術支持統計迴歸。[5]

雖然 SVM 和多數機器學習，看似是對類別變數作分類的計算。但是，類似 lm 到 glm，迴歸樹到決策樹，SVM 的分類概念皆可直接用於連續資料，其概念類似統計的連續資料所產生的信賴區間，統計依照條件期望值和標準差建構信賴區間，然後取一個 size (0.05) 把被解釋變數歸成兩類。

我們的 SVM 架構基本上有 AR(P)，據此可以產生動態預測，依次添加時間確定趨勢和季節虛擬變數矩陣。

(8) 自動化機器學習 autoML

我們採用 H_2O.ai 平臺提供的自動機器學習和大數據公開資源解決方案。H_2O.ai 是專業雲端運算平臺，平臺上可使用由 H_2O.ai 公司開發的深度學習人工智慧組件(例如 H_2O.glm 即是在 H_2O 環境建立一個一般化線性模型)，組件中包括尖端機器學習演算法、績效矩陣與附屬函數等，使得機器學習方法強大而不失簡潔。我們在 H_2O.ai 平臺上採用自動化演算嘗試以多個機器學習模式進行演算，包括：監督深度學習 (類神經網絡)、隨機森林 (random forest)、一般化線性模型、梯度提高機器 (gradient boosting machine)、素樸貝式 (naïve Bayes)、疊積集成 (stacked ensembles) 等等。以上模式在訓練和驗證兩個子樣本之間經大規模演算

[5] 對變數型態是 numeric 時，SVM 延伸為 Support Vector Regression (SVR)，即可處理連續的預測問題。應用在經濟成長率預測時，通過經濟成長率或其他變數等特徵值，訓練出這些特徵值和經濟成長率“增加”和“減少”的關係，即通過特徵值劃分指定經濟成長率“增加”和“減少”的邊界，繼之進行預測。

後取出各模型的最佳結果，最後以疊積強化 (stacked ensembles) 合併出最適的預測(代表符號為 autoML)，在機器學習中此方式也稱為委員會方法 (committee approach)，類似統計的預測平均法 (forecasting average)。也就是說，autoML 求得的預測數列並非基於單一種機器學習模型，而是多模型組合模式下的綜合結果。

autoML 是目前機器學習的典型架構，它會依照資料的時間序列結構，產生大量的特徵 (features) 作為解釋變數，如圖六所示：

```
# A tibble: 229 x 31
   date       value index.num    diff  year year.iso  half quarter month month.xts month.lbl   day
   <date>     <dbl>     <int>   <int> <int>    <int> <int>   <int> <int>     <int> <ord>     <int>
 1 1962-06-30  9.11   -2.37e8 7.86e6  1962     1962     1       2     6         5 六月         30
 2 1962-09-30 12.4    -2.29e8 7.95e6  1962     1962     2       3     9         8 九月         30
 3 1962-12-31  6.05   -2.21e8 7.95e6  1962     1963     2       4    12        11 十二月       31
 4 1963-03-31 14.4    -2.13e8 7.78e6  1963     1963     1       1     3         2 三月         31
 5 1963-06-30 10.9    -2.05e8 7.86e6  1963     1963     1       2     6         5 六月         30
 6 1963-09-30  7.35   -1.97e8 7.95e6  1963     1963     2       3     9         8 九月         30
 7 1963-12-31 10.3    -1.89e8 7.95e6  1963     1964     2       4    12        11 十二月       31
 8 1964-03-31 12.9    -1.82e8 7.86e6  1964     1964     1       1     3         2 三月         31
 9 1964-06-30 14.0    -1.74e8 7.86e6  1964     1964     1       2     6         5 六月         30
10 1964-09-30 13.4    -1.66e8 7.95e6  1964     1964     2       3     9         8 九月         30
# ... with 219 more rows, and 19 more variables: hour <int>, minute <int>, second <int>,
#   hour12 <int>, am.pm <int>, wday <int>, wday.xts <int>, wday.lbl <ord>, mday <int>,
#   qday <int>, yday <int>, mweek <int>, week <int>, week.iso <int>, week2 <int>, week3 <int>,
#   week4 <int>, mday7 <int>, ar1 <dbl>
```

資料來源：本研究自行整理。

圖六、自動化機器學習 autoML 時間特徵變數示意圖

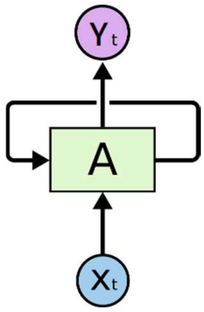
然後根據這些特徵資料，進行大規模的分類演算。例如前述 SVM 我們可添加 AR (P) 以產生動態特徵，但 autoML 已有時間特徵虛擬變數，我們就不需添加諸如趨勢和季節等額外虛擬變數，。

(9) 具 LSTM 效果的 RNN

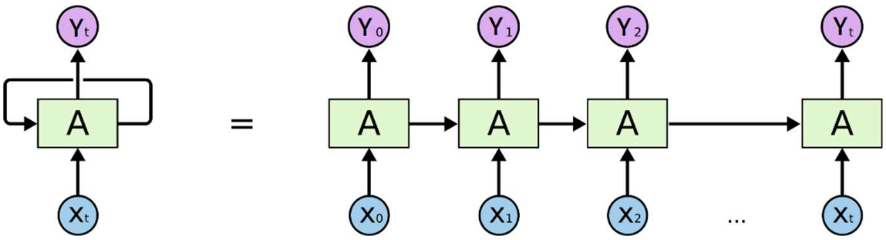
深度學習方法的應用場域是型態辨識 (pattern recognition)，不是為了時間序列預測所開發。不過辨識只是從所掃描的圖形萃取特徵，再和資料庫特徵比對，然後做預測，它其實就是資料驅動的預測。因此若應用在時間序列預測，就需要

透過歷史資料訓練，建一個型態資料庫，然後比對產生預測。而深度學習的類神經模式，對於層級和節點的函數，是由演算法根據資料的複雜關聯產生。

要理解長短期記憶模型(LSTM)需要先理解遞迴類神經網路(Recursive Neural Network, RNN）的概念，3.1 節末介紹的簡易類神經網路的預測路徑(資料輸入與輸出)純粹是單向的，但對某些深度學習與預測問題而言，資料的出現可能有高度的前後相關性。以閱讀為例，人們並非從閱讀一篇文章時才開始思考，而是在閱讀一篇文章時，從之前已認識的文字來理解其意涵。換言之，「想法是持久的」，我們並非將之前每件事物丟棄後重新思考，而是在閱讀過程中持續保留與捨棄某些資訊(或記憶)。又例如在觀看一場電影時，若想將每個時點發生的每件事做分類，很難得知如何使用電影先前事件的推理告知下個事件。以經濟成長率預測為例，若造成經濟成長率變動的因素（例如生產力衝擊或信用危機）具有持續性，則本期經濟成長率的預測值很大一部分受到前期預測值影響。但傳統的前饋型類神經網絡卻無法做到持久性想法。從資料的角度，保留短期記憶是避免局部特徵被全面資料給稀釋 (可理解為平均趨勢稀釋了資料的局部特徵，例如線性迴歸將所有資料以單一斜率配置)，因此對經濟成長預測有幫助。以圖七為例，遞迴類神經網路(RNN) 對此議題主張「有著迴圈的網絡使訊息得以持久」，將前期產生的資訊投入到後期的深度學習程序中。



（A）



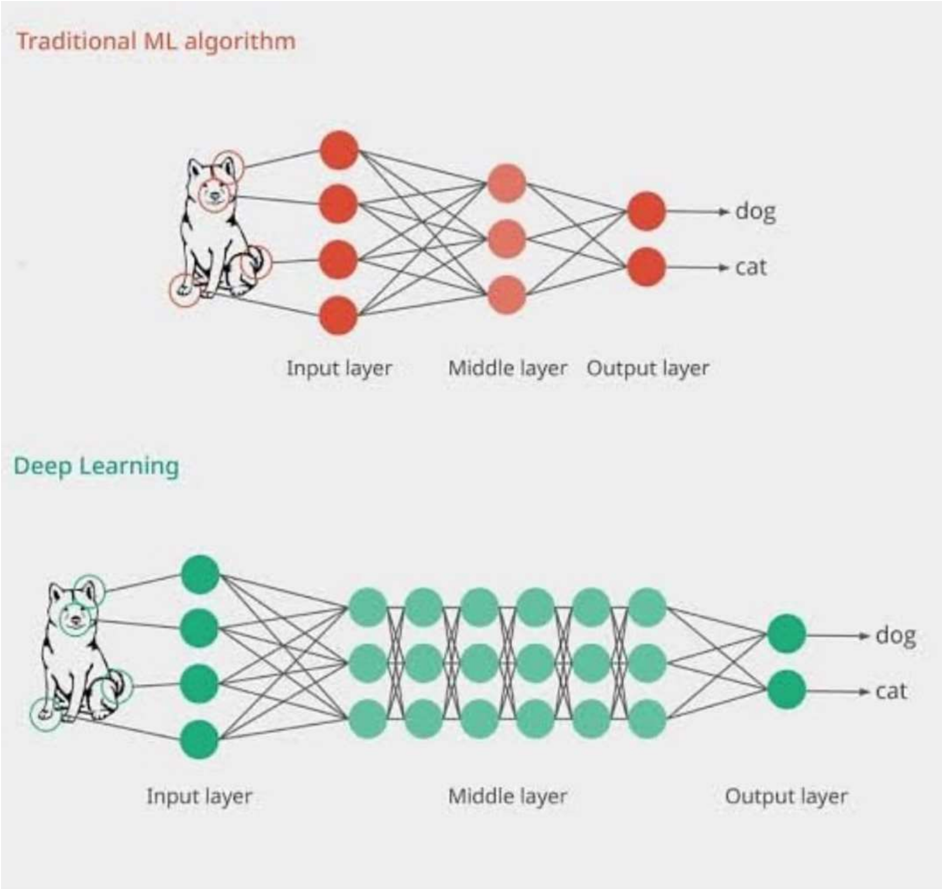
（B）

資料來源：Olah, C. (2015), "Understanding LSTM Networks," https://colah.github.io/posts/2015-08-Understanding-LSTMs/.。

圖七、RNN 迴圈

其中矩形面積 **A** 為一個類神經網絡連接著投入 $\mathbf{x_t}$ 和產出值 $\mathbf{Y_t}$，迴圈使訊息從網絡的某一步驟傳輸至下一步驟。

進一步探究便可發現，圖七（A）相較於一般類神經網絡並非如此不同。RNN 可視為相同網絡的複本 (RNNs)，每個步驟將訊息傳遞至後續者。考量展開迴圈如圖七（B）近似通路的本質，顯示 RNN 各環節密切地序列相關。

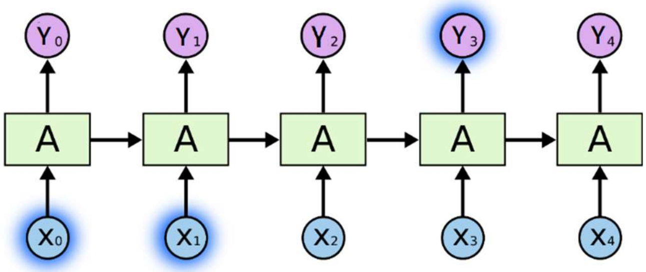


圖八、傳統機器學習和深度學習的比較

圖八比較傳統機器學習和深度學習，後者其實是一種更複雜的神經網路。RNNs 近年在文獻中被大量使用，並應用於如語音辨識 (speech recognition)、語言模型 (language modeling)、翻譯、圖像描述 (image captioning) 等等，且各種應用仍持續有所進展，可參閱 Karpathy (2015) 的詳盡說明。[6] 遞迴類神經網路 (RNN) 網路架構特性使其本身具有記憶性，故已成功應用在時序數據的處理上。應用在預測經濟成長率時，投入 x_t (向前一期經濟成長率及其他變數) 並透過類神經網絡，可預測出 Y_t (經濟成長率預測)。

[6] http://karpathy.github.io/2015/05/21/rnn-effectiveness/

RNNs 其中一個特性，在於能夠將前期與當前的訊息連接，但這視情況而定，因為有時我們只需注視最近訊息執行當前任務。舉例來說，考量一個嘗試以之前文字預測下個文字的語言模型，如果我們試著從以下這句話 “There are many birds flying in the ____”，預測最後一個文字，其實我們不需要更多上下文，很明顯地那個字會是 “sky”。在此情況下，相關訊息和需要位置之間的差異很小，RNNs 可使用過去訊息學習，如下圖 x_0 和 x_1 可得 Y_3。



資料來源：Olah, C. (2015), “Understanding LSTM Networks,” https://colah.github.io/posts/2015-08-Understanding-LSTMs/.。

圖九、在 **RNN** 迴圈使用過去訊息學習

然而大多數情況下我們需要更多前後文。例如我們試著從以下這句話 “I was born in China… I speak fluent ____” 來預測最後一個文字，當前訊息建議下個文字可能是一個語言的名字。但如果我們欲縮小語言範圍，我們就需要更多前後文。有時相關訊息和需要位置之間的差異很大，而 RNNs 無法學習將訊息結合，此稱之為依賴長期記憶問題。

理論來說 RNN 可以處理長期記憶依賴，人們謹慎選取參數，進而以這種形式解係數估計值。不過 RNNs 實際上似乎無法學習這個部分，稱之為梯度消失問題 (vanishing gradient problem)，這在梯度變得相當大或相當小時出現，因而造成輸入資料庫結構長期全距依賴性 (long-range dependencies) 模型化的困難，此時

最具效率性的方法，是使用 DL4J 支持下 RNN 的 LSTM 變異，也被稱為 LSTM 網絡。

如 SVM，LSTM 也添加 AR(P) 以及趨勢和季節等額外虛擬變數。另外，本研究係在模型的合理值範圍內做 grid search，由 RMSE 最小值來判斷最適的層級與節點數量。

3.3 評估模型的預測績效

令 y 和 $\hat{y}$ 分別代表真實值和預測值, 我們應用四個精確衡量指標，就預測誤差 $e_t=y_t-\hat{y}_t$ 評估預測準確性。衡量指標有多個，Makridakis (1993) 針對預測指標穩定性的研究，指出 Mean Absolute Prediction Error (MAPE) 相對穩定。Hyndman and Koehler (2006) 則建議 Mean Absolute Scaled Errors (MASE)。因為本研究為小樣本，scaled 的處理沒有差異。因此採取多個指標綜合判斷，敘述如下：

(1) 兩個尺度依存誤差 (two scale-dependent errors)

首先為 RMSE，定義如下：

$$\text{RMSE}=\sqrt{mean(e_t^2)}$$

，$mean$ 代表平均值，e_t 為誤差，

其次是平均絕對誤差 (mean absolute error, MAE)：

$$\text{MAE}=mean(|e_t|),$$

(2) 百分比誤差 (percentage errors)

令 $p_t=\dfrac{y_t-\hat{y}_t}{y_t}$，平均絕對誤差率 (mean absolute percentage errors, MAPE) 定義如下：

$$\text{MAPE}=mean(|p_t|),$$

(3) Theil's U

用來評估預測準確性的 Theil's U 之定義如下：

$$\text{Theil's U}=\sqrt{\frac{\sum(y_t-\hat{y}_t)^2}{\sum y_t^2}},$$

(4) 殘差項在落後一期的自我相關

殘差項在落後一期的自我相關 (autocorrelation of errors at lag 1, ACF1) 衡量預測誤差的持久性，以下列 ACF 公式中的係數 ϕ 表示：

$$e_t=\phi e_{t-1}+\varepsilon_t,$$

係數 ϕ 亦衡量殘差項隨時間衰減的速度，若 ϕ 值近於 1 則預測誤差較持久，意味著該模型預測力較差，反之則表示預測誤差將迅速衰減至零。

另外，這些預測績效指標在一些研究會展現樣本變異性 (sampling variations)，即隨著抽樣數增加，指標的相對變化 (例如 RMSE/MAPE)。此作法在機器學習的 k-fold 交叉驗證 (cross validation, CV) 也十分常見，但因本案樣本數不多，我們沒有設計不同的訓練和測試時段。我們重心在於比較三個機器學習模式，在各種投入變數組合下的預測表現，藉以探詢作為時間序列預測方法的可行性。所以僅計算預測誤差及其預測績效。

4 資料與預測方法

4.1 經濟成長季資料

行政院主計總處依照國民經濟會計制度 (System of National Accounts，SNA) 規範，分成支出面、生產面以及消費面，按季編製臺灣國內生產毛額 (gross domestic product，GDP) 及其成長率，衡量期間內的生產活動表現。如表一所示，我們採用支出面的實質 GDP(以 2016 年為基期)成長率，將 1962 年第一季至 2019 年第四季，再分成 1962 年第一季至 2016 年第四季，以及 2017 年第一季至 2019 年第四季等兩個子樣本期間，在預測方面，本文分別以傳統時間序列和機器學習方法為臺灣實質 GDP 年成長率進行預測。

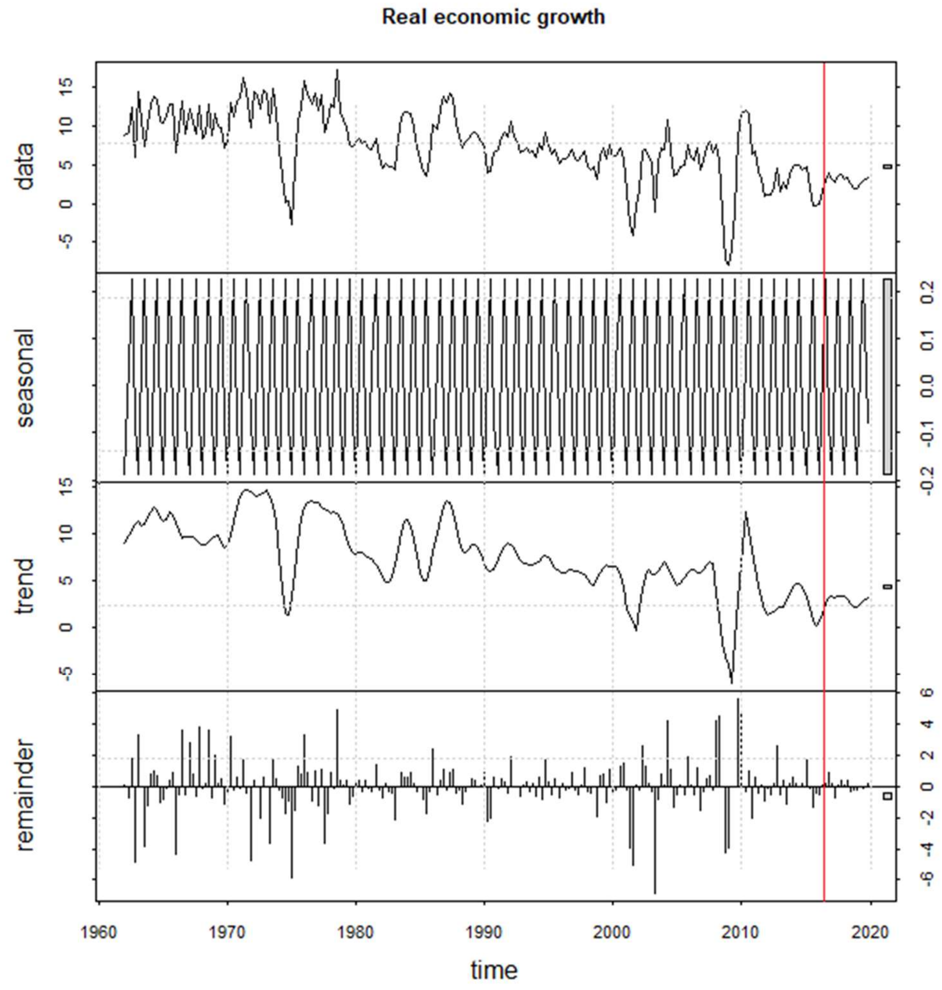
表一、臺灣經濟成長率之敘述統計

	經濟成長率(%)	
	1962Q1-2016Q4	2017Q1-2019Q4
平均數	7.49	3
中位數	7.26	3.23
標準差	4.38	0.65
偏度	-0.55	-0.39
峰度	0.74	-1.19
最小值	-7.88	1.84
最大值	17.26	3.92
樣本數	219	13

資料來源：中華民國統計資訊網；本研究自行整理。

本預測模型子樣本期間如下：樣本內期間為 1962 年第一季至 2016 年第四季，樣本外期間為 2017 年第一季至 2019 年第四季。樣本內期間為訓練樣本 (training sample)，由此樣本配適出合適的模型或係數，樣本外期間則是交叉驗證的測試樣本 (testing sample)。兩個子樣本期間的時間趨勢圖及敘述統計如前述圖一和表一所示，本研究採用 1962 年第一季至 2019 年第四季，以支出面計算的臺灣經濟成長率季資料。圖一呈現兩段資料的時間序列走勢圖，可比較其波動幅度。期間臺灣經濟成長率大多為正值，在 1975 年受到石油危機衝擊，從 15%巨幅下跌至約-2%，隨後反彈並在 1970 年代末期達到將近 17.3%的最高水準，其後逐漸趨緩。於 2001 年和 2003 年分別因科技類股泡沫化和 SARS，經濟成長率呈現負值，2009 年由於全球金融危機，經濟成長率近-7.9%。2010 年升至近 12%，之後多維持在 0% 至 5% 之間。由表一可發現：(1)第一段子樣本期間經濟成長率平均值、中位數與標準差皆高於第二段子樣本期間；(2)在偏態方面，兩段子樣本皆較呈現常態分配，在峰態部分，第一段樣本期間為高峻峰，第二段為低闊峰。

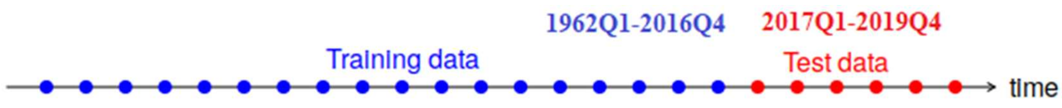
基於如圖十所示 Loess 函數的季節-趨勢分解法 (seasonal and trend decomposition, STL)，各列右側的直條長度為比例尺，以供分解出的各個成份下進行尺度比較。經濟成長率為非定態 (non-stationary)，其趨勢呈現出明顯的隨機性質。季節性成分係指季節變動所形成的短期周期波動，顯示該波動幅度並不規律。



資料來源：本研究自行整理。

圖十、臺灣經濟成長率的 **STL** 分解

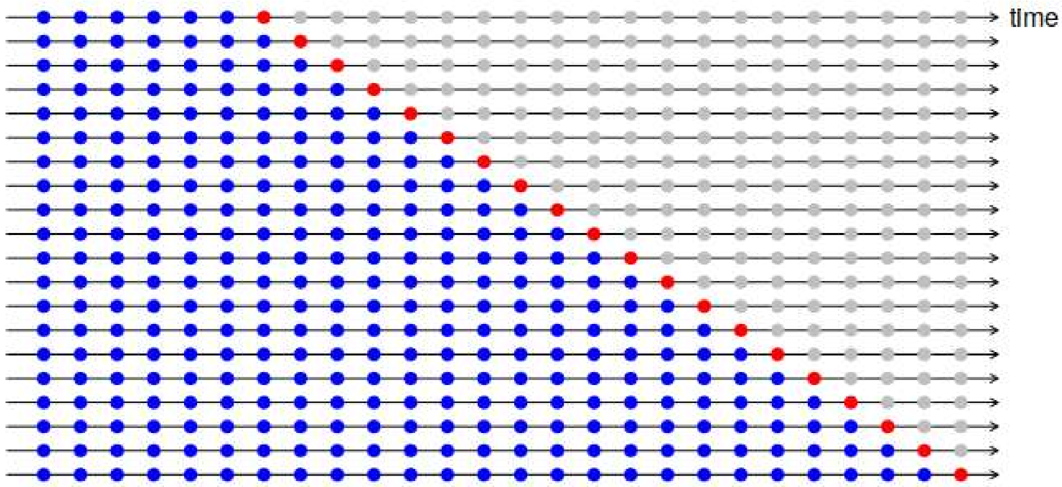
由於使用「訓練、驗證、測試」三期作法，資料量必須夠大，故我們的方法設計如圖十一所示：1962Q1-2016Q4 為訓練期，2017Q1-2019Q4 為測試期，不另外採用驗證期 (validation) 的原因在於演算法在訓練期都已內建驗證。SVM 採用 10-fold 交叉驗證，autoML 和 LSTM 採用 5-fold 交叉驗證，因此訓練期產生的模式就是驗證後的模式。在機器學習設計中，訓練期會 learning by iterating 直到選出最佳模型。



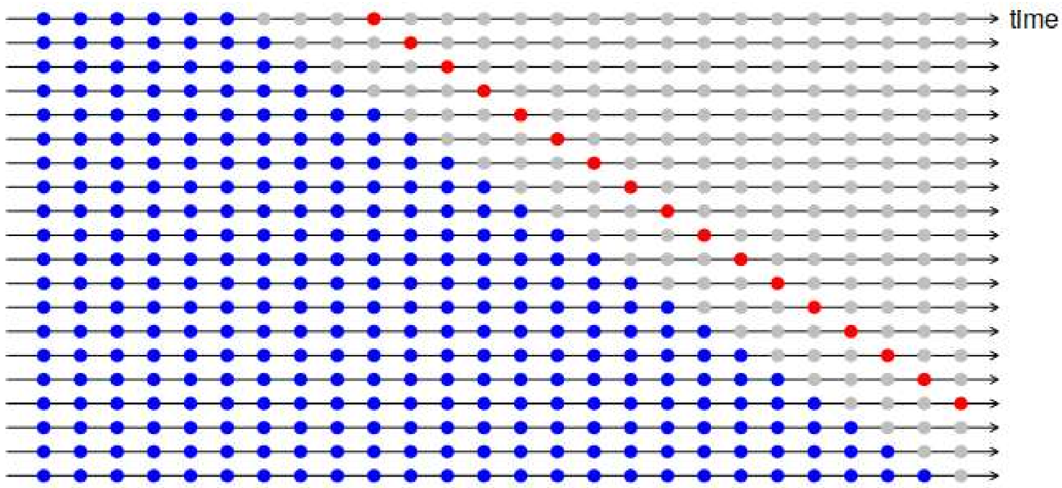
資料來源：本研究自行整理。

圖十一、本研究之機器學習訓練及測試期

臺灣經濟成長率共 232 季的觀察值，用 (6:2:2) 標準分割對傳統時間序列模型是足夠的，但對目前冠上學習演算法之名的 autoML 或 LSTM，至少需要 250 個觀察值，因為資料的序列分段，會導致樣本偏少而訓練不足。此處採用預測樣本外 12 季，也是遷就下的選擇。兩種資料滾動方式如圖十二所示：Chakraborty and Joseph (2017) 的英格蘭銀行研究樣本外只用 8 季，以 60 季開始用逐次增 1 的方法 recursively update，如圖十二 (A) 的 rolling。我們沒有使用 rolling 樣本遞增方式的原因，在於本案沒有要探索精確指標如 MAPE 的樣本變異性，就訓練階段所採用的 CV，用一筆最長資料 (1962Q1-2016Q4) 訓練出來的結果就足以預測，不需要分段 rolling。Chakraborty and Joseph (2017) 因為沒有採用自動化機器學習做大規模強化預測，所以可以小規模學習。就一般機器學習而言，小樣本或許只是訓練不足，但是就自動化機器學習而言，往往是訓練不良，容易產生過度配置。因此在達到本研究目的考量下，我們沒有做他種拆分。



（A）單步滾動



（B）多步滾動：Evaluation on a rolling forecasting origin

資料來源：本研究自行整理。

圖十二、遞迴預測方法示意圖

4.2 預測值的計算方法

本文模型除了相關虛擬變數如季節和趨勢，均會使用至少一階 AR。因此除了單步 (One-step) 靜態預測，另增加了多步 (multistep) 預測。在樣本內期間訓練數據，亦即由數據配適出最合適的模型，再進行單步與多步預測。單步預測亦稱靜態預測，是將 t+2 的預測，建立在 t+1 期的資料。多步預測在文獻中有遞迴 (Recursive) 和直接 (Direct) 兩種方法。

單步預測是將 t+2 的預測，建立在 t+1 期的預測，步驟如下：

(1) 估計樣本內相關參數；

(2) 樣本外第 1 期，使用前 1 期解釋變數資料，計算樣本外第 2 期預測；

(3) 依序再代入向前 2….N 期解釋變數真實值，與樣本內資料估計係數，產生被解釋變數的向前 2….N 期預測值。亦即利用解釋變數真實值的落後期，從預測樣本的第一期，計算樣本外預測。

多步預測步驟如下：

(1) 估計樣本內相關參數；

(2) 樣本外第 1 期，使用前 1 期的預測，計算樣本外第 2 期預測；

(3) 用第 2 期的預測值，計算第 3 期的預測值；

(4) 重複步驟 (2) 及 (3)，依序產生向前 2….N 期被解釋變數預測值。

以前期的預測值再進行未來期數值的預測，稱為「動態多步預測」。遞迴多步是一步一步遞迴而成多步。

假設一個向量變數x_t，該變數與其落後期相關，樣本期間為 t=1, …, n，現有訊息是$X_n = \{x_1, \dots, x_n\}$，$x_t$為以下 AR(1)模型：

$$x_t = a + b_1 x_{t-1} + \varepsilon_t. \tag{12}$$

在時間點為 t= n+1 的 AR(1) 模型：

$$x_{n+1} = a + b_1 x_n + \varepsilon_{n+1}, \tag{13}$$

根據$\mathrm{E}[\varepsilon_{n+1}|X_n] = 0$, 樣本外第一期的預測值為：

$$\hat{x}_{n+1} = E[x_{n+1}|X_n]$$

$$\rightarrow \hat{x}_{n+1} = E[a + b_1 x_n + \varepsilon_{n+1}|X_n] = a + b_1 x_n + \mathrm{E}[\varepsilon_{n+1}|X_n] = a + b_1 x_n, \tag{14}$$

預測誤差及其變異數分別為$x_{n+1} - \hat{x}_{n+1} = \varepsilon_{n+1}$ 和 σ^2。此為單步預測。

在時間點為 t= n+2 的 AR (1) 模型：

$$x_{n+2} = a + b_1 x_{n+1} + \varepsilon_{n+2}, \tag{15}$$

根據$\mathrm{E}[\varepsilon_{n+2}|X_n] = 0$, 樣本外第二期的預測值為：

$$\hat{x}_{n+2} = E[x_{n+2}|X_n]$$

$$\rightarrow \hat{x}_{n+2} = E[a + b_1 x_{n+1} + \varepsilon_{n+2}|X_n]$$

$$\rightarrow \hat{x}_{n+2} = a + b_1 E[x_{n+1}|X_n] + \mathrm{E}[\varepsilon_{n+2}|X_n] = a + b_1 \hat{x}_{n+1}\ , \tag{16}$$

預測誤差及其變異數分別為：

$$x_{n+2} - \hat{x}_{n+2} = \varepsilon_{n+2} + b_1(x_{n+1} - \hat{x}_{n+1}) = \varepsilon_{n+2} + b_1 \varepsilon_{n+1} \text{ 和 } \sigma^2(1 + b_1)。$$

按照此步驟，則在時間點為 t= n+h 的 AR(1)模型：

$$x_{n+h} = a + b_1 x_{n+h-1} + \varepsilon_{n+h}, \tag{17}$$

根據$\mathrm{E}[\varepsilon_{n+h}|X_n] = 0$, h 期以後的預測值為

$$\hat{x}_{n+h} = E[x_{n+h}|X_n]$$

$$\rightarrow \hat{x}_{n+h} = E[a + b_1 x_{n+h-1} + \varepsilon_{n+h}|X_n]$$

$\rightarrow \hat{x}_{n+h} = a + b_1 E[x_{n+h-1}|X_n] + \mathrm{E}[\varepsilon_{n+h}|X_n] = a + b_1 \hat{x}_{n+h-1}$ ， (18)

預測誤差為：

$x_{n+h} - \hat{x}_{n+h} = \varepsilon_{n+h} + b_1(x_{n+h-1} - \hat{x}_{n+h-1}) = \varepsilon_{n+h} + b_1 \varepsilon_{n+h-1}$，預測誤差之變異數隨著預測期間愈大，此為多步預測。

直接多步則直接依據預測期估計模型。例如要預測未來 4 期，可以直接估計一個 $y_t = f(y_{t-4})$，然後帶入$(y_{t-3}, y_{t-2}, y_{t-1}, y_t)$。因此實際上的模式更為複雜，因為 $y_t = f(y_{t-5})$也可以達到同樣的預測目的。所以在直接落後中，落後期是向量，不是純量。一般說明如下：

$$y_t = m_h(\mathrm{A}_{t-h};\gamma_h) + e_{t,h}\ ,\ \mathbf{h}=1,2\ldots, H, \tag{19}$$

上式中 m_h 為 h-期的模型函數，$A_{t-h} = [y_{t-h,}\cdots, y_{t-h-p_h}]$，$p_h$ 是 h 階次模型的 delay order，γ_h 是模型參數集合。同 recursive 方法，直接預測法的解是極小化 *h*-step-ahead 的預測誤差：

$$(p_h, \hat{\psi}_h) = \arg\min_{p,\psi} \sum_{(A_{t-h}, y_t)\in D_{cv,h}} [y_t - m_h(z_{t-1};\psi, \hat{\beta}_h)]^2 \ , \tag{20}$$

然而如同 Bontempi *et al.* (2013) 指出，機器學習的直接多步預測，針對一個預測期 h，要解一個模型 m_h 的多種組合，演算十分龐大。我們在第六章之後，皆採取直接預測，做大規模運算。

5 時間序列模型與機器學習模型搭配 AR(1) 的預測表現

5.1 單步靜態預測

本研究比較多個時間序列模型與機器學習模型的單步靜態預測績效，以 RMSE、MAE、 MAPE 、Theil's U 和 ACF (1) 來衡量，結果如表二及圖十三。圖十三 (A)-(D) 繪出各模型 2017 年第一季之後的預測能力比較，以 RMSE、MAE、MAPE、Theil's U，顯示各模型預測正確性。若綜合考量其結果，時間序列模型優於機器學習方法，其中時間序列的 baggedModel 和 ARIMA (2,1,2)(2,0,1) 相關設定模型最佳，機器學習中考量趨勢項的 SVM:AR(1) 與純粹 SVM:AR(1) 模型次之， AutoML:AR(1) 模型殿後，再來則是虛擬季節性變數和趨勢項的 SVM:AR(1)，而 LSTM:AR (1) 模型表現較差。然而 Theil's U 指出 AutoML:AR(1) 模型最佳， baggedModel 次之，接著為 LSTM:AR (1) 和 ARIMA (2,1,2)(2,0,1) 之家族設定模型居中，SVM:AR(1) 家族模型排名位居後段。

圖十四為單步預測誤差的 ACF (1) 值，顯示預測誤差的持續性，並指出機器學習中虛擬季節性變數和趨勢項的 SVM:AR(1)、虛擬季節性變數的 SVM:AR(1)、LSTM:AR(1)，以及時間序列模型中考量趨勢項的 ARIMA (2,1,2)(2,0,1) 模型預測誤差 ACF (1) 值最小。機器學習的 AutoML:AR(1)、加入趨勢項的 SVM:AR(1)、時間序列方法 ARIMA (2,1,2)(2,0,1) 的設定模型和 baggedModel 的 ACF (1) 值表現居後。前述四個指標表現較佳的時間序列模型，如 ARIMA (2,1,2)(2,0,1) 模型和 baggedModel 在 ACF (1)值的表現較差，而在四個指標表現居於後段的機器學習模型 SVM:AR(1)、LSTM:AR(1) 和 AutoML:AR(1) 則在 ACF (1) 表現較佳。

綜合前述，各機器學習模型與時間序列模型的預測表現上，前者以考量趨勢項的 SVM:AR(1) 與 AutoML:AR(1) 模型較佳，後者則是 bagged 模型和 ARIMA(2,1,2)(2,0,1) 家族模型。但就整體而言，時間序列模型預測表現和機器學習模型沒有太大差距，這樣我們就要檢視 ACF(1) 係數。根據圖十四，前 10 名都沒有太大的 ACF(1) 值。因此在單步預測，機器學習和時間序列模型皆有不錯的表現。

最後檢視單步預測的整體時間序列表現。圖十五至圖十七比較機器學習模型，包括 SVM 四個模型設定、AutoML 和 LSTM 的單步預測值 (含 2016 年第四季後的樣本外預測值)。圖十五顯示 SVM 之四個模型中，AR(1) 和含季節性虛擬變數之 AR (1) 等模型，不僅在捕捉樣本內序列最好，樣本外預測表現亦最佳。

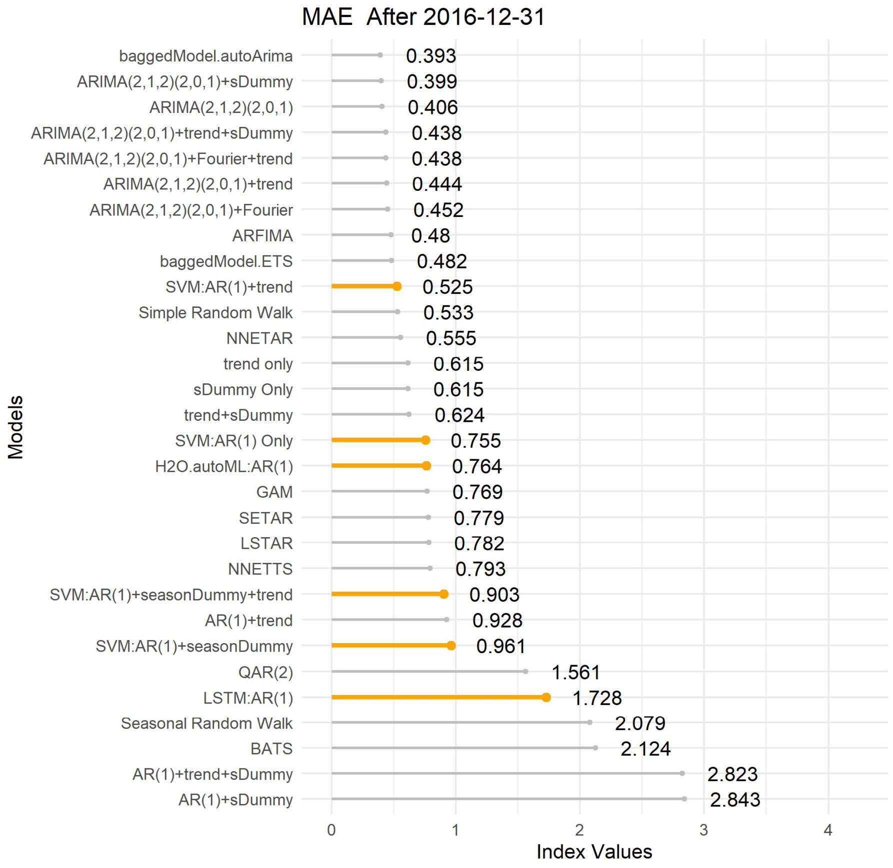
比較圖十六的 AutoML 以及圖十七的 LSTM，以 2000 年後的期間而言，本文發現 LSTM 不論是在樣本內或樣本外預測表現都是較好的。即使受到金融海嘯的衝擊，該機器學習模型在 2009 年的低谷，與 2010 年因基期較低而導致的大幅回升，都能夠進行良好的配適。在單步預測方面，SVM 在所有機器學習方法中的預測績效最好。

綜合上述，如果預測單位只需要單步預測，則傳統時間序列模型就可以有很好的表現，不必浪費時間去執行冗長的計算，此與 Makridakis and Hibon (2000) 和 Morlidge (2014) 一致。接下來我們將檢視多步遞迴預測。

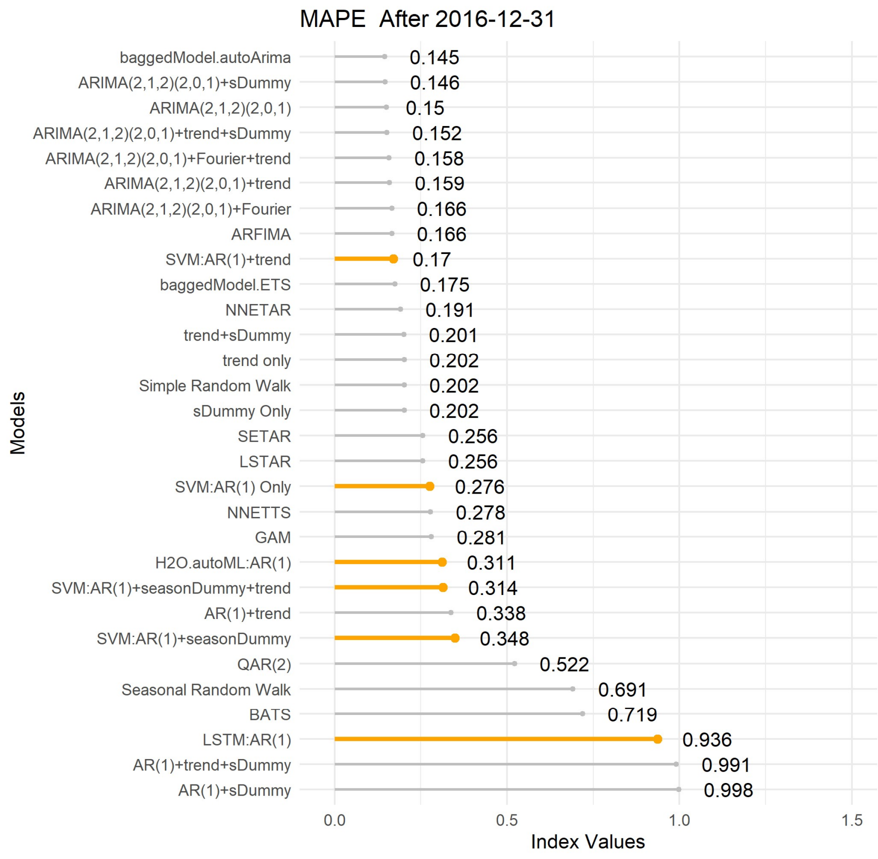
表二、單步 **One-step** 預測績效

	RMSE	MAE	MAPE	ACF1	Theil's U
Simple Random Walk	0.624	0.533	0.202	0.511	0.0116
Seasonal Random Walk	2.488	2.079	0.691	0.067	0.0344
trend only	0.696	0.615	0.202	0.432	0.0107
sDummy Only	0.696	0.615	0.202	0.432	0.0107
trend+sDummy	0.709	0.624	0.201	0.405	0.0101
AR(1)+trend	1.104	0.928	0.338	0.159	0.0159
AR(1)+sDummy	3.004	2.843	0.998	0.138	0.0445
AR(1)+trend+sDummy	2.981	2.823	0.991	0.124	0.0444
ARIMA(2,1,2)(2,0,1)	0.472	0.406	0.150	0.024	0.0087
ARIMA(2,1,2)(2,0,1)+trend	0.502	0.444	0.159	-0.147	0.0086
ARIMA(2,1,2)(2,0,1)+sDummy	0.458	0.399	0.146	0.048	0.0086
ARIMA(2,1,2)(2,0,1)+trend+sDummy	0.502	0.438	0.152	$-6 \cdot 10^{-4}$	0.0086
ARIMA(2,1,2)(2,0,1)+Fourier	0.550	0.452	0.166	0.062	0.01
ARIMA(2,1,2)(2,0,1)+Fourier+trend	0.524	0.438	0.158	0.086	0.0098
baggedModel.ETS	0.592	0.482	0.175	0.293	0.0085
baggedModel.autoArima	0.468	0.393	0.145	0.262	0.0074
NNETAR	0.649	0.555	0.191	0.263	0.0116
BATS	2.684	2.124	0.719	0.111	0.0282
ARFIMA	0.544	0.480	0.166	-0.076	0.0095
LSTAR	0.947	0.782	0.256	0.040	0.0135
SETAR	0.940	0.779	0.256	0.018	0.0135
GAM	0.905	0.769	0.281	0.094	0.0138
NNETTS	0.920	0.793	0.278	0.070	0.0141
QAR(2)	2.582	1.561	0.522	0.400	0.0466
SVM:AR(1)+seasonDummy	1.068	0.961	0.348	-0.204	0.018
SVM:AR(1)+trend	0.622	0.525	0.170	0.037	0.01
SVM:AR(1)+seasonDummy+trend	0.967	0.903	0.314	-0.245	0.0175
SVM:AR(1) Only	0.892	0.755	0.276	0.015	0.0136
autoML:AR(1)	0.937	0.765	0.311	0.018	0.0055
LSTM:AR(1)	2.027	1.729	0.936	-0.095	0.0087

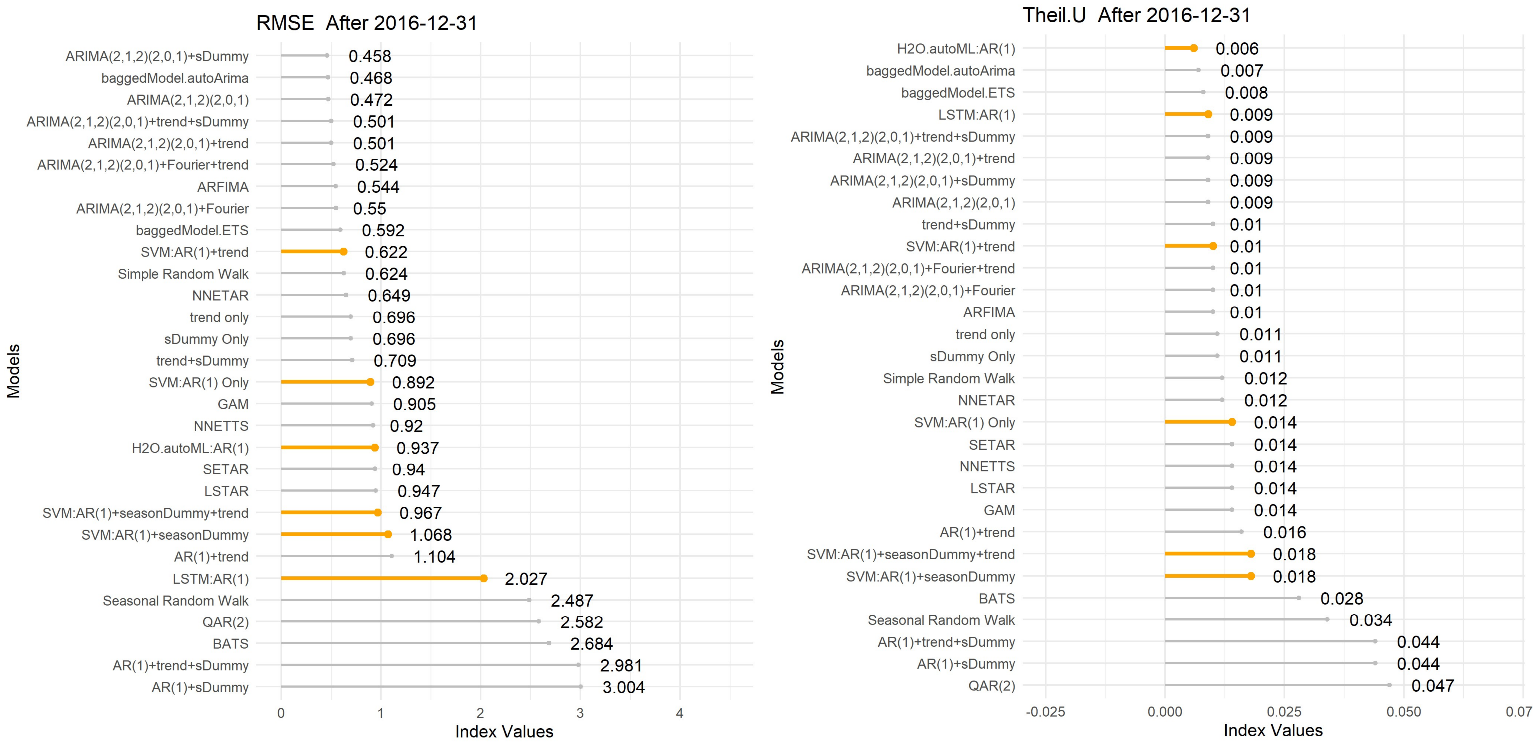
資料來源：本研究自行整理。


MAE After 2016-12-31
Models
Index Values
baggedModel.autoArima 0.393
ARIMA(2,1,2)(2,0,1)+sDummy 0.399
ARIMA(2,1,2)(2,0,1) 0.406
ARIMA(2,1,2)(2,0,1)+trend+sDummy 0.438
ARIMA(2,1,2)(2,0,1)+Fourier+trend 0.438
ARIMA(2,1,2)(2,0,1)+trend 0.444
ARIMA(2,1,2)(2,0,1)+Fourier 0.452
ARFIMA 0.48
baggedModel.ETS 0.482
SVM:AR(1)+trend 0.525
Simple Random Walk 0.533
NNETAR 0.555
trend only 0.615
sDummy Only 0.615
trend+sDummy 0.624
SVM:AR(1) Only 0.755
H2O.autoML:AR(1) 0.764
GAM 0.769
SETAR 0.779
LSTAR 0.782
NNETTS 0.793
SVM:AR(1)+seasonDummy+trend 0.903
AR(1)+trend 0.928
SVM:AR(1)+seasonDummy 0.961
QAR(2) 1.561
LSTM:AR(1) 1.728
Seasonal Random Walk 2.079
BATS 2.124
AR(1)+trend+sDummy 2.823
AR(1)+sDummy 2.843
0
1
2
3
4

(A)


MAPE After 2016-12-31
Models
Index Values
baggedModel.autoArima 0.145
ARIMA(2,1,2)(2,0,1)+sDummy 0.146
ARIMA(2,1,2)(2,0,1) 0.15
ARIMA(2,1,2)(2,0,1)+trend+sDummy 0.152
ARIMA(2,1,2)(2,0,1)+Fourier+trend 0.158
ARIMA(2,1,2)(2,0,1)+trend 0.159
ARIMA(2,1,2)(2,0,1)+Fourier 0.166
ARFIMA 0.166
SVM:AR(1)+trend 0.17
baggedModel.ETS 0.175
NNETAR 0.191
trend+sDummy 0.201
trend only 0.202
Simple Random Walk 0.202
sDummy Only 0.202
SETAR 0.256
LSTAR 0.256
SVM:AR(1) Only 0.276
NNETTS 0.278
GAM 0.281
H2O.autoML:AR(1) 0.311
SVM:AR(1)+seasonDummy+trend 0.314
AR(1)+trend 0.338
SVM:AR(1)+seasonDummy 0.348
QAR(2) 0.522
Seasonal Random Walk 0.691
BATS 0.719
LSTM:AR(1) 0.936
AR(1)+trend+sDummy 0.991
AR(1)+sDummy 0.998
0.0
0.5
1.0
1.5

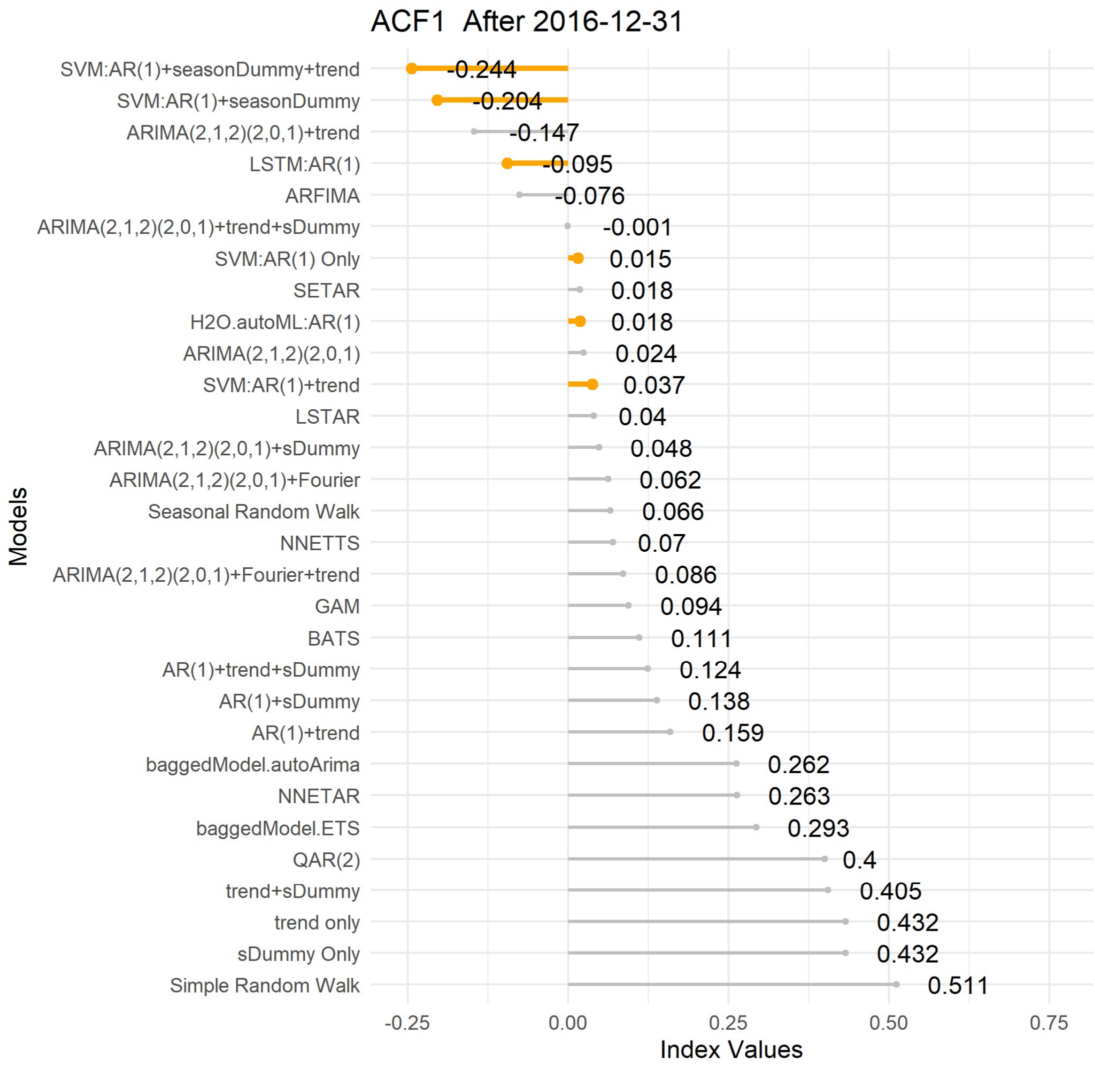
(B)



(C) (D)

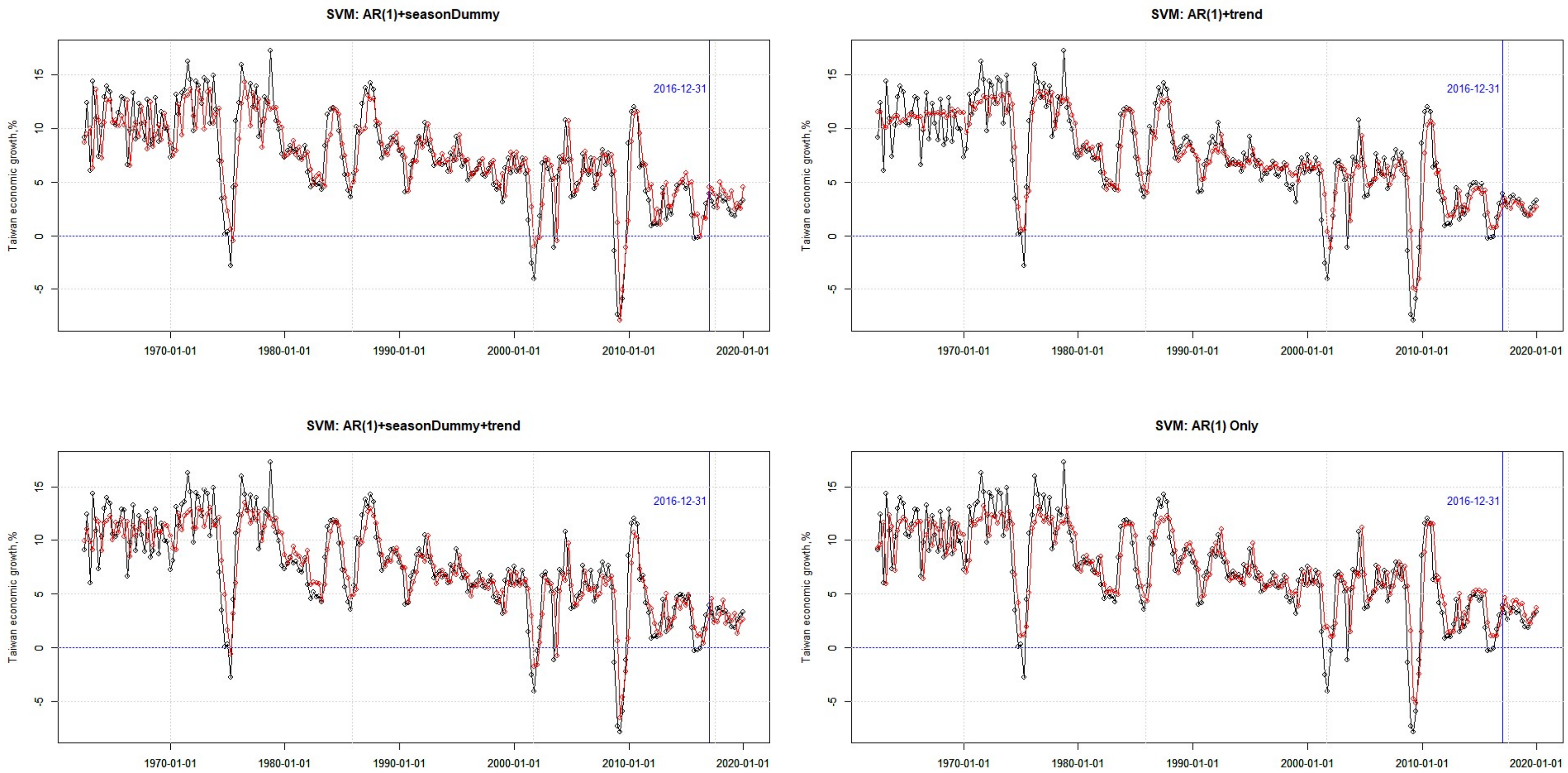
資料來源：本研究自行整理。

圖十三、單步預測正確性：**4** 個指標



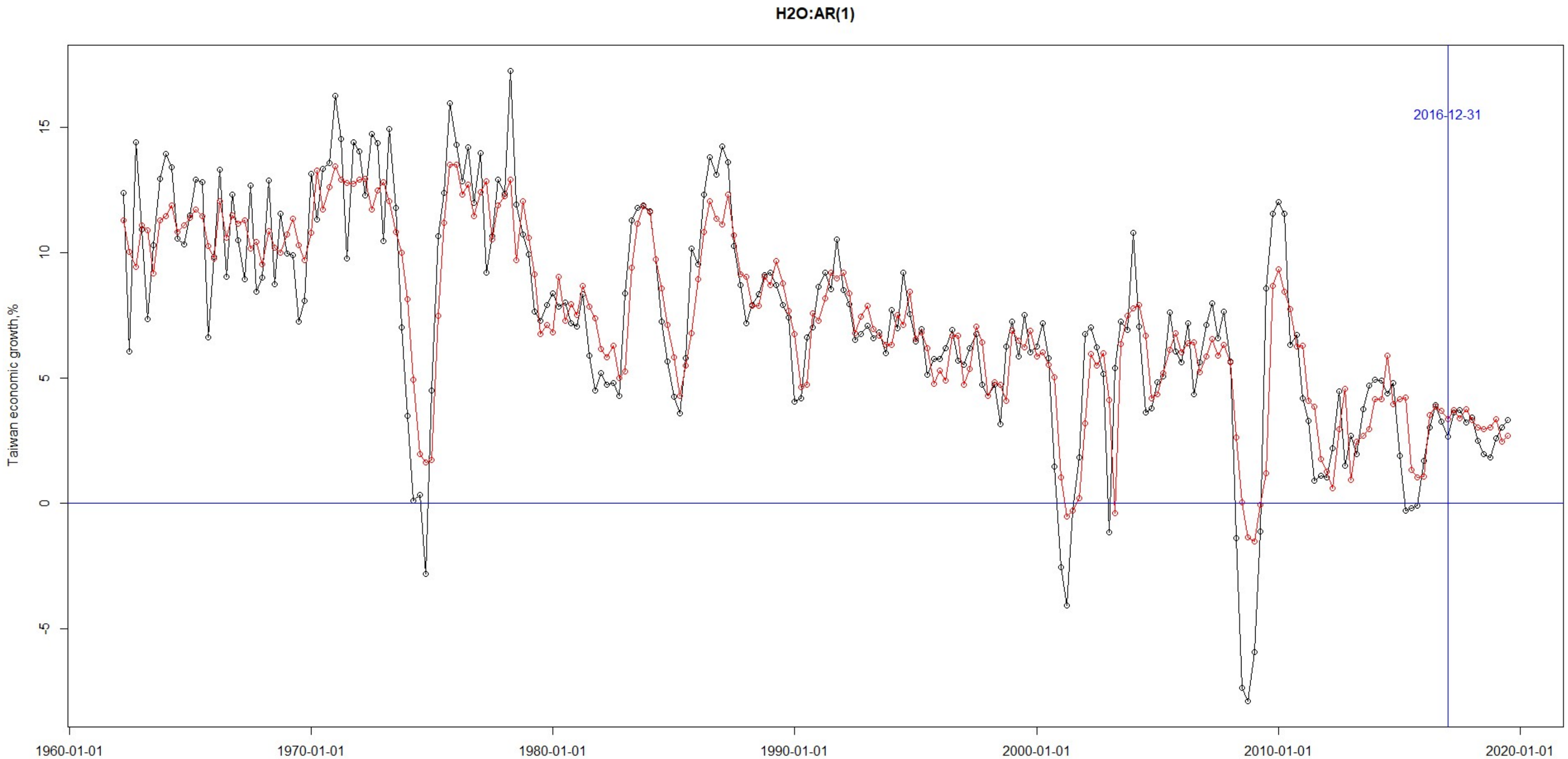
資料來源：本研究自行整理。

圖十四、單步預測正確性：**ACF(1)** 值



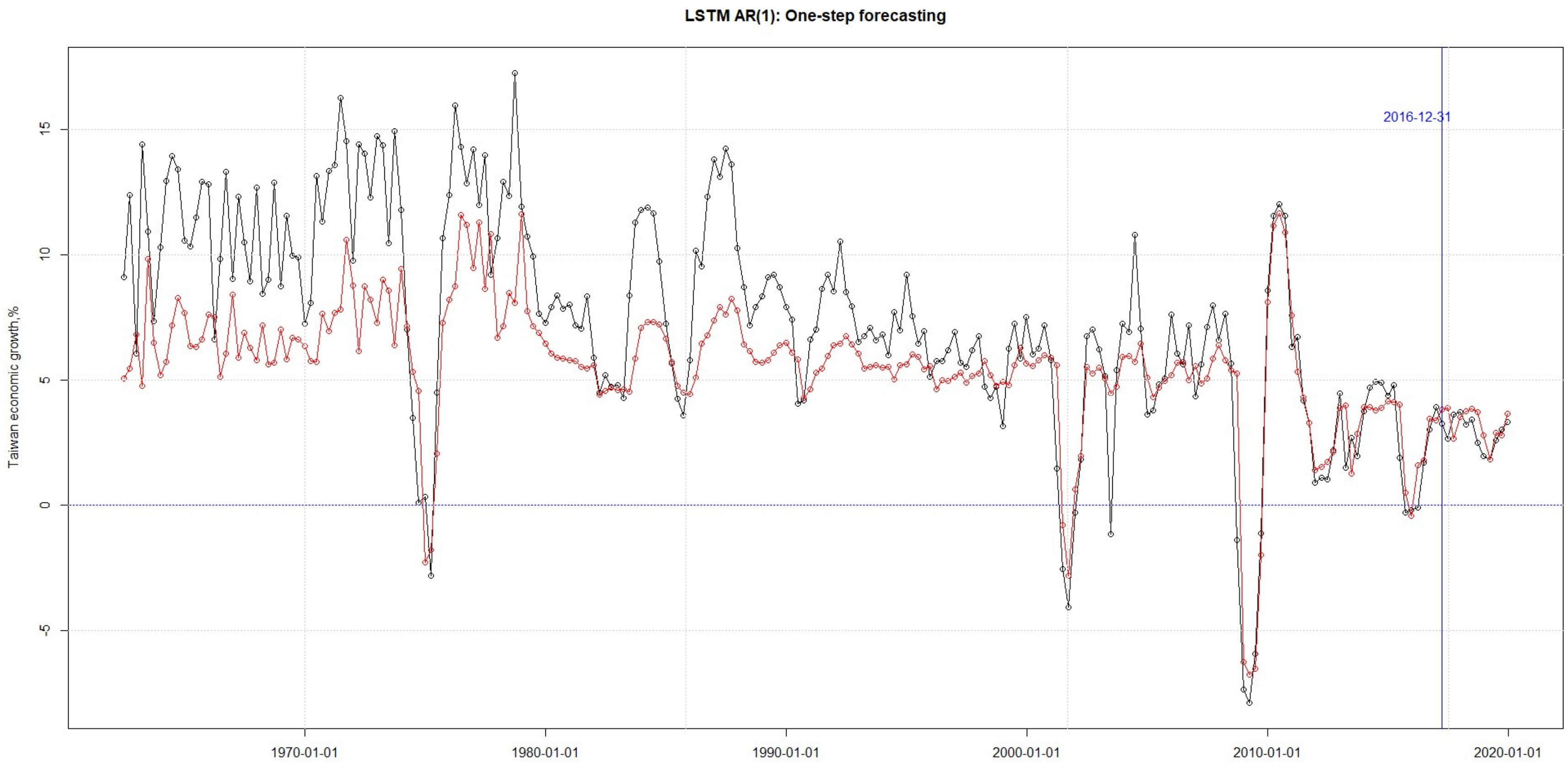
資料來源：本研究自行整理。

圖十五、單步預測的整體時間序列表現：**SVM 4** 個設定



資料來源：本研究自行整理。

圖十六、單步預測的整體時間序列表現：**autoML**



資料來源：本研究自行整理。

圖十七、單步預測的整體時間序列表現：**LSTM**

5.2 多步動態遞迴預測

依據前述架構，本文也進行 2016 年第四季後多步動態遞迴預測並評估其表現，所有模型預測績效指標，按表現優劣排列如圖十八及十九。

圖十八顯示在預測正確性方面，autoML multistep:Recursive 在 RMSE、MAPE、MAE 及 Theil's U 上均有名列前茅的表現，baggedModel 以及 ARIMA (2,1,2)(2,0,1) 相關設定模型則次之，再來是考量趨勢項和季節性虛擬變數的 SVM:AR (1)。SVM:AR(1) 家族的其他設定模型排名居中，並且位居 LSTM multistep:Recursive 之前。但是由數據來看，前 10 名的預測誤差相距甚小，因此我們繼續檢視圖十九的預測誤差持續性。

就圖十九的前幾名來看，加入趨勢項和季節性虛擬變數的 SVM:AR(1)，AutoML multistep:Recursive，以及 ARIMA (2,1,2)(2,0,1) 相關設定模型等三個模式的 ACF(1) 係數介於 0.232-0.389，持續預測錯誤的表現亦佳。然而 SVM:AR (1) 其他三個設定的 ACF(1) 係數較大，在所有模型位居中間與較後端排名，這表示 SVM 設置不正確，就難以產生良好的預測模型。至於 LSTM multiStep:Recursive 排名則與 SVM: AR (1) 其他三個設定模型差異不大。再者，在多步動態遞迴預測的 LSTM 模型，其 ACF(1) 係數較大且排名較後段，這與單步預測的 ACF(1) 呈現結果相反。

最後檢視多步動態遞迴預測的整體時間序列表現，本文將機器學習下的預測值與真實時間序列描繪於圖二十至圖二十二，其中預測值在 2016 年第四季前為樣本內預測值，2016 年第四季後則為樣本外多步預測值。

首先在 SVM 四個設定中，如圖二十所示，可發現以下重要結果：

(1) 與單步預測一致，加入趨勢項及季節性虛擬變數的 AR (1) 模型，在樣本內的預測表現與在樣本外的預測表現比其他模型佳；

(2) SVM 其他三個設定，在捕捉樣本內的 1970 年代的能源危機和 2001 年科技類股泡沫化之表現較弱，預測值在樣本外期間偏離真實值較多。

其次，根據圖二十一與圖二十二，autoML multistep:Recursive 模型在樣本內與樣本外之表現，與 LSTM multistep:Recursive 預測值與經濟成長率真實值走勢有些許落差，然而前者對樣本外預測有較好的掌握。相較於單步預測，autoML 多步動態遞迴預測在捕捉樣本內 2008 年金融海嘯期間，及樣本外的 2016 第四季後表現更佳。LSTM 在捕捉樣本內 1970 年代能源危機，和 2008 年金融海嘯期間表現比 autoML 更佳，但是在樣本內其他期間和樣本外期間預測表現皆出現較大的偏離。

綜合上述，如果預測單位需要執行臺灣經濟成長率的多步動態遞迴預測，使用機器學習模型是較好的選擇，此結果呼應過去文獻對機器學習方法的肯定 (Baek and Kim, 2018; Beyca *et al*., 2019; Bolhuis and Rayner, 2020; Chakraborty and Joseph, 2017; Chatzis *et al*., 2018; Kong *et al*., 2017; Richardson and Mulder, 2018; Torres *et al*., 2018; Zhao *et al*., 2017a)。三種方法以 SVM 最佳，autoML 次之，最後是 LSTM。

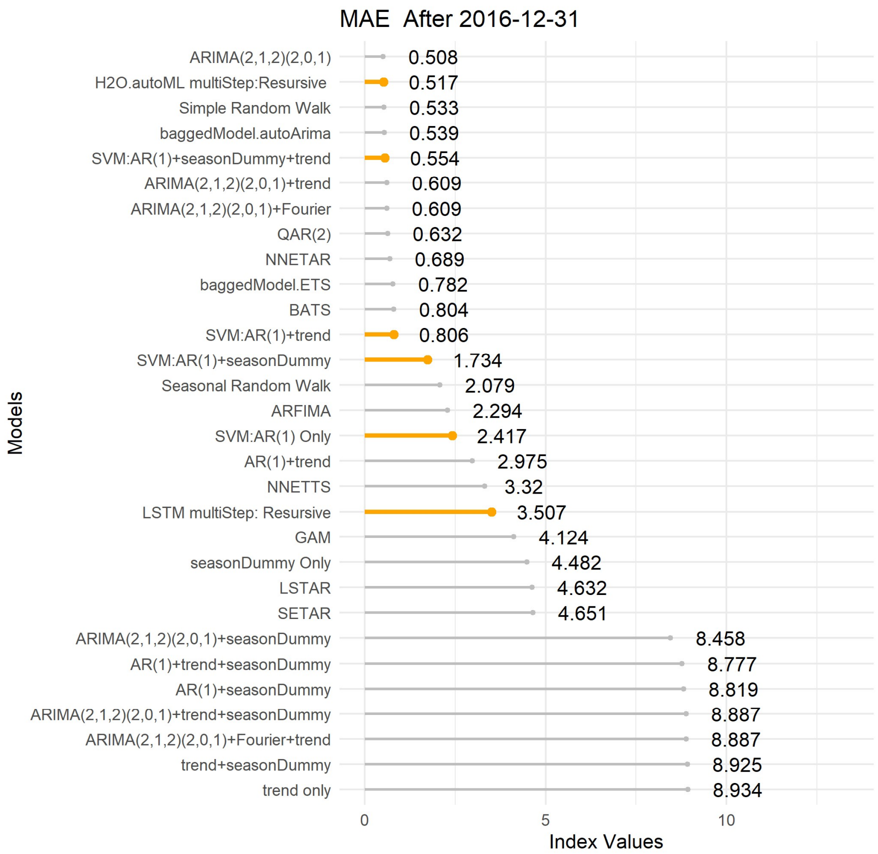
本研究結果亦發現在單步預測下，傳統時間序列模型 ARIMA 和 bagged 模型表現很好，但多步預測時機器學習方法即優於時間序列方法。可能的解釋原因，是多步預測運用前期預測值進行未來預測，而機器學習的優勢在於透過大量訓練找出最適參數，訓練過程可減少前期預測誤差，呈現較佳的預測績效。單步預測運用前期真實值進行未來預測，時間序列方法因估計參數較少且自由度高，故能得到較佳的預測績效。

資料來源：本研究自行整理。

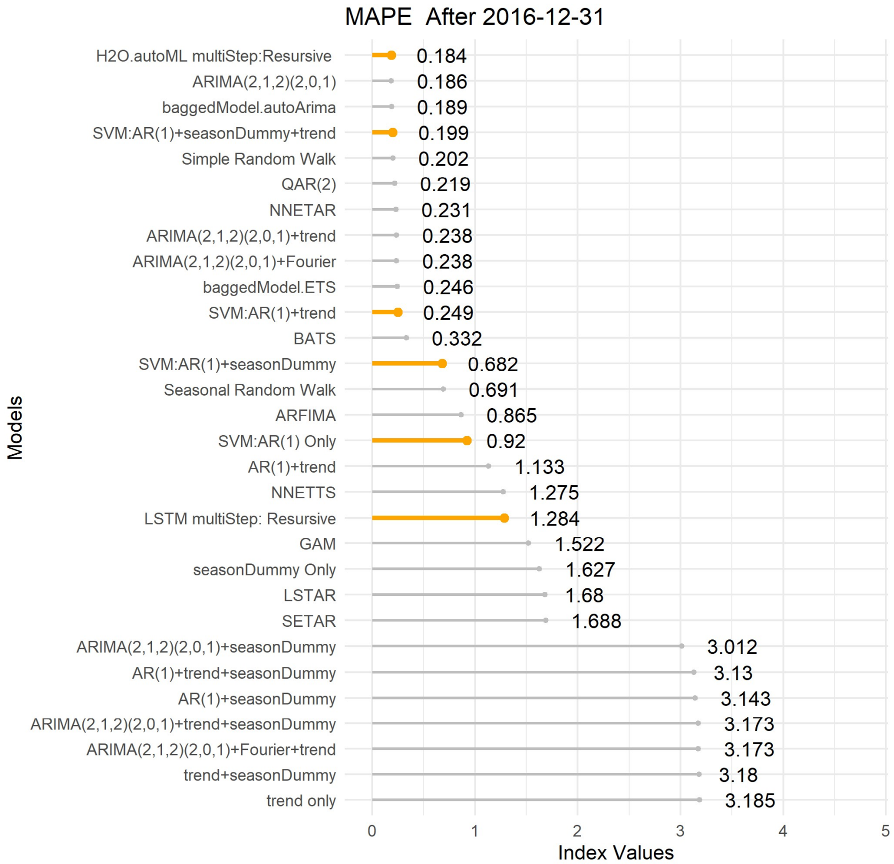
表三、多步 **Multi-step** 動態遞迴預測績效

	RMSE	MAE	MAPE	ACF1	Theil's U
Simple Random Walk	0.624	0.533	0.202	0.511	0.012
Seasonal Random Walk	2.488	2.079	0.691	0.067	0.034
trend only	8.952	8.935	3.185	0.430	0.157
seasonDummy Only	4.528	4.482	1.627	0.476	0.082
trend+seasonDummy	8.944	8.925	3.181	0.389	0.157
AR(1)+trend	3.275	2.975	1.133	0.681	0.065
AR(1)+seasonDummy	8.837	8.819	3.143	0.427	0.155
AR(1)+trend+seasonDummy	8.797	8.777	3.130	0.385	0.155
ARIMA(2,1,2)(2,0,1)	0.585	0.508	0.186	0.492	0.011
ARIMA(2,1,2)(2,0,1)+trend	0.783	0.609	0.238	0.473	0.014
ARIMA(2,1,2)(2,0,1)+seasonDummy	8.475	8.458	3.012	0.442	0.148
ARIMA(2,1,2)(2,0,1)+trend+seasonDummy	8.911	8.887	3.173	0.389	0.157
ARIMA(2,1,2)(2,0,1)+Fourier	0.783	0.609	0.238	0.473	0.014
ARIMA(2,1,2)(2,0,1)+Fourier+trend	8.912	8.887	3.173	0.389	0.157
baggedModel.ETS	0.897	0.783	0.246	0.511	0.013
baggedModel.autoArima	0.606	0.539	0.189	0.432	0.010
NNETAR	0.799	0.689	0.231	0.569	0.012
BATS	1.015	0.804	0.332	0.579	0.021
ARFIMA	2.433	2.294	0.865	0.396	0.047
LSTAR	4.740	4.632	1.680	0.215	0.086
SETAR	4.757	4.651	1.689	0.236	0.086
GAM	4.286	4.124	1.522	0.407	0.081
NNETTS	3.725	3.320	1.275	0.732	0.075
QAR(2)	0.723	0.632	0.219	0.524	0.011
SVM:AR(1)+seasonDummy	2.001	1.734	0.683	0.665	0.041
SVM:AR(1)+trend	0.927	0.806	0.249	0.437	0.014
SVM:AR(1)+seasonDummy+trend	0.657	0.554	0.199	0.232	0.011
SVM:AR(1) Only	2.609	2.417	0.920	0.563	0.051
autoML multiStep:Resursive	0.607	0.517	0.185	0.274	0.010
LSTM multiStep: Resursive	3.563	3.507	1.285	0.511	0.066

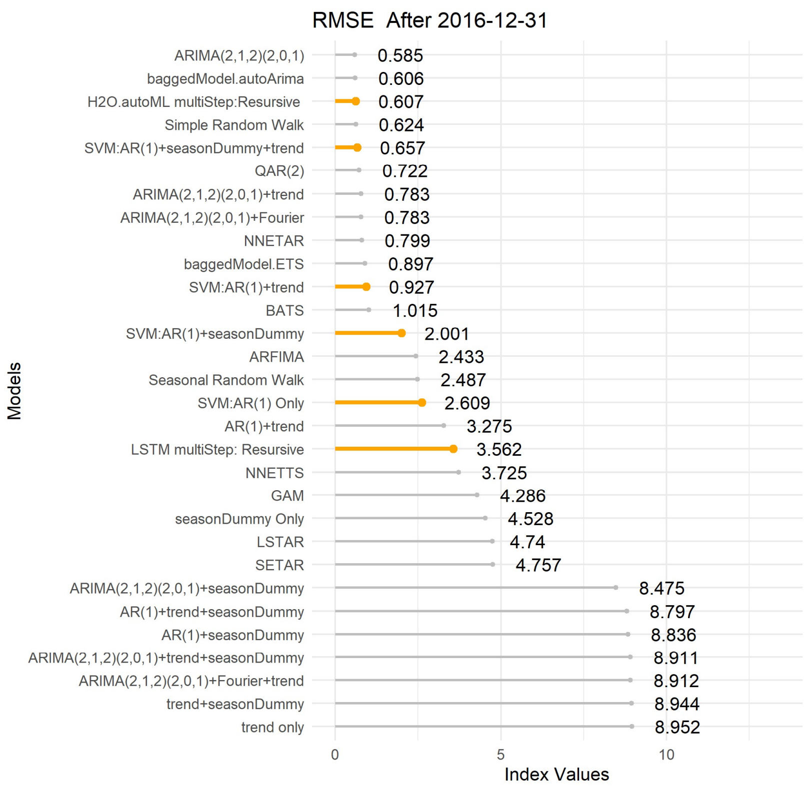
資料來源：本研究自行整理。


MAE After 2016-12-31
Models
Index Values
ARIMA(2,1,2)(2,0,1) 0.508
H2O.autoML multiStep:Resursive 0.517
Simple Random Walk 0.533
baggedModel.autoArima 0.539
SVM:AR(1)+seasonDummy+trend 0.554
ARIMA(2,1,2)(2,0,1)+trend 0.609
ARIMA(2,1,2)(2,0,1)+Fourier 0.609
QAR(2) 0.632
NNETAR 0.689
baggedModel.ETS 0.782
BATS 0.804
SVM:AR(1)+trend 0.806
SVM:AR(1)+seasonDummy 1.734
Seasonal Random Walk 2.079
ARFIMA 2.294
SVM:AR(1) Only 2.417
AR(1)+trend 2.975
NNETTS 3.32
LSTM multiStep: Resursive 3.507
GAM 4.124
seasonDummy Only 4.482
LSTAR 4.632
SETAR 4.651
ARIMA(2,1,2)(2,0,1)+seasonDummy 8.458
AR(1)+trend+seasonDummy 8.777
AR(1)+seasonDummy 8.819
ARIMA(2,1,2)(2,0,1)+trend+seasonDummy 8.887
ARIMA(2,1,2)(2,0,1)+Fourier+trend 8.887
trend+seasonDummy 8.925
trend only 8.934
0
5
10

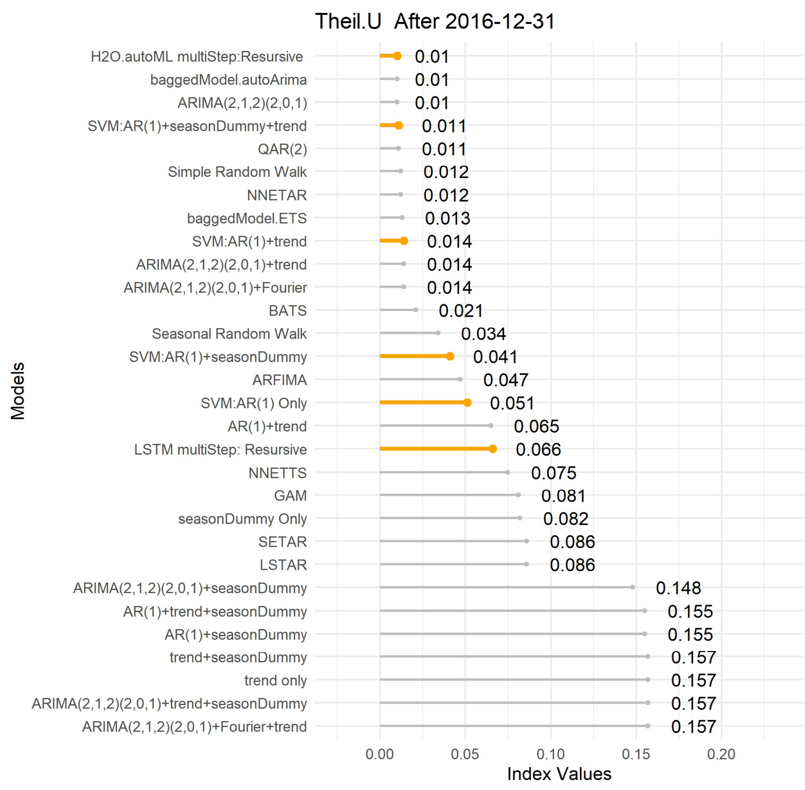
(A)


MAPE After 2016-12-31
Models
Index Values
H2O.autoML multiStep:Resursive 0.184
ARIMA(2,1,2)(2,0,1) 0.186
baggedModel.autoArima 0.189
SVM:AR(1)+seasonDummy+trend 0.199
Simple Random Walk 0.202
QAR(2) 0.219
NNETAR 0.231
ARIMA(2,1,2)(2,0,1)+trend 0.238
ARIMA(2,1,2)(2,0,1)+Fourier 0.238
baggedModel.ETS 0.246
SVM:AR(1)+trend 0.249
BATS 0.332
SVM:AR(1)+seasonDummy 0.682
Seasonal Random Walk 0.691
ARFIMA 0.865
SVM:AR(1) Only 0.92
AR(1)+trend 1.133
NNETTS 1.275
LSTM multiStep: Resursive 1.284
GAM 1.522
seasonDummy Only 1.627
LSTAR 1.68
SETAR 1.688
ARIMA(2,1,2)(2,0,1)+seasonDummy 3.012
AR(1)+trend+seasonDummy 3.13
AR(1)+seasonDummy 3.143
ARIMA(2,1,2)(2,0,1)+trend+seasonDummy 3.173
ARIMA(2,1,2)(2,0,1)+Fourier+trend 3.173
trend+seasonDummy 3.18
trend only 3.185
0
1
2
3
4
5

(B)



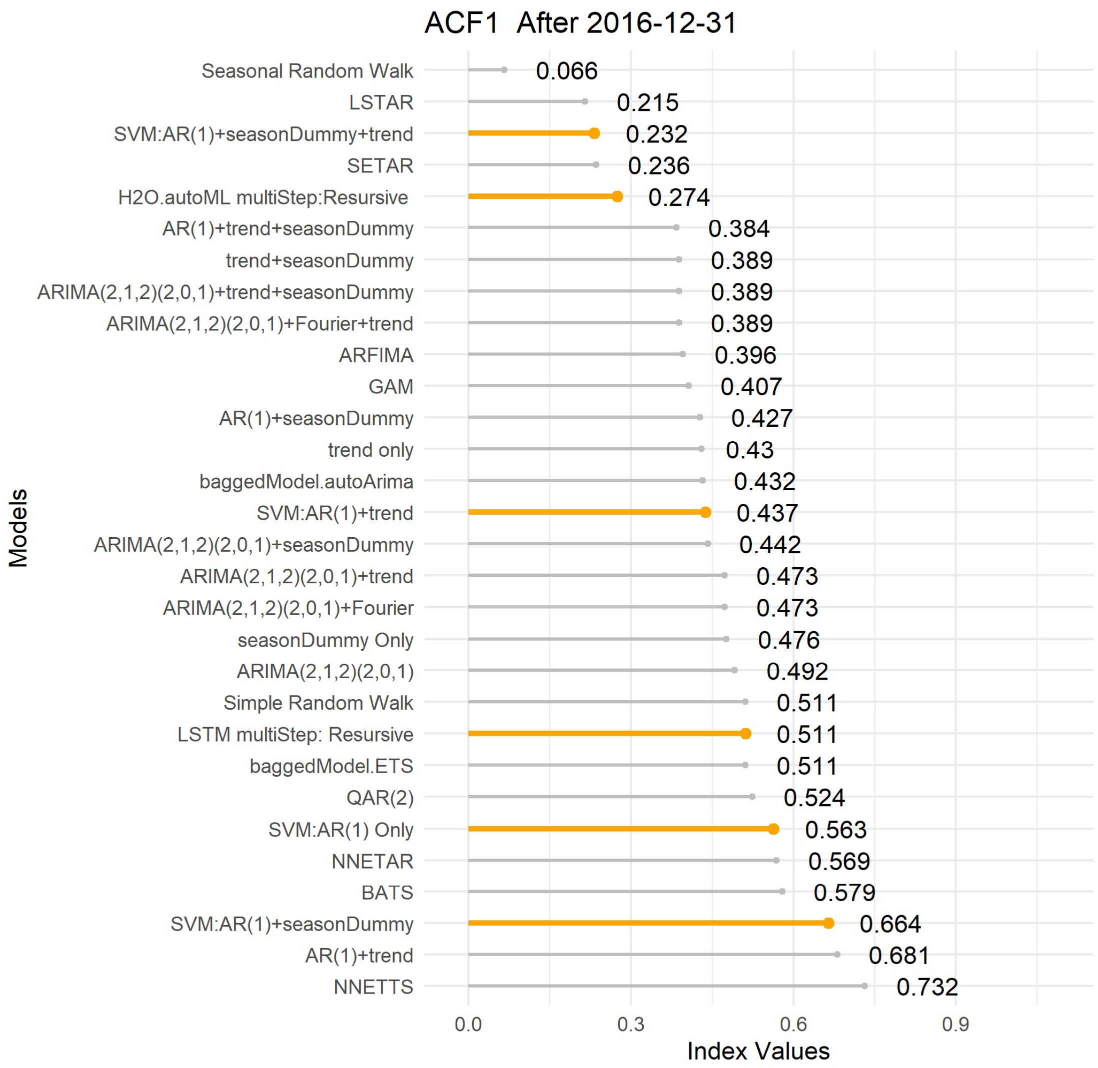
(C)



(D)

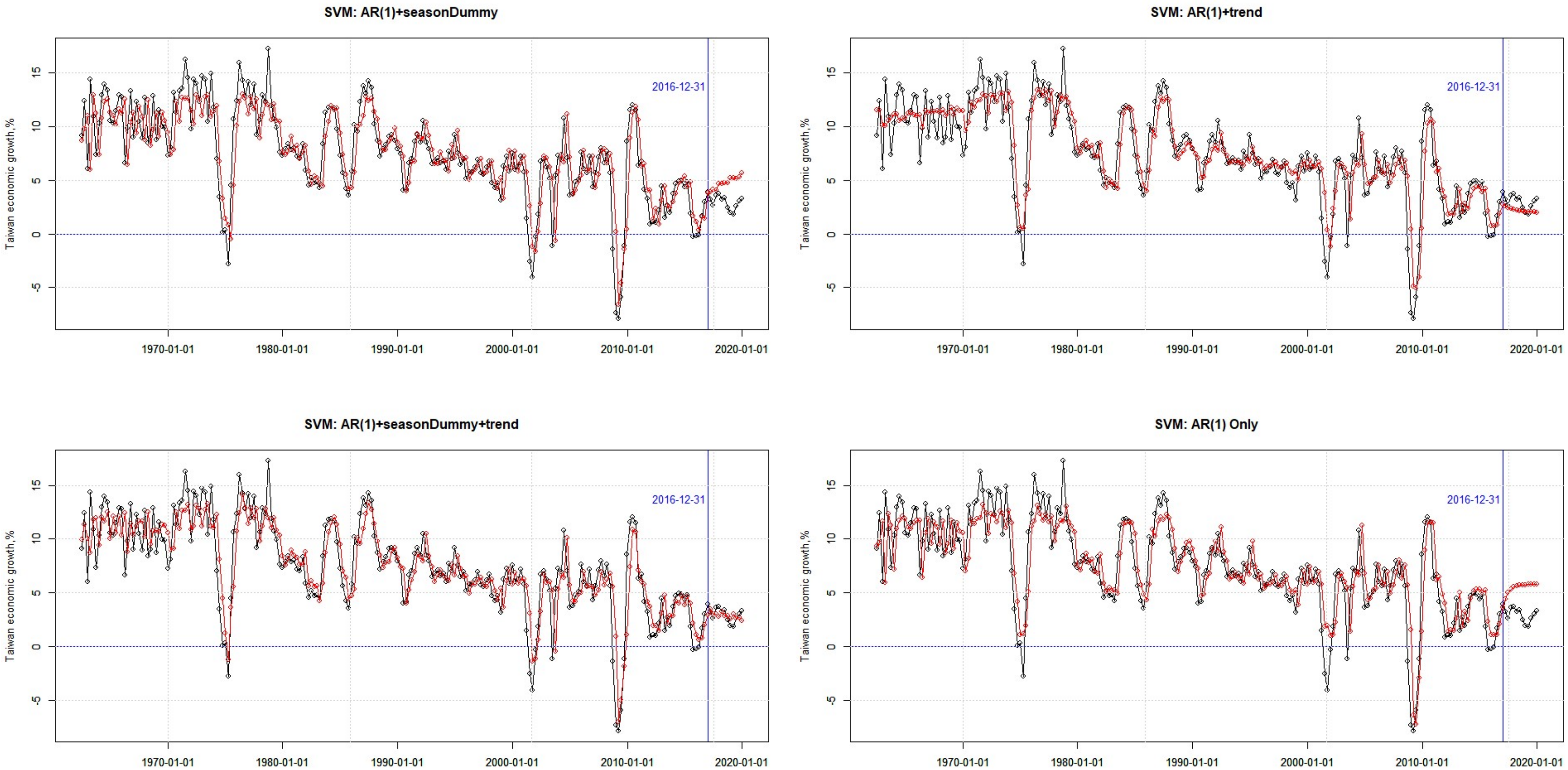
資料來源：本研究自行整理。

圖十八、多步動態遞迴預測的整體時間序列表現：**4** 個指標



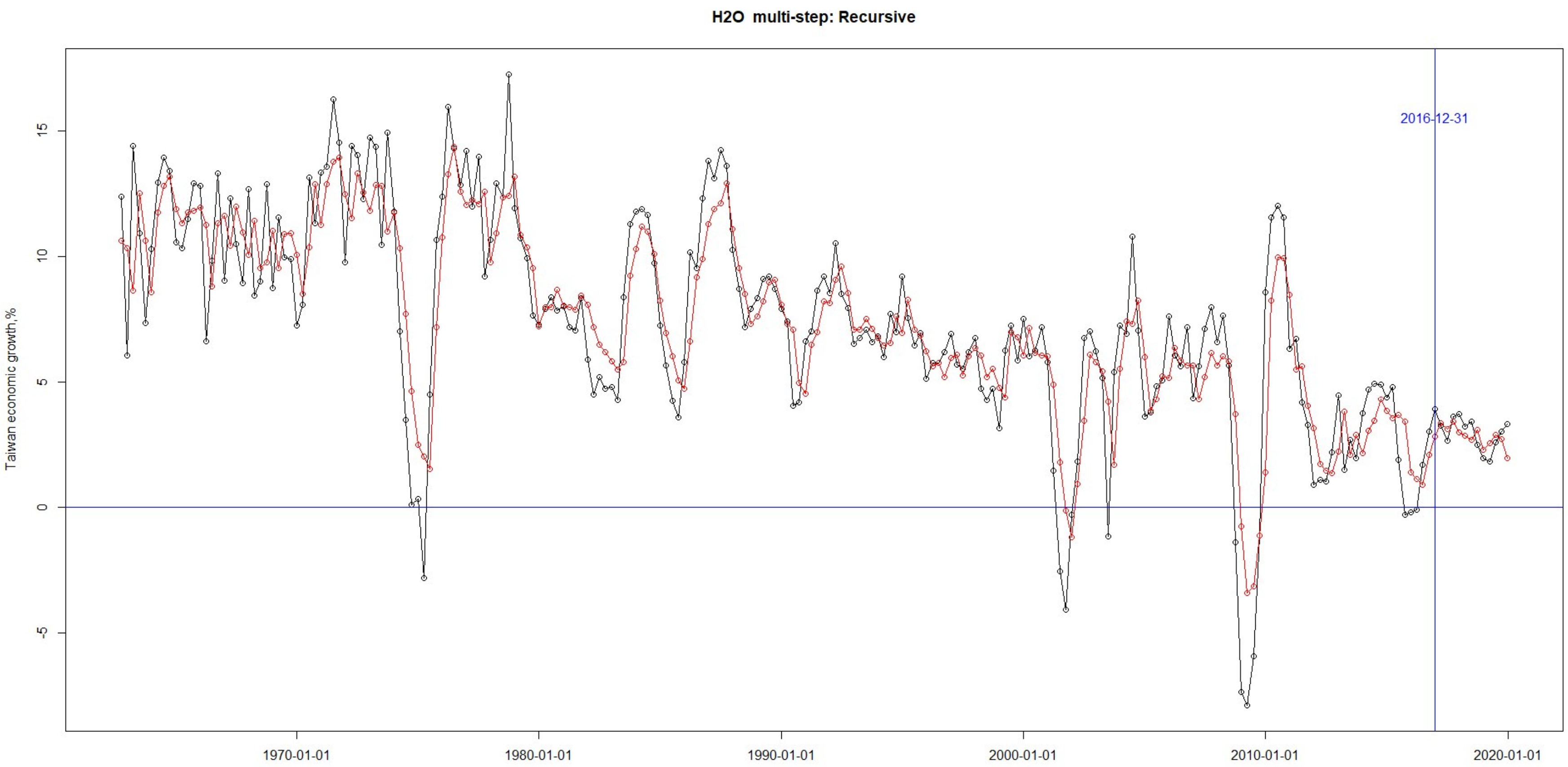
資料來源：本研究自行整理。

圖十九、多步動態遞迴預測正確性：誤差的 **ACF(1)** 值



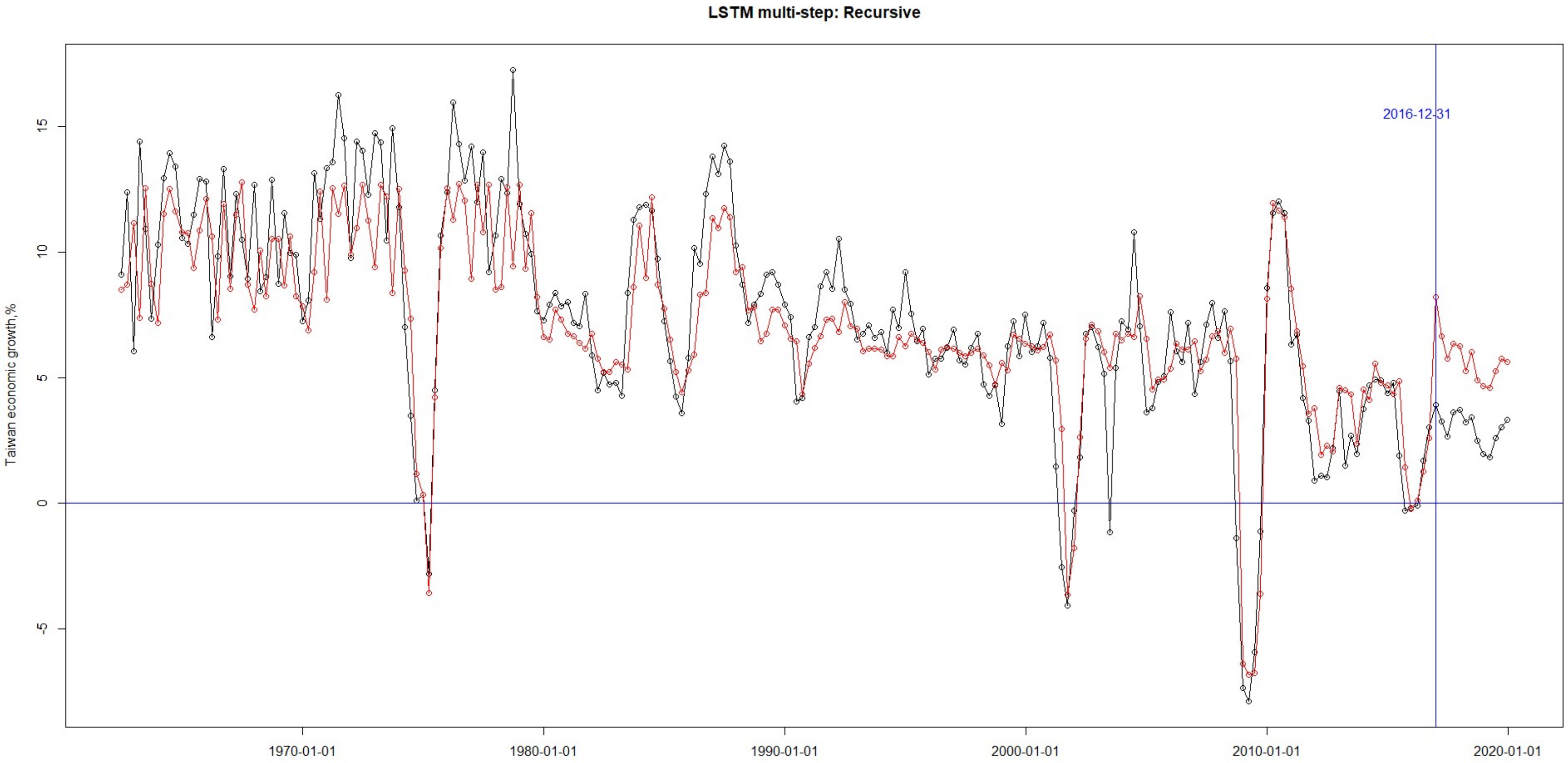
資料來源：本研究自行整理。

圖二十、多步動態遞迴預測的整體時間序列表現：**SVM 4** 個設定



資料來源：本研究自行整理。

圖二十一、多步動態遞迴預測的整體時間序列表現：**autoML**



資料來源：本研究自行整理。

圖二十二、多步動態遞迴預測的整體時間序列表現：**LSTM**

6 混合落後結構和添加其他變數

我們將之前的單變量 AR(1) 模式延伸為兩種：混合落後結構與添加其他變數。令 F 代表機器學習的演算模式，其投入變數 (input variables) 的綜合說明如下：

對被解釋變 (y) 的連續落後，我們沿用 AR 寫法 $\mathbf{L}^{p}$

$$y_t = F(\mathbf{L}^p(y_t)) = F(y_{t-1}, y_{t-2}, \ldots y_{t-p}) + e_t,$$

對解釋變數 $\mathbf{X}$ 的連續落後，我們用 $\mathbf{L}^{h}$

$$y_t = F(\mathbf{L}^h(X_t)) = F(X_{t-1}, X_{t-2}, \ldots X_{t-h}) + e_t,$$

對非連續落後，我們用 k

$$\begin{aligned} & y_t = F(\mathbf{y}_{t-k_1}; \mathbf{x}_{t\text{-}k_2}) + e_t \\ & k_1, k_2 \in \{(1,2),(1,3),(1,4),(2,3),\cdots(1,2,3,4)\} \end{aligned},$$

此處投入變數範例如下

$$\begin{aligned} & y_t = a + F(\mathbf{y}_{t-1}, \mathbf{y}_{t-4}; X_{t\text{-}2}, X_{t\text{-}3}) + e_t \\ & k_1 = (1,4), k_2 = (2,3) \end{aligned},$$

其他變數 $\mathbf{X}$ 中，採用和經濟成長率期間相同的總經數據，詳細說明請見 6.2 節。

6.1 自我相關落後模型

本節首先以自我相關落後模型進行 2016 年第四季後多步預測的表現評估，此例模型設定可以表示如下

$$\mathrm{Y_t} = F\left(\mathbf{L^k}(\mathrm{y_t})\right),$$

$$\mathbf{k} \in \{(1),(2),(3),(4),(1,2),(1,3),(1,4),(2,3),\dots,(1,2,3),(1,2,4),\dots,(1,2,3,4)\}.$$

此處投入變數範例如下：

$$\mathbf{k} = (1),\ \ \mathrm{y_t} = F(y_{t-1}),$$

$$\mathbf{k} = (2),\ \ \mathrm{y_t} = F(y_{t-2}),$$

$$\mathbf{k} = (2,3),\ \ \mathrm{y_t} = F(y_{t-2}, y_{t-3}),$$

$$\mathbf{k} = (1,2,3),\ \ \mathrm{y_t} = F(y_{t-1}, y_{t-2}, y_{t-3}),$$

$$\mathbf{k} = (1,2,4),\ \ \mathrm{y_t} = F(y_{t-1}, y_{t-2}, y_{t-4}).$$

其中當 k 的元素為連續正整數時，模型即是 AR 模型。由前述章節已知三類機器學習模型 (autoML, SVM, LSTM) 預測績效結果，並按優劣排列如圖二十三。整體而言機器學習模型的預測誤差都相當低，各類機器學習模型均有非常優異的預測表現。從 MAE、MAPE、RMSE、Theil's U 指標均發現，autoML 均具有最佳表現，SVM 其次，LSTM 第三。此外，此圖二十三也發現，當連續落後期模型改為混合落後期時，預測績效均可再改善。例如當連續落後模型 k=(1,2,3) 縮減為 k=(2,3) 或 k= (1,2) 時，autoML 模型的 MAPE 從 0.1822 降低至 0.1483 及 0.1239；Theil's U 由 0.01 降低至 0.0086 及 0.0072。連續落後模型 k=(1,2,3,4) 縮減為 k=(1,2)、k= (1,4)、k= (3,4) 時，autoML 模型下的 MAPE 從 0.1761 降至 0.1239-0.15，由此可知採用混合落後期，有機會改善預測績效。

在預測誤差持續性方面，如圖二十四所示，整體而言 ACF(1) 絕對值與選用的機器學習模型無明顯關連。此外亦發現連續落後期模型(如 k=(1,2,3) 或 k=(1,2,3,4)) 在刪去 1 或 2 個落後期後，在大部分情況下將使 ACF(1) 絕對值增加，預測誤差持續性加大。

接著檢視預測的整體時間序列表現，以 autoML 模型預測連續與混合落後模型的時間序列描繪於圖二十五，以 LSTM 模型預測的描繪於圖二十六，以 SVM 模型預測的描繪於圖二十七。其中預測值在 2016 年第四季前為樣本內預測值，2016 年第四季後則為樣本外多步預測值。

首先，autoML 預測的連續落後期模型如圖二十五(A)-(C) 所示，發現以下重要結果：相較於圖十六的 AR(1) 模型，k=(1,2) 在樣本內與樣本外的預測值與真實值接近，顯示增加落後期數可提升預測表現。然而不論何種設定，捕捉樣本內 1974-1975 年與 2009 年高峰或低谷的表現均不甚佳，但增加落後期數 (k=(1,2,3) 及 k=(1,2,3,4)) 之後，捕捉高峰或低谷表現將可提升。另外，預測值的波峰或波谷發生時點會較晚於真實值。[7]

autoML 的非連續落後結果如圖二十五(D)~(K) 所示：大多數情況而言，k=(1,2,3) 及 k=(1,2,3,4) 模型在刪去 1 或 2 個落後期後，不論是樣本內的趨勢預

[7] (C)圖樣本內似乎是完全配適，但樣本外則存在落後的狀況，不只是機器學習，統計時間序列也不敢說樣本內配適度和樣本外的預測績效之間有一定相關，尤其是未來出現重大 shocks 時。然而，機器學習模型可能發生樣本內配適結果良好，但樣本外預測績效太差，遠大於樣本內的預測績效的情形，此即過度配適 (overfitting)。本研究樣本內外的預測績效差異在容許範圍內，過度配適問題並不顯著。

測，或是對高峰或低谷資料的預測能力均變差，與真實值差距變大。唯一特例是，樣本內趨勢及高峰或低谷的預測，在 k=(1,3,4) 時的表現優於 k=(1,2,3,4)。

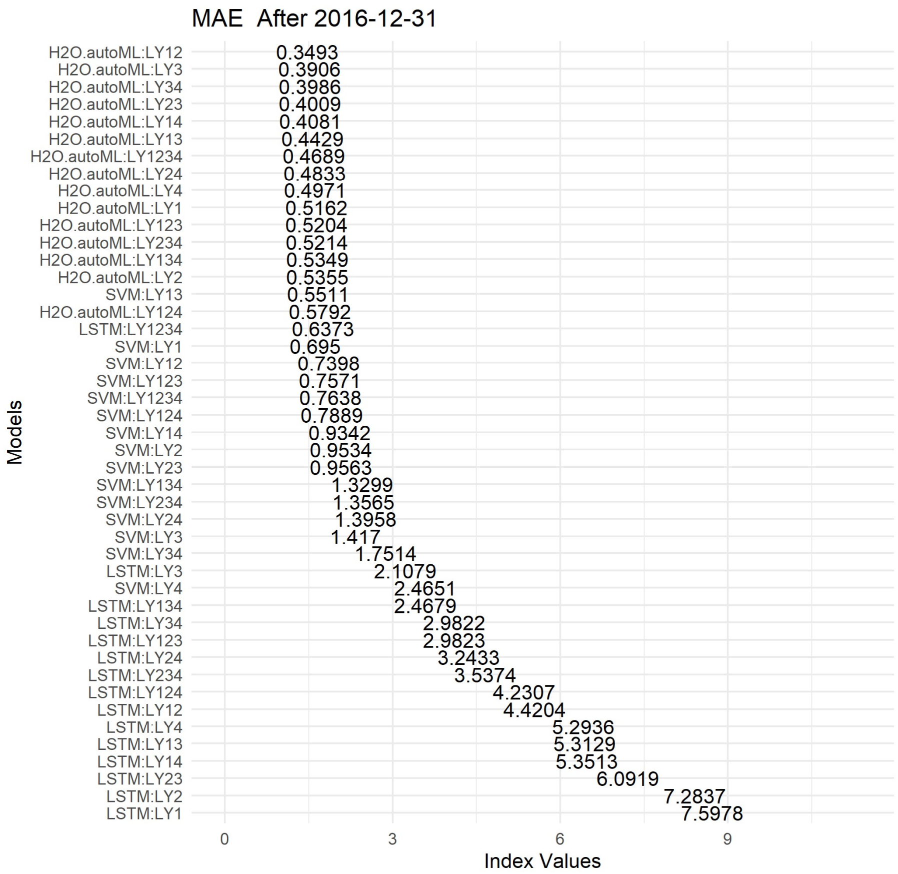
其次，LSTM 連續落後期模型整體時間序列預測如圖二十六(A)~(C) 所示，與圖十七的 AR (1) 模型相比，加入落後期數越多 (由 k=(1,2) 至 k=(1,2,3,4))，樣本內與樣本外的預測表現越良好，捕捉樣本內 1974-1975 年與 2009 年高峰或低谷的表現明顯改善。但若資料出現高頻率波動如 1970 年初，LSTM 將難以預測。

在非連續落後模型預測方面，圖二十六(D)~(K) 顯示，當連續落後期模型刪去 1 或 2 個落後期，LSTM 預測時間序列的特性更明顯。LSTM 受到歷史資料記憶影響，預測值的高點及低點往往受限於一個範圍內，如 k= (2,3)、k= (2,4)、k= (3,4)。

最後，SVM 的連續落後模型整體時間序列預測如圖二十七(A)~(C) 所示，不論在整體趨勢或捕捉樣本內高峰或低谷之表現上，均較前二種機器學習模型為佳。尤其在捕捉樣本內的 1974-1975 年與 2009 年高峰或低谷之表現，預測值已能逼近真實值。與圖十五的 AR (1) 模型相比，則發現增加落後期數後 (由 k=(1,2) 至 k=(1,2,3,4))，樣本外預測值更接近真實時間序列。SVM 在不連續落後期下，圖二十七(D)~(K) 顯示整體時間趨勢及捕捉樣本內高峰或低谷表現優 LSTM 模型，且預測值有明顯落後於真實值的情形。

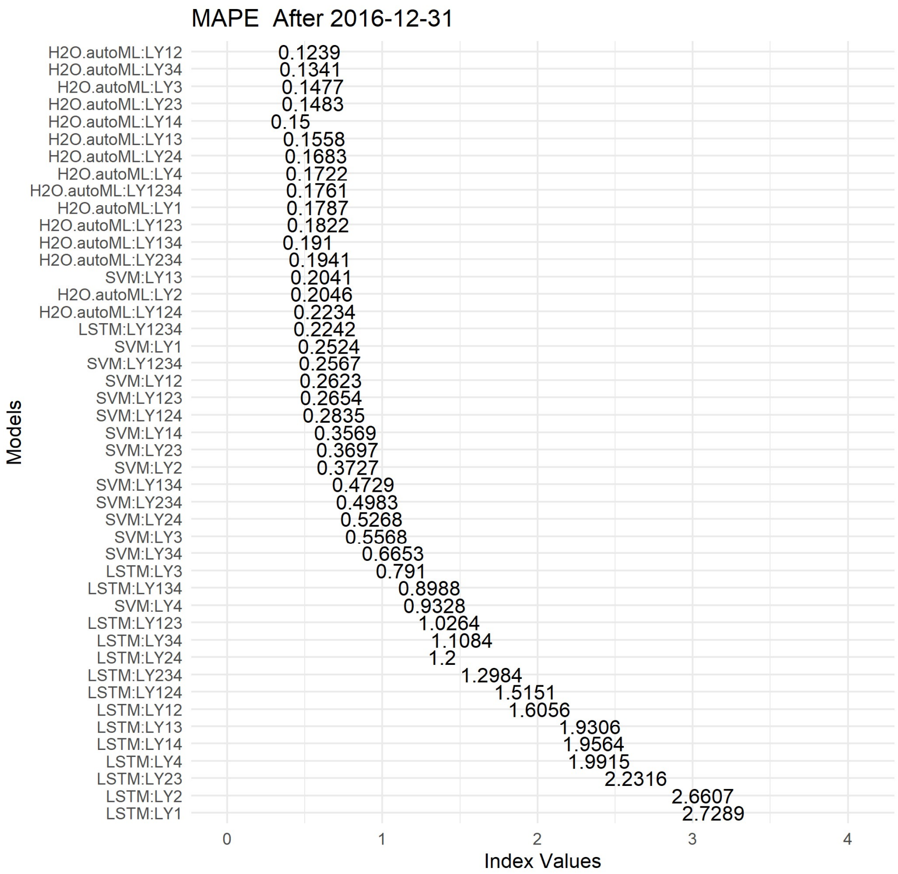
綜合上述，三個機器學習模型均有相當優異的預測績效，且以 autoML 為最佳。而在整體時間序列結果上， LSTM 難以有效捕捉樣本內高峰或低谷，autoML 在某些情形下表現良好，SVM 則相對有穩定良好表現。當改為不連續落後期模型後，有機會提升預測績效，但在對高峰或低谷資料的預測能力均變差，與真實值差距變大，並且預測值明顯有落後於真實值的情形。

非連續落後期模型在時間序列 ARMA 架構下並不常見，然而機器學習實務上旨在萃取可預測訊息，並不需要遵從 AR。本節運用非連續落後期數下的自我相關落後模型，在經濟意義解釋上，因經濟衝擊對經濟成長率的影響有時間落後，故出現非連續落後期數含有可預測資訊的狀況。事實上類似的問題在 ARIMA 也有，即處理長期記憶時間序列的 ARFIMA (F 表示 fractionally)，此情形下 ARIMA(p,d,q)差分項 d 不一定是整數。



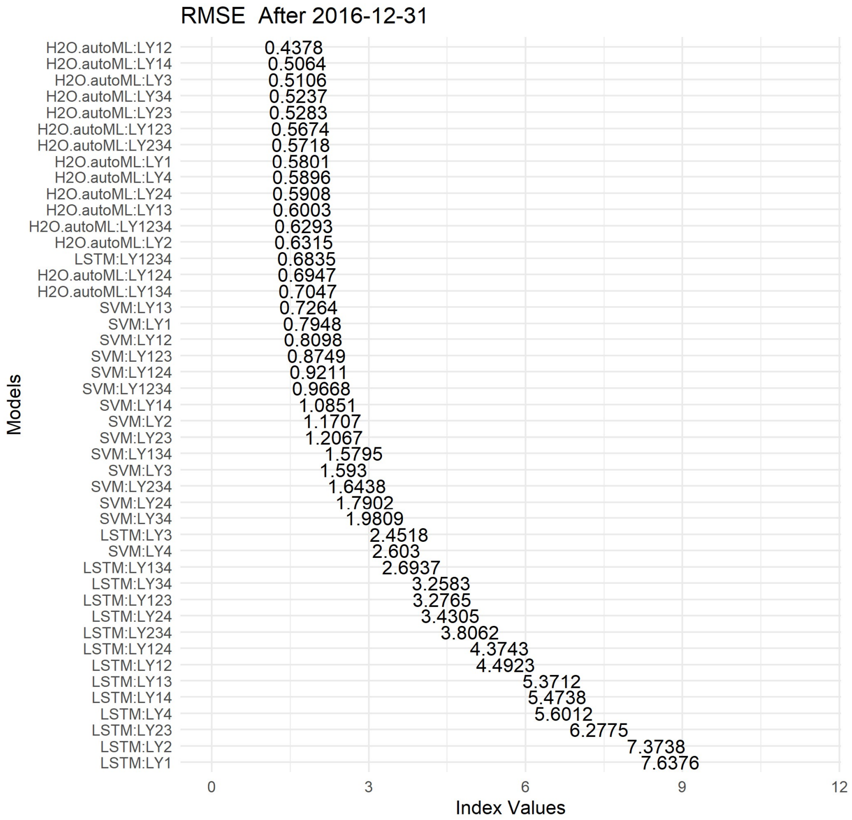
MAE After 2016-12-31
Models
Index Values
H2O.autoML:LY12 0.3493
H2O.autoML:LY3 0.3906
H2O.autoML:LY34 0.3986
H2O.autoML:LY23 0.4009
H2O.autoML:LY14 0.4081
H2O.autoML:LY13 0.4429
H2O.autoML:LY1234 0.4689
H2O.autoML:LY24 0.4833
H2O.autoML:LY4 0.4971
H2O.autoML:LY1 0.5162
H2O.autoML:LY123 0.5204
H2O.autoML:LY234 0.5214
H2O.autoML:LY134 0.5349
H2O.autoML:LY2 0.5355
SVM:LY13 0.5511
H2O.autoML:LY124 0.5792
LSTM:LY1234 0.6373
SVM:LY1 0.695
SVM:LY12 0.7398
SVM:LY123 0.7571
SVM:LY1234 0.7638
SVM:LY124 0.7889
SVM:LY14 0.9342
SVM:LY2 0.9534
SVM:LY23 0.9563
SVM:LY134 1.3299
SVM:LY234 1.3565
SVM:LY24 1.3958
SVM:LY3 1.417
SVM:LY34 1.7514
LSTM:LY3 2.1079
SVM:LY4 2.4651
LSTM:LY134 2.4679
LSTM:LY34 2.9822
LSTM:LY123 2.9823
LSTM:LY24 3.2433
LSTM:LY234 3.5374
LSTM:LY124 4.2307
LSTM:LY12 4.4204
LSTM:LY4 5.2936
LSTM:LY13 5.3129
LSTM:LY14 5.3513
LSTM:LY23 6.0919
LSTM:LY2 7.2837
LSTM:LY1 7.5978
0
3
6
9

(A)

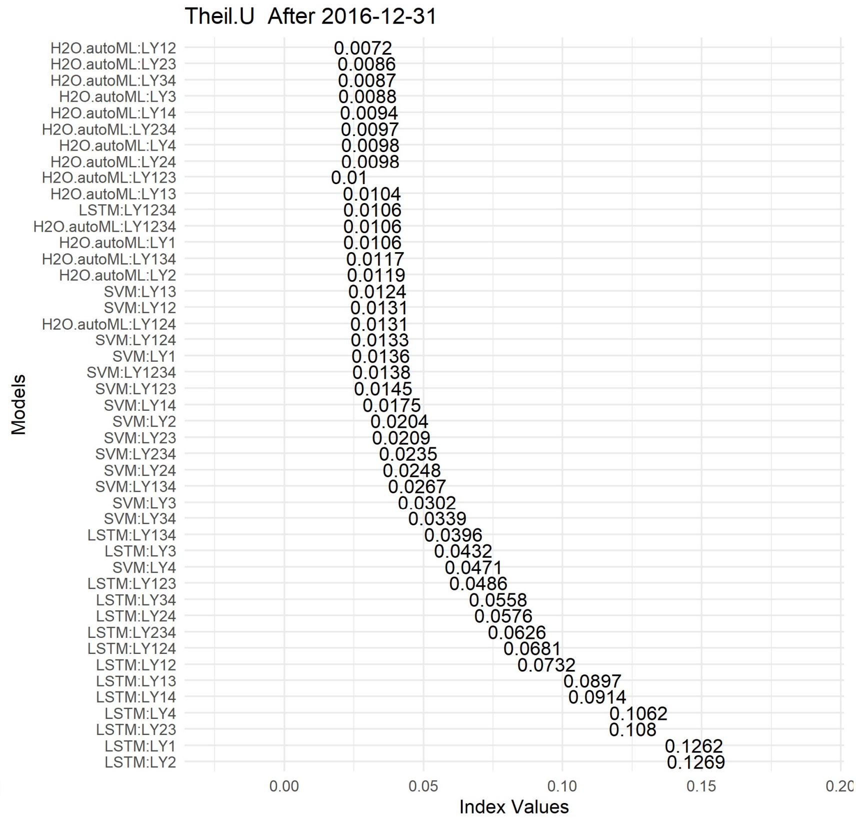


MAPE After 2016-12-31
Models
Index Values
H2O.autoML:LY12 0.1239
H2O.autoML:LY34 0.1341
H2O.autoML:LY3 0.1477
H2O.autoML:LY23 0.1483
H2O.autoML:LY14 0.15
H2O.autoML:LY13 0.1558
H2O.autoML:LY24 0.1683
H2O.autoML:LY4 0.1722
H2O.autoML:LY1234 0.1761
H2O.autoML:LY1 0.1787
H2O.autoML:LY123 0.1822
H2O.autoML:LY134 0.191
H2O.autoML:LY234 0.1941
SVM:LY13 0.2041
H2O.autoML:LY2 0.2046
H2O.autoML:LY124 0.2234
LSTM:LY1234 0.2242
SVM:LY1 0.2524
SVM:LY1234 0.2567
SVM:LY12 0.2623
SVM:LY123 0.2654
SVM:LY124 0.2835
SVM:LY14 0.3569
SVM:LY23 0.3697
SVM:LY2 0.3727
SVM:LY134 0.4729
SVM:LY234 0.4983
SVM:LY24 0.5268
SVM:LY3 0.5568
SVM:LY34 0.6653
LSTM:LY3 0.791
LSTM:LY134 0.8988
SVM:LY4 0.9328
LSTM:LY123 1.0264
LSTM:LY34 1.1084
LSTM:LY24 1.2
LSTM:LY234 1.2984
LSTM:LY124 1.5151
LSTM:LY12 1.6056
LSTM:LY13 1.9306
LSTM:LY14 1.9564
LSTM:LY4 1.9915
LSTM:LY23 2.2316
LSTM:LY2 2.6607
LSTM:LY1 2.7289
0
1
2
3
4

(B)



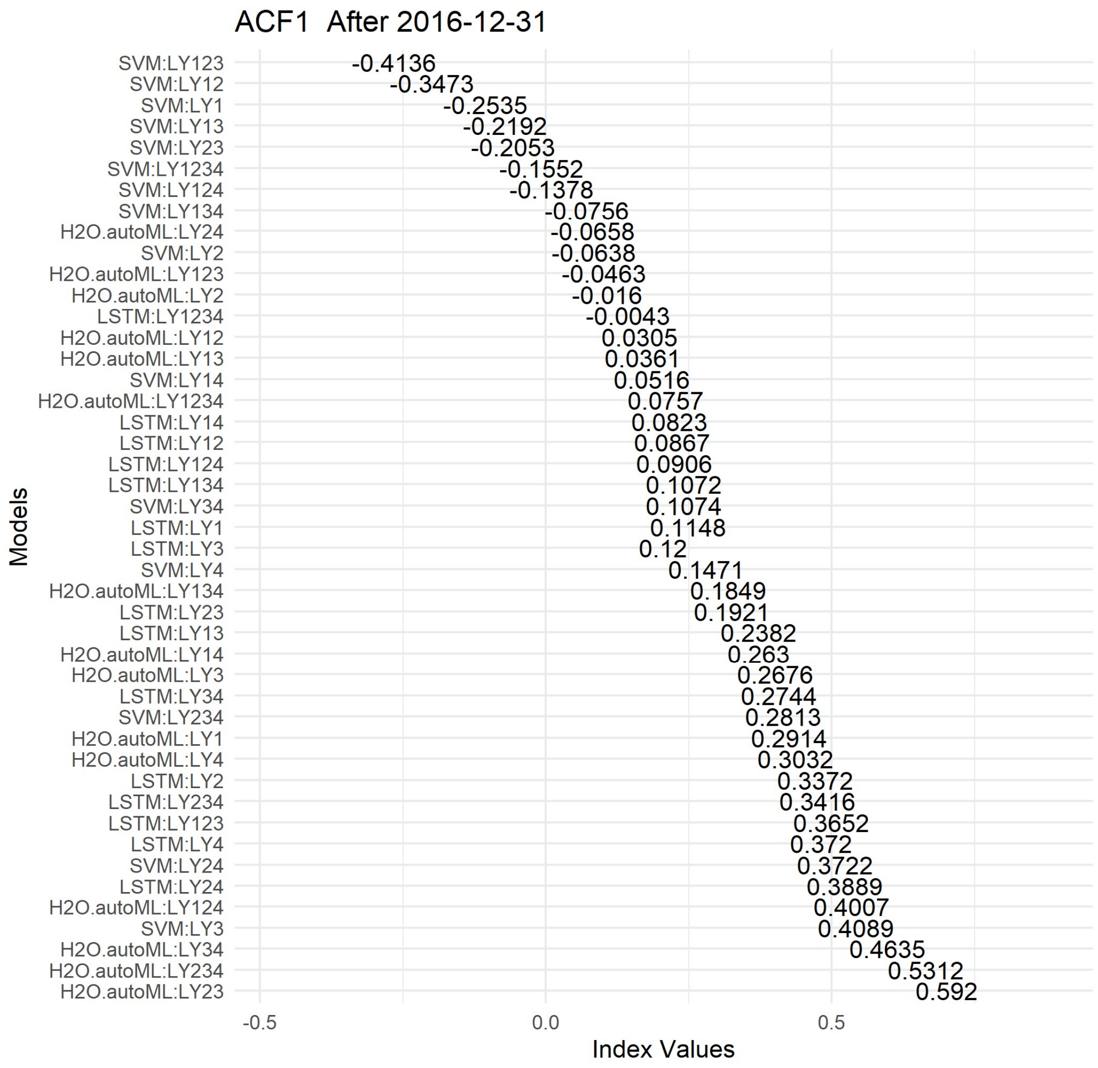
(C)



(D)

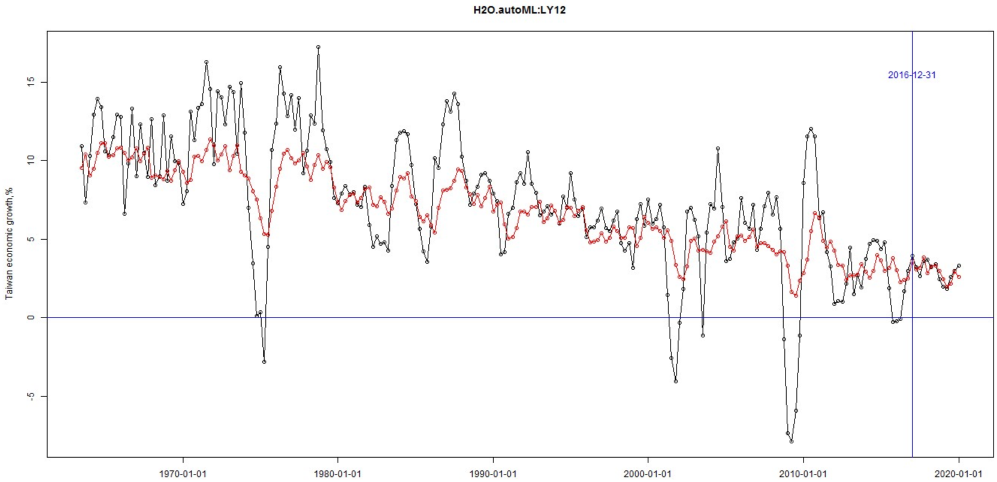
資料來源：本研究自行整理。

圖二十三、自我相關落後模型的 **4** 個預測精確度指標

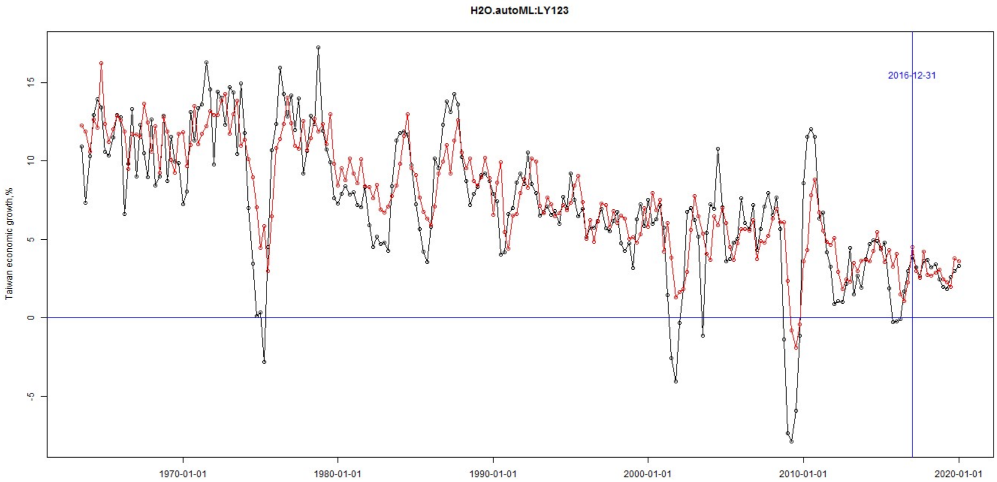


資料來源：本研究自行整理。

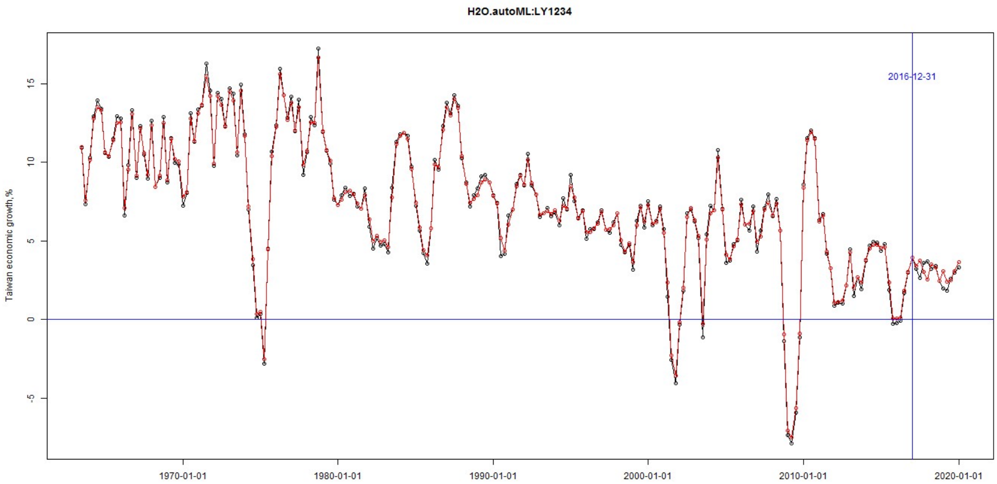
圖二十四、多步預測正確性：**ACF(1)** 值


H2O.autoML:LY12
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

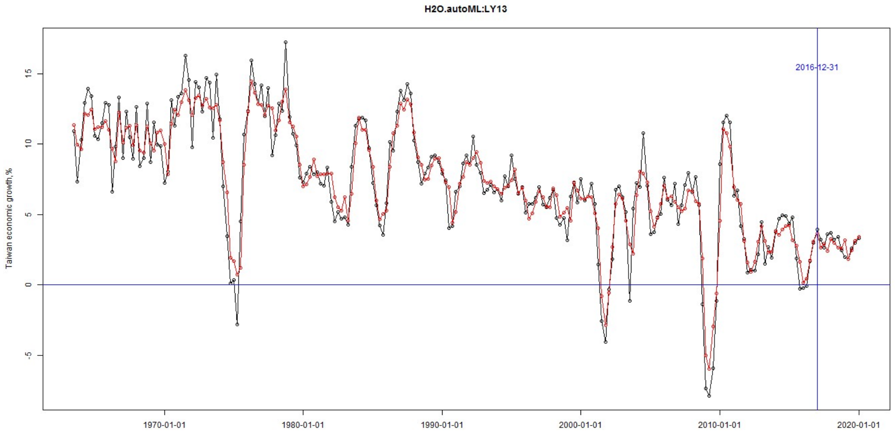
(A)


H2O.autoML:LY123
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

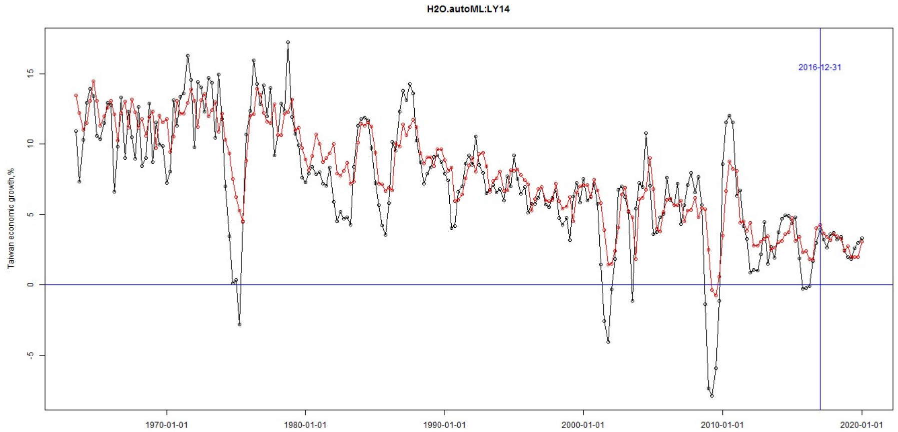
(B)


H2O.autoML:LY1234
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

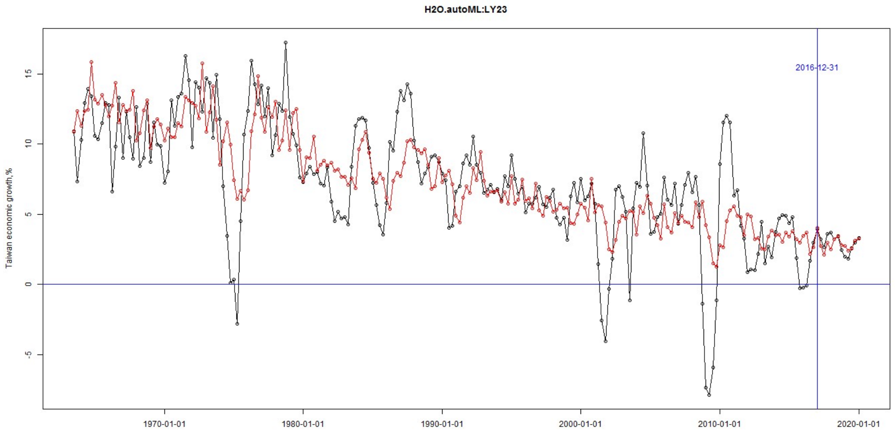
(C)


H2O.autoML:LY13
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

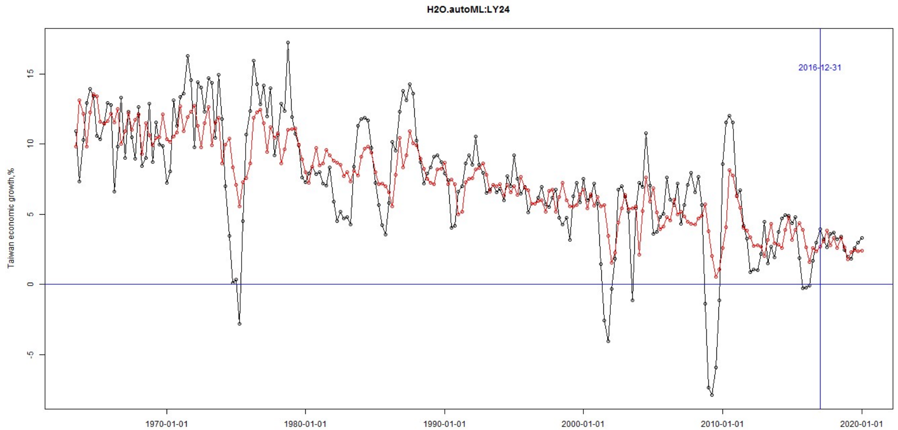
(D)


H2O.autoML:LY14
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

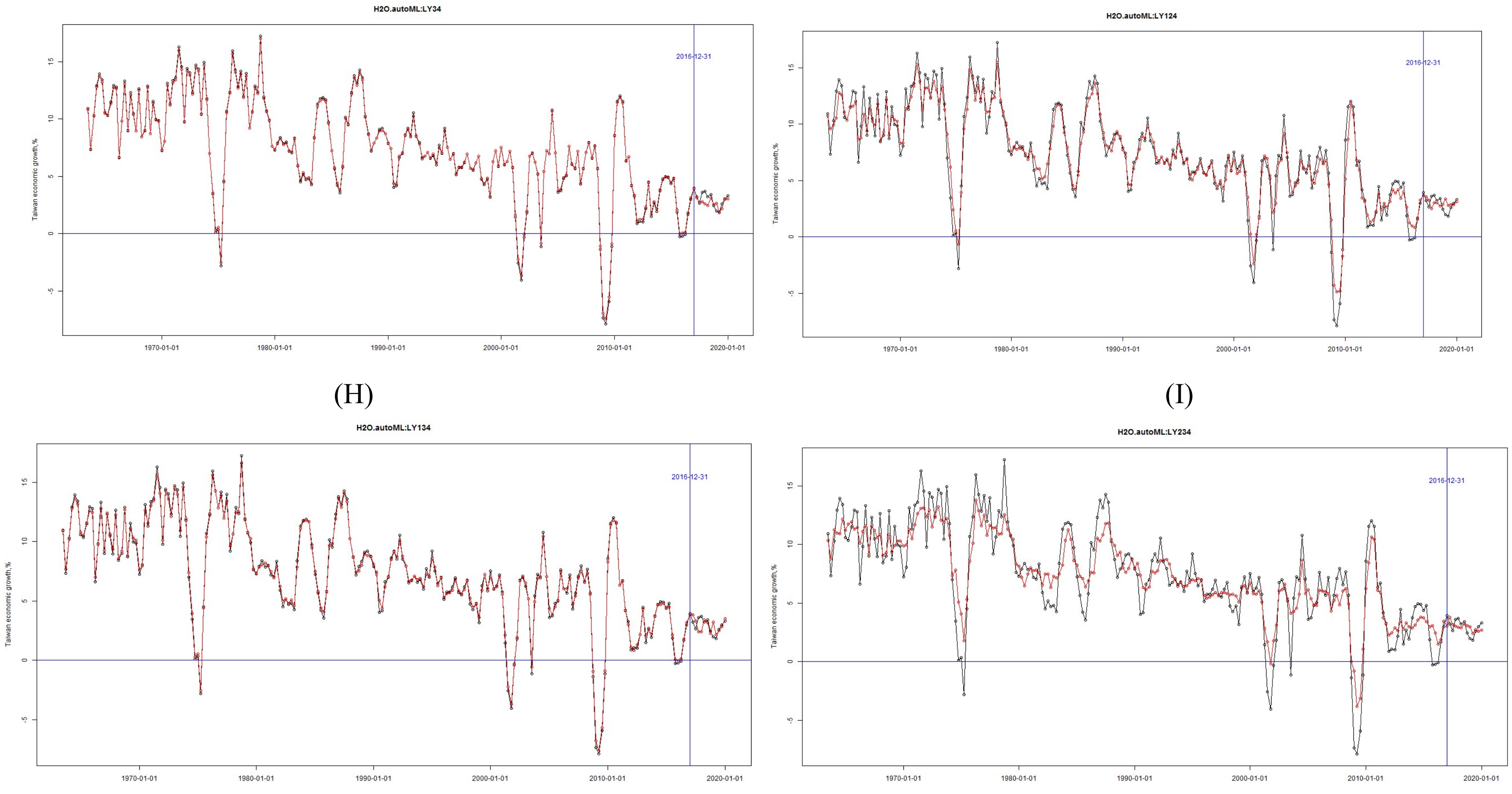
(E)


H2O.autoML:LY23
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

(F)


H2O.autoML:LY24
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

(G)

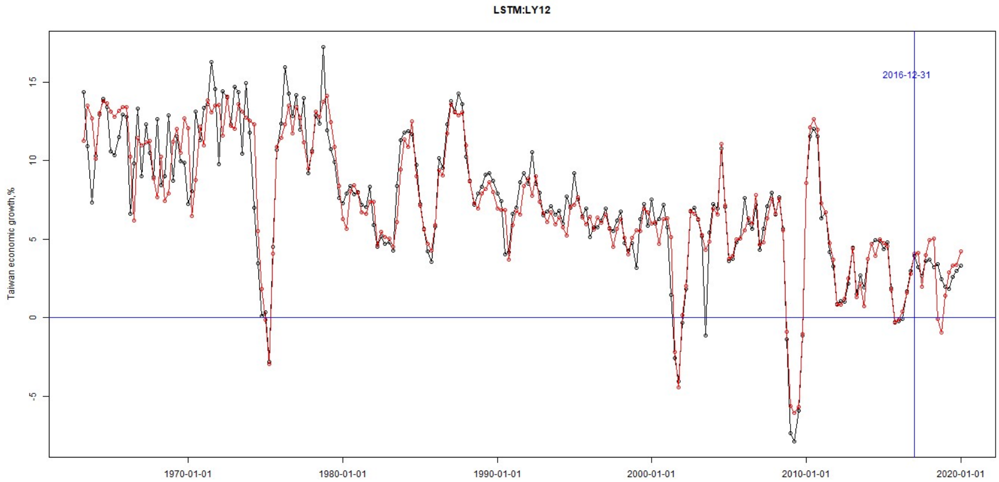


(H) (I)

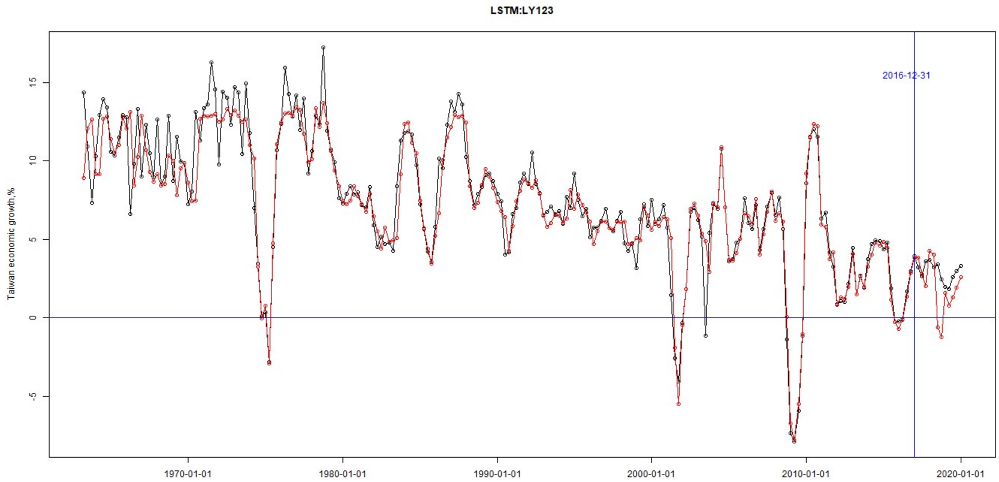
(J) (K)

資料來源：本研究自行整理。

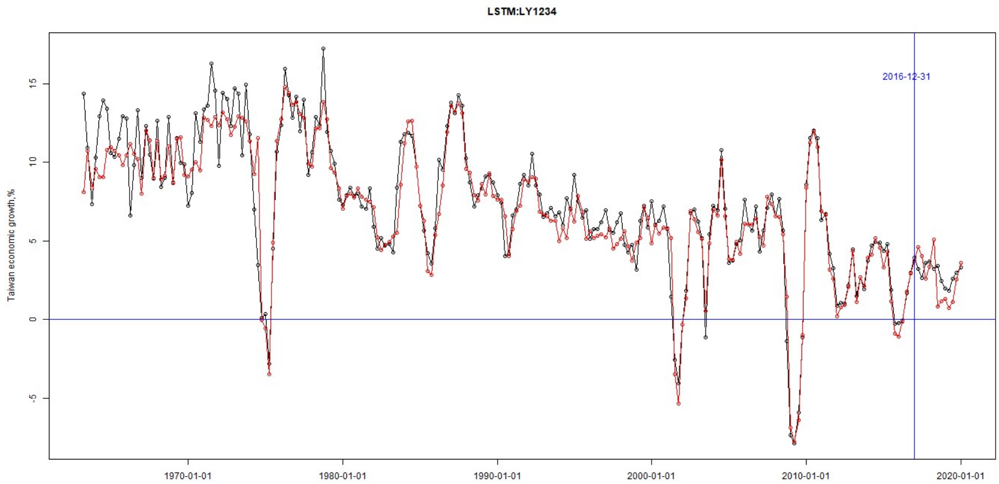
圖二十五、自我相關落後期模型：**autoML**


LSTM:LY12
2016-12-31
Taiwan economic growth,%
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

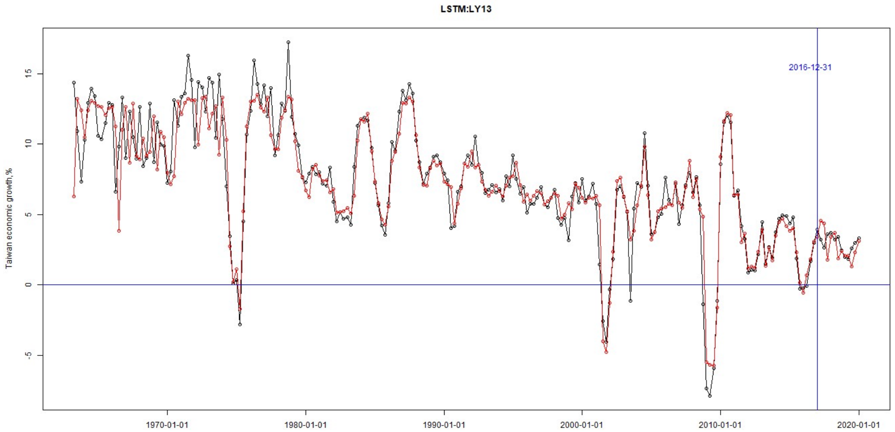
(A)


LSTM:LY123
2016-12-31
Taiwan economic growth,%
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

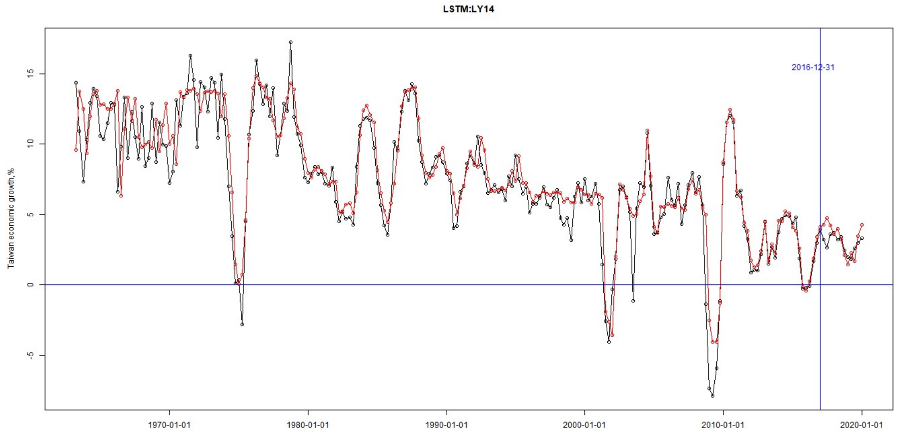
(B)


LSTM:LY1234
2016-12-31
Taiwan economic growth,%
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

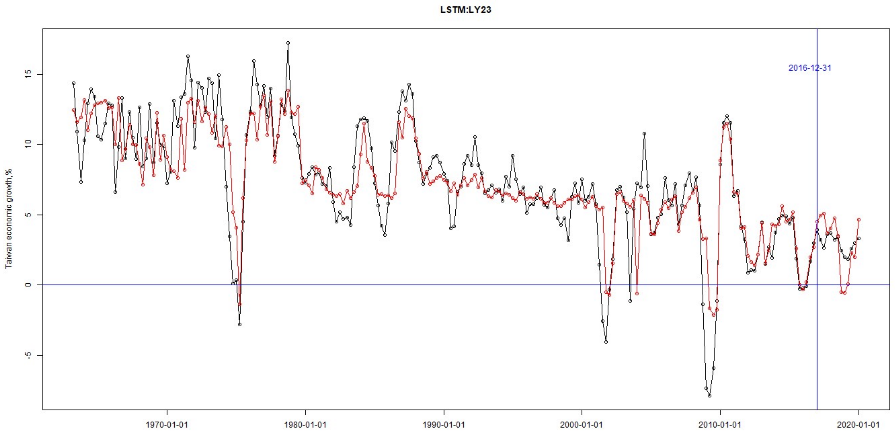
(C)


LSTM:LY13
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

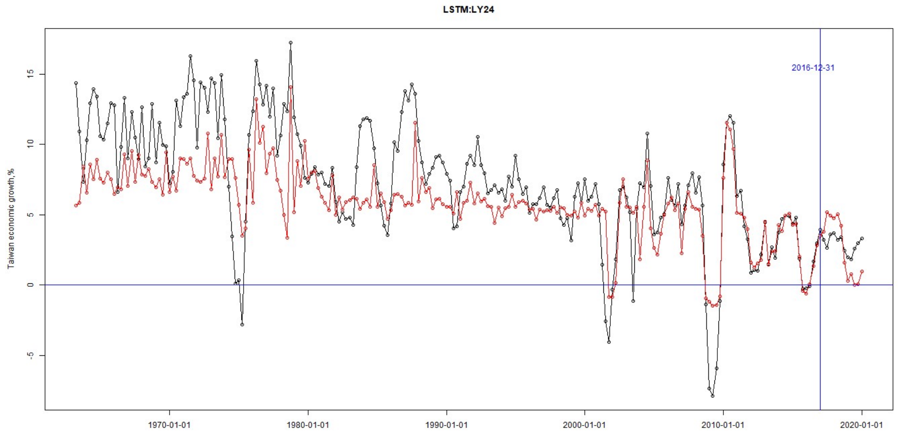
(D)


LSTM:LY14
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

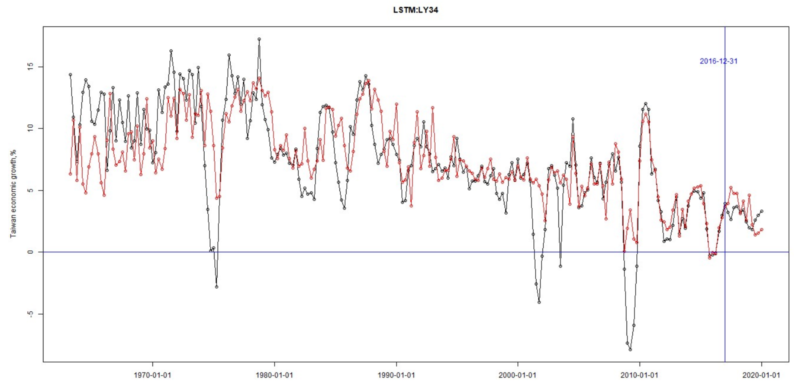
(E)


LSTM:LY23
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

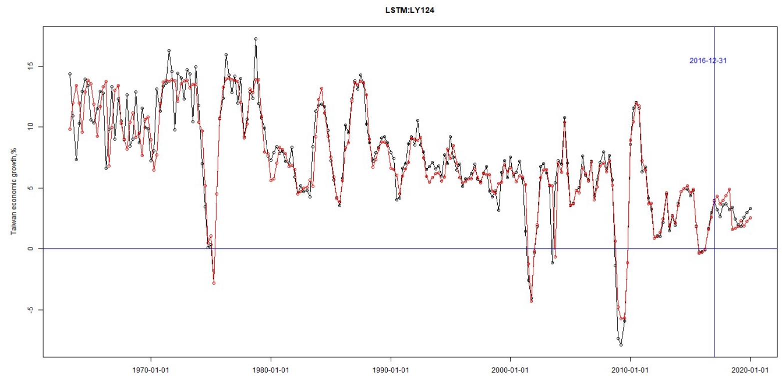
(F)


LSTM:LY24
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

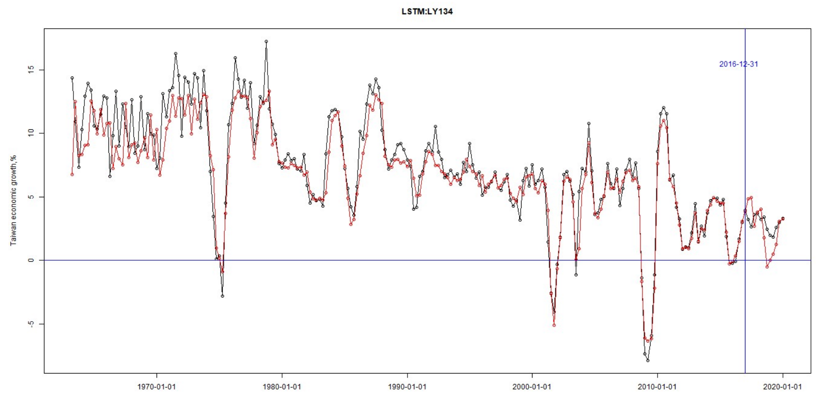
(G)



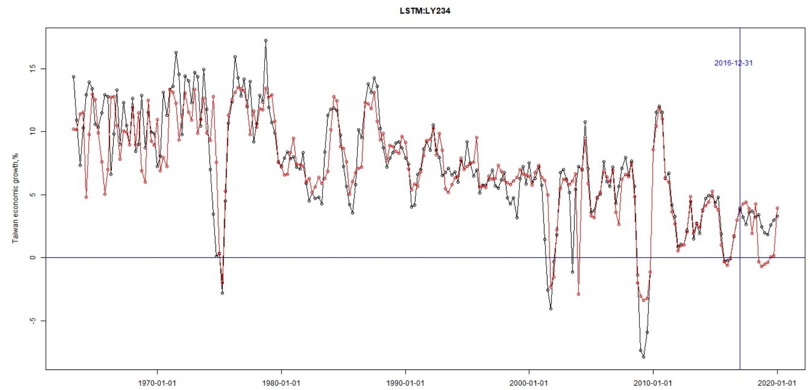
(H)



(I)



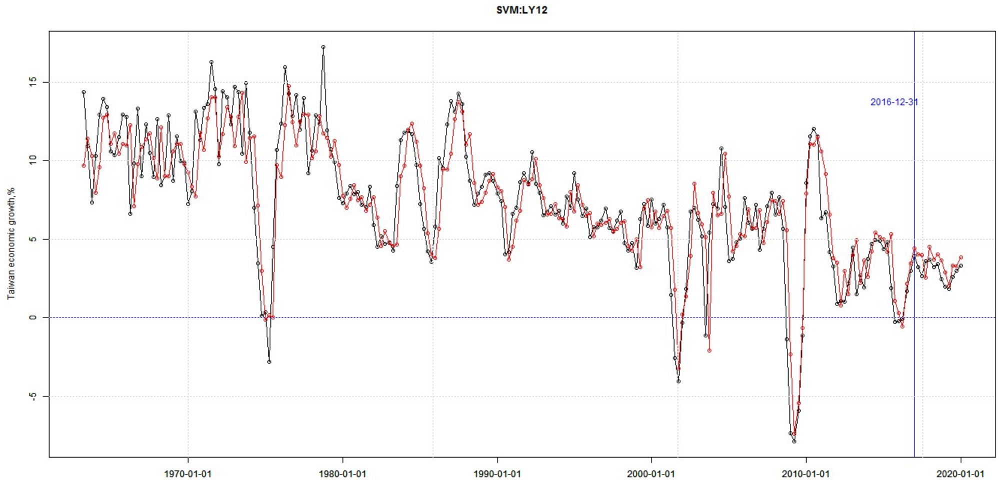
(J)



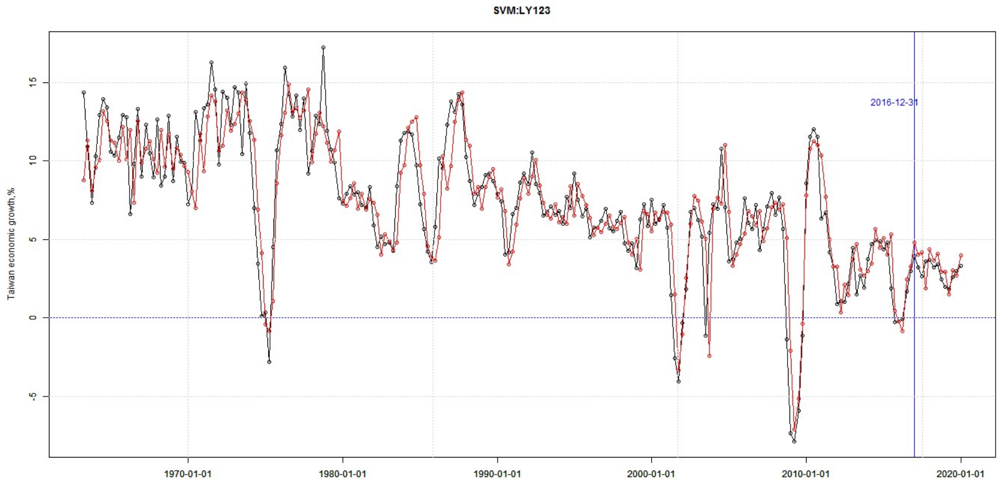
(K)

資料來源：本研究自行整理。

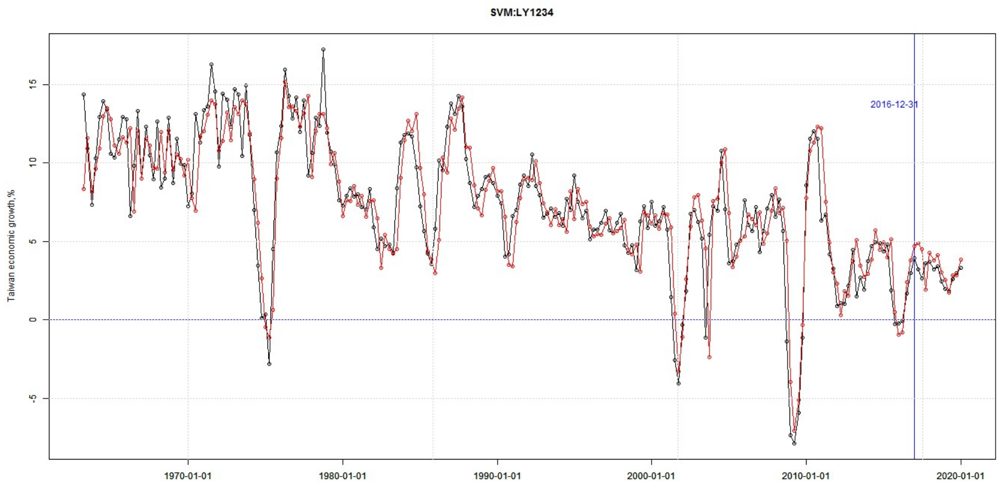
圖二十六、自我相關落後期模型：**LSTM**


SVM:LY12
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

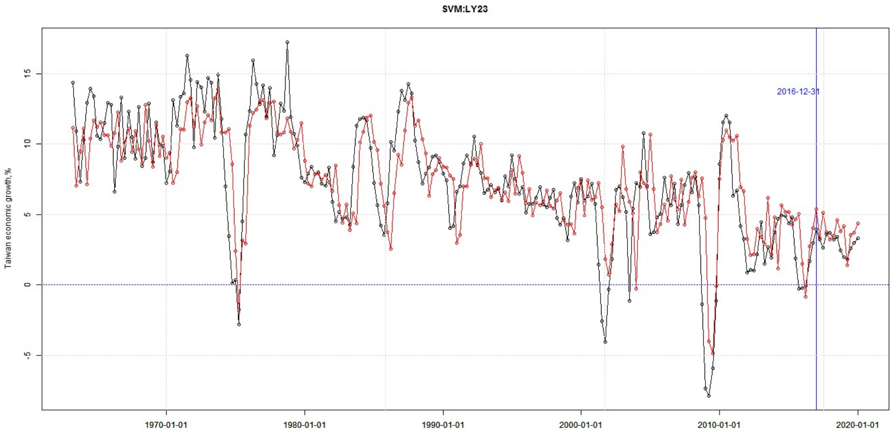
(A)


SVM:LY123
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

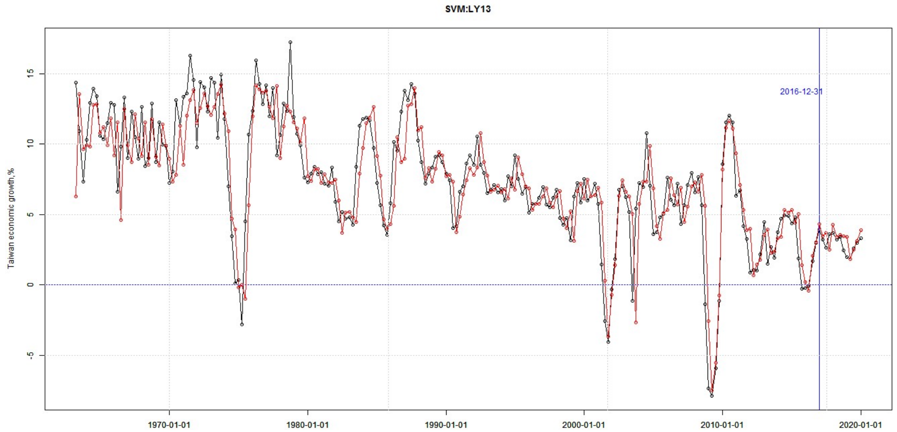
(B)


SVM:LY1234
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

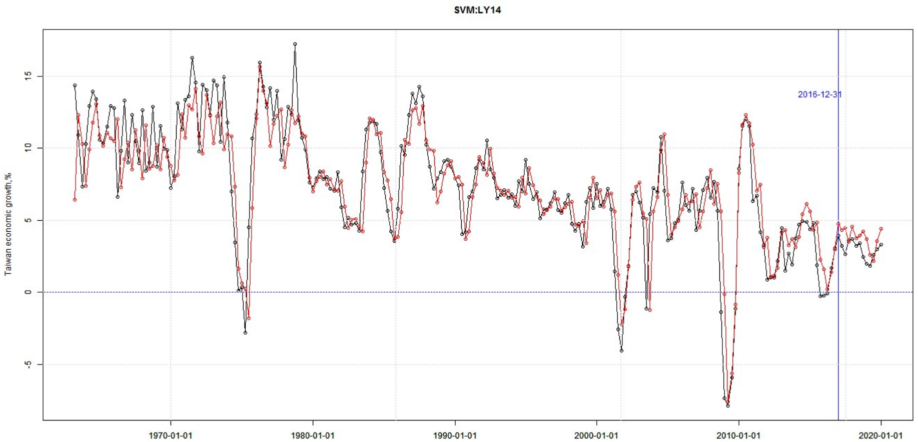
(C)


SVM:LY23
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

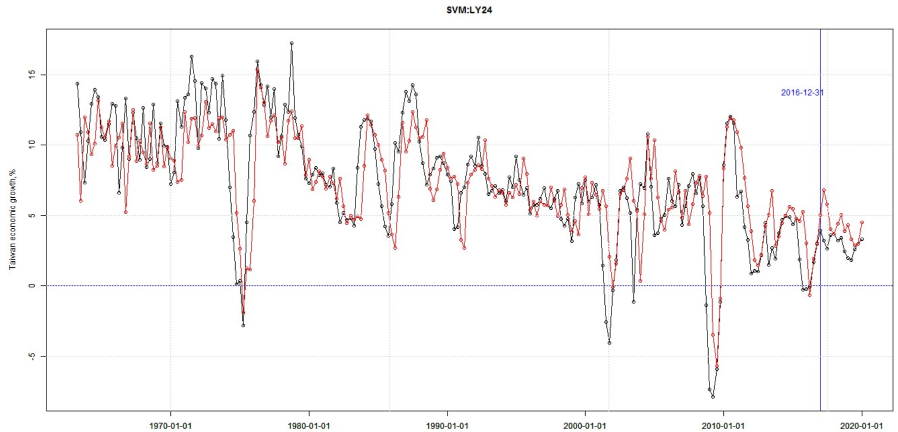
(D)


SVM:LY13
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

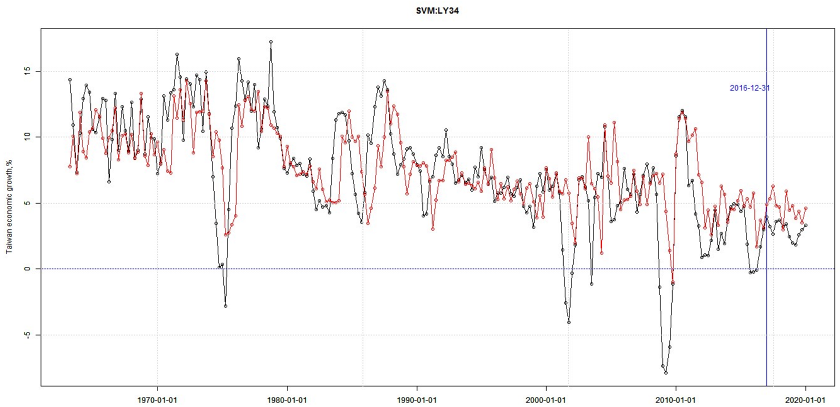
(E)


SVM:LY14
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

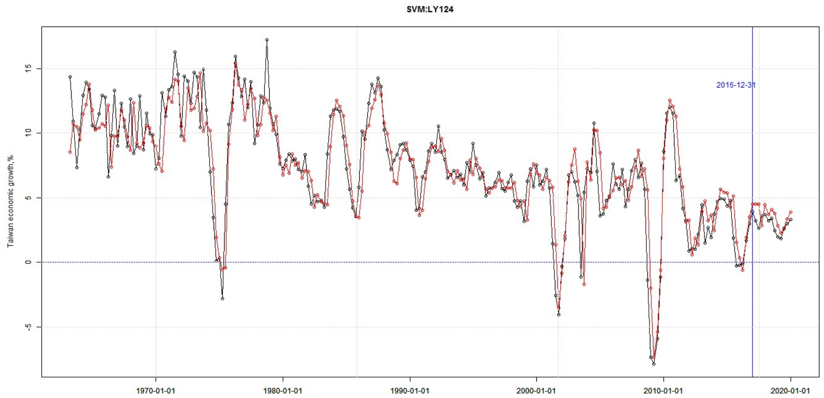
(F)


SVM:LY24
2016-12-31
Taiwan economic growth,%
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

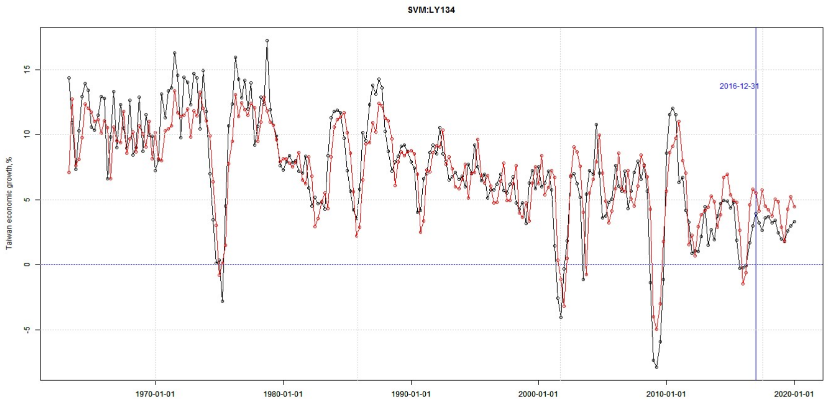
(G)



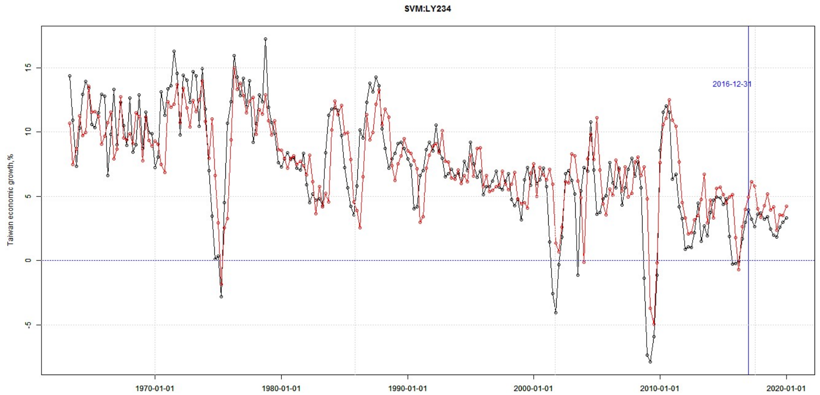
(H)



(I)



(J)



(K)

資料來源：本研究自行整理。

圖二十七、自我相關落後期模型：**SVM**

6.2 經濟成長率的 AR(*P*) 和添加其他變數的 AR(*P*)

令 $y_t = F(\mathbf{L}^p(y_t, \mathbf{X}_t)), p = 1,2,3,4$,

此處投入變數範例如下

$$y_t = F(y_{t-1}, y_{t-2}, X_{t-1}, X_{t-2}), p = 2,$$

其他變數 X：臺灣經濟變數和國外經濟成長率

臺灣經濟數據我們採用中央銀行經研處的建議，但是遷就時間，只有以下變數起始季和經濟成長率相同，其餘變數則因資料太短無法採用，例如短期利率始自 1987Q2，長期利率始自 1995Q4，通貨膨脹率始自 1982Q1。使用的四筆國內資料如下：

(1) 商品及服務輸出：1961Q1-2019Q4，主計總處。

(2) 商品及服務輸入：1961Q1-2019Q4，主計總處。

(3) 工業生產指數：1961Q1-2019Q4，經濟部工業局。

(4) 台幣兌美元匯率：1961Q1-2019Q4，中央銀行。

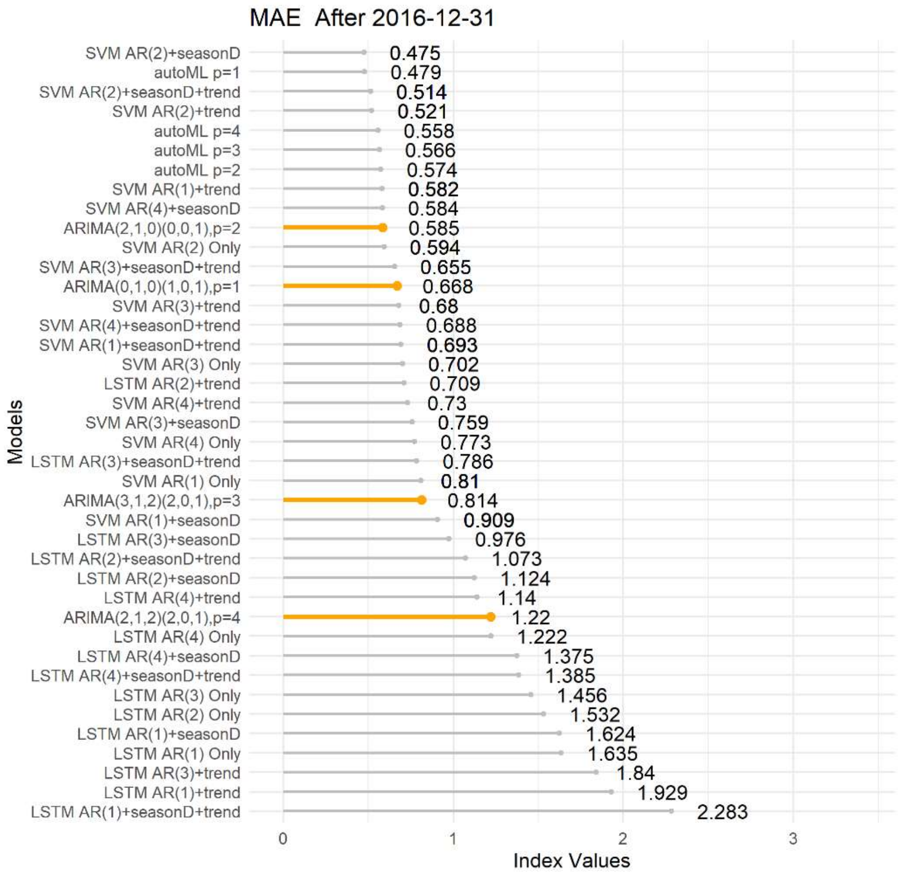
對應經濟成長率，上述工業生產指數與台幣兌美元匯率資料係由月頻率轉成季頻率，台幣兌美元匯率換頻率時，採用平均 (average) 和期末 (end-of-period) 兩種方式，故以上共有 5 筆資料。變動率計算時，使用當季對上年同季的年變動率。

國外數據為美國、OECD 及 G7 的經濟成長率。雖然 OECD 和 G7 的集團概念都包括美國，但是兩者也不盡然相同，如附錄一中的第一張圖。

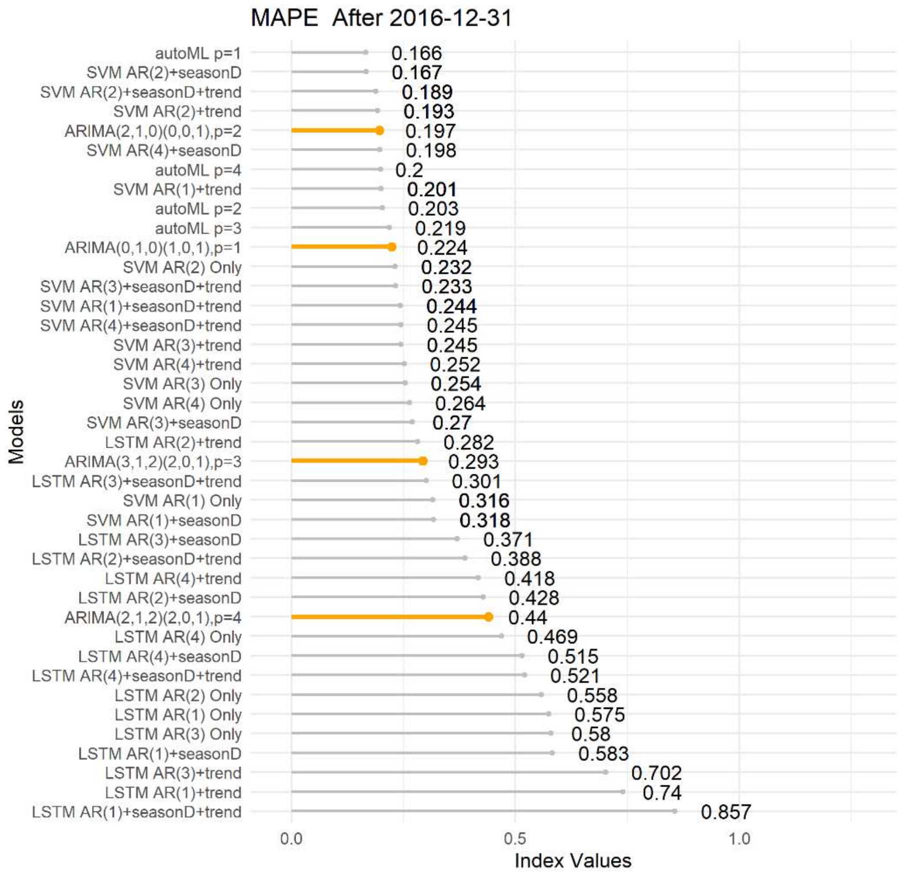
此節的其他變數共 8 個，包括 5 個臺灣變數變動率加 3 個外國經濟成長率。圖二十八和二十九將預測精確度指標排序，然後將最佳 ARIMAX (添加其他解釋變數 X 的 ARIMA 模型) 的結果用橘色標示以利比較。

從預測精確度指標看來，LSTM 雖然在深度學習享有盛名，但是本案時間序列的樣本太短，導致其訓練不足，無法展現其預測優勢。其次，最佳 ARIMAX 的結果並不差，和 SVM 與 autoML 相比，顯示機器學習方法不盡然佔據絕對優勢。

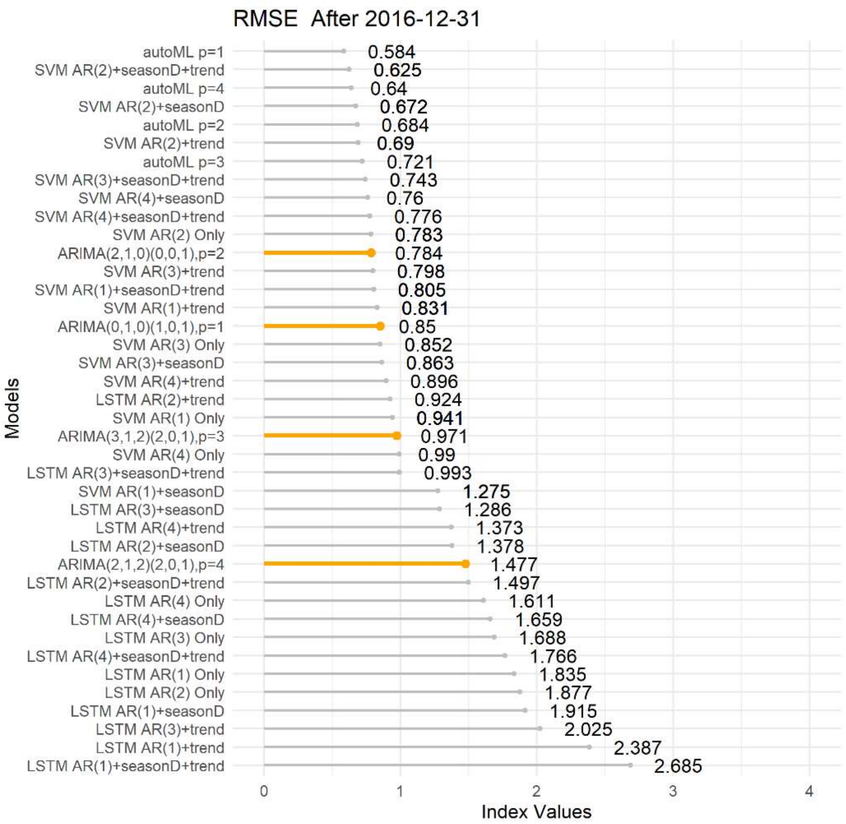
由於本節各模型產生大量圖形，我們將 36 張配適圖列於附錄一中，以免佔據正文太多空間而不利於閱讀。另外，因應央行及外部審查人建議，我們使用三個模式 (分別是 SVM，autoML 和傳統時間序列模型) 預測未來 8 季 (2020 年第 1 季至 2021 年第 4 季) 的經濟成長率，其結果詳見附錄二。


MAE After 2016-12-31
SVM AR(2)+seasonD 0.475
autoML p=1 0.479
SVM AR(2)+seasonD+trend 0.514
SVM AR(2)+trend 0.521
autoML p=4 0.558
autoML p=3 0.566
autoML p=2 0.574
SVM AR(1)+trend 0.582
SVM AR(4)+seasonD 0.584
ARIMA(2,1,0)(0,0,1),p=2 0.585
SVM AR(2) Only 0.594
SVM AR(3)+seasonD+trend 0.655
ARIMA(0,1,0)(1,0,1),p=1 0.668
SVM AR(3)+trend 0.68
SVM AR(4)+seasonD+trend 0.688
SVM AR(1)+seasonD+trend 0.693
SVM AR(3) Only 0.702
LSTM AR(2)+trend 0.709
SVM AR(4)+trend 0.73
SVM AR(3)+seasonD 0.759
SVM AR(4) Only 0.773
LSTM AR(3)+seasonD+trend 0.786
SVM AR(1) Only 0.81
ARIMA(3,1,2)(2,0,1),p=3 0.814
SVM AR(1)+seasonD 0.909
LSTM AR(3)+seasonD 0.976
LSTM AR(2)+seasonD+trend 1.073
LSTM AR(2)+seasonD 1.124
LSTM AR(4)+trend 1.14
ARIMA(2,1,2)(2,0,1),p=4 1.22
LSTM AR(4) Only 1.222
LSTM AR(4)+seasonD 1.375
LSTM AR(4)+seasonD+trend 1.385
LSTM AR(3) Only 1.456
LSTM AR(2) Only 1.532
LSTM AR(1)+seasonD 1.624
LSTM AR(1) Only 1.635
LSTM AR(3)+trend 1.84
LSTM AR(1)+trend 1.929
LSTM AR(1)+seasonD+trend 2.283
Models
0
1
2
3
Index Values

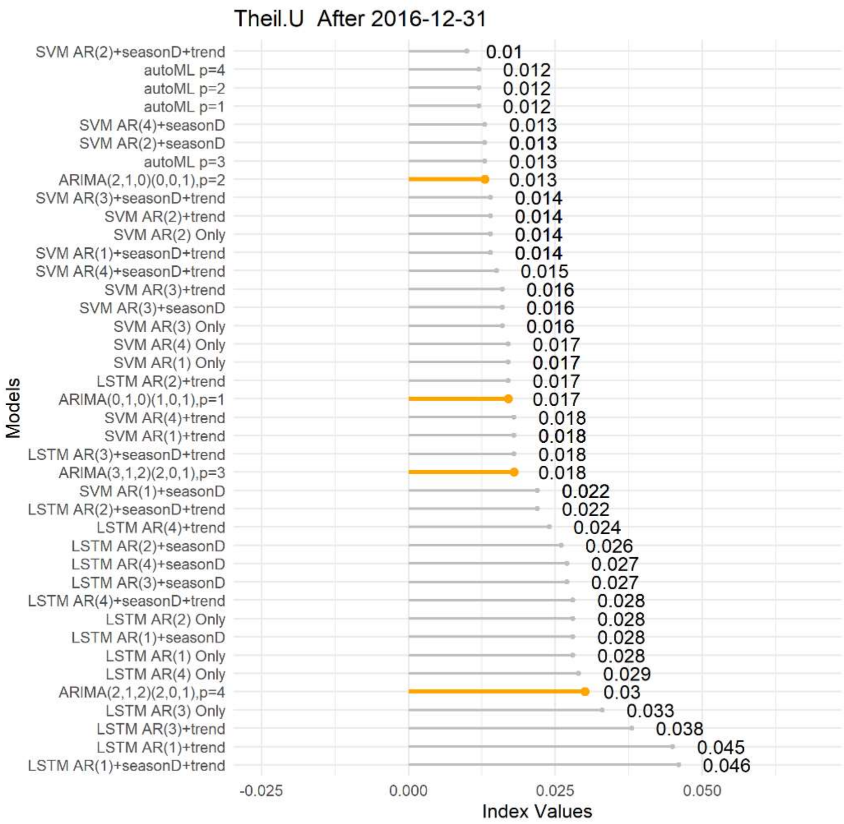
(A)


MAPE After 2016-12-31
autoML p=1 0.166
SVM AR(2)+seasonD 0.167
SVM AR(2)+seasonD+trend 0.189
SVM AR(2)+trend 0.193
ARIMA(2,1,0)(0,0,1),p=2 0.197
SVM AR(4)+seasonD 0.198
autoML p=4 0.2
SVM AR(1)+trend 0.201
autoML p=2 0.203
autoML p=3 0.219
ARIMA(0,1,0)(1,0,1),p=1 0.224
SVM AR(2) Only 0.232
SVM AR(3)+seasonD+trend 0.233
SVM AR(1)+seasonD+trend 0.244
SVM AR(4)+seasonD+trend 0.245
SVM AR(3)+trend 0.245
SVM AR(4)+trend 0.252
SVM AR(3) Only 0.254
SVM AR(4) Only 0.264
SVM AR(3)+seasonD 0.27
LSTM AR(2)+trend 0.282
ARIMA(3,1,2)(2,0,1),p=3 0.293
LSTM AR(3)+seasonD+trend 0.301
SVM AR(1) Only 0.316
SVM AR(1)+seasonD 0.318
LSTM AR(3)+seasonD 0.371
LSTM AR(2)+seasonD+trend 0.388
LSTM AR(4)+trend 0.418
LSTM AR(2)+seasonD 0.428
ARIMA(2,1,2)(2,0,1),p=4 0.44
LSTM AR(4) Only 0.469
LSTM AR(4)+seasonD 0.515
LSTM AR(4)+seasonD+trend 0.521
LSTM AR(2) Only 0.558
LSTM AR(1) Only 0.575
LSTM AR(3) Only 0.58
LSTM AR(1)+seasonD 0.583
LSTM AR(3)+trend 0.702
LSTM AR(1)+trend 0.74
LSTM AR(1)+seasonD+trend 0.857
Models
0.0
0.5
1.0
Index Values

(B)



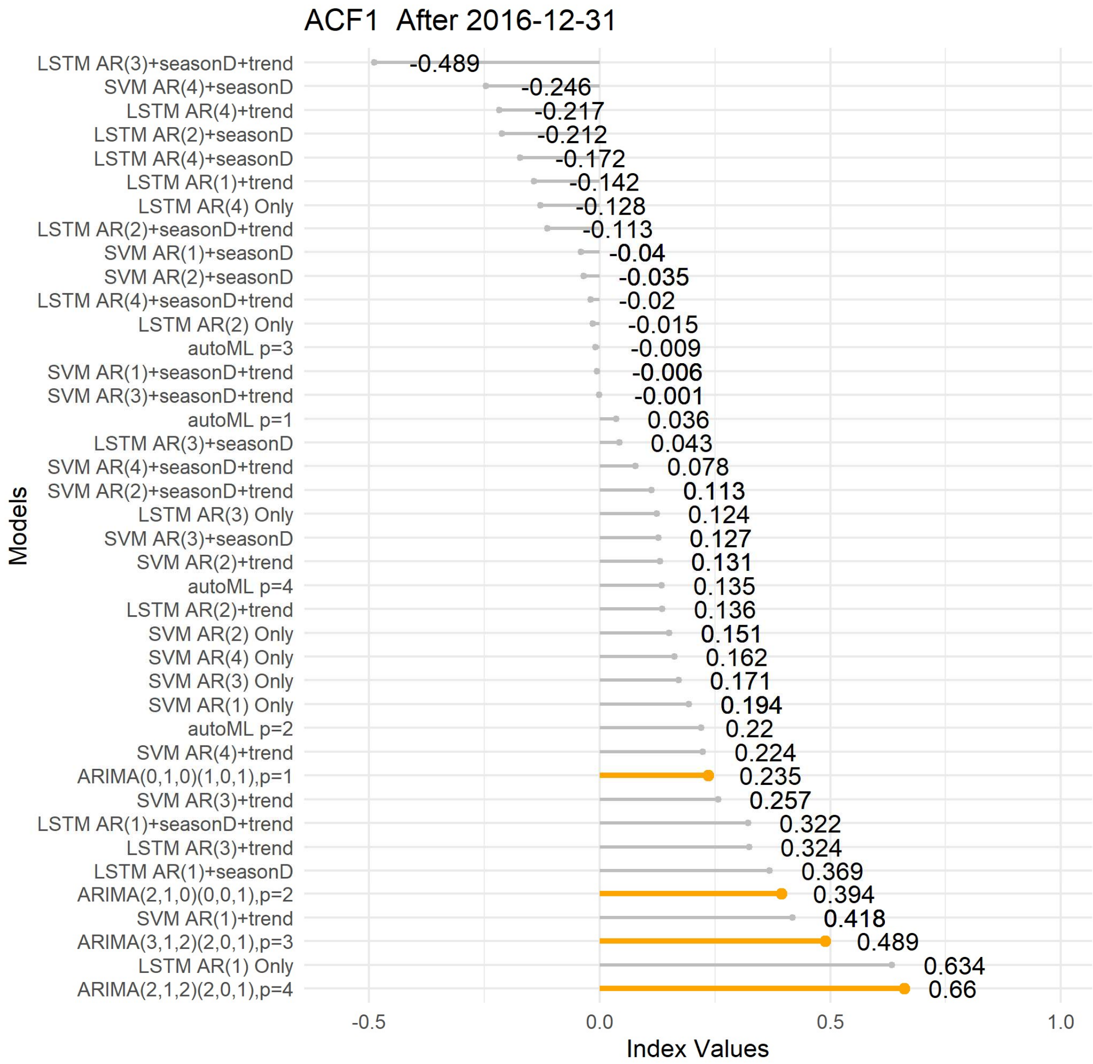
(C)



(D)

資料來源：本研究自行整理。

圖二十八、增加其他變數連續落後 **h** 階的 **4** 個預測精確度指標



資料來源：本研究自行整理。

圖二十九、增加其他變數連續落後 **h** 階：誤差的 **ACF(1)** 值

6.3 所有變數的混合落後

當我們遇到很多投入變數時，有兩個解決途徑：其一是逐次清算，計算量大；其二是降維，計算量少。降維度的方法可以採用主成分 (Principal Component Analysis, PCA) 或獨立成分法 (Independent Component Analysis, ICA)。如果資料量很大，降維或許是一個選擇，但 PCA 和 ICA 結果並不理想，呈現列附錄三。此節採用逐次清算作法，理由如下：

(1) 可以檢視變數組合的關鍵。如果逐一清算可以發現少量變數，其實就不需要 ICA 降維。降維度的合理性，也可以由逐一清算的方式得到支持。

(2) 本案有明確的預測變數 (經濟成長率)。機器學習的預測，不是單方面對投入變數獨自降維即可一體適用。換言之降維也必須多次計算，適合 SVM 的不一定適合 LSTM。

$y_t = F(\mathbf{L}^{\mathbf{k}_1}(y_t);\mathbf{L}^{\mathbf{k}_2}(X_t))$, $\mathbf{k}$ 集合定義如前，此處投入變數範例如下

$$y_t = F(y_{t-1}, y_{t-4}; X_{t-2}, X_{t-4}).$$

賡續前述，多步預測在演算上更大的負擔，就是如何決定解釋變數的混合落後。為避免演算規模過於龐大，此處解釋變數 X 只選取 3 個變數：OECD 經濟成長率 (x_1)，OECD 領先指標成長率 (x_2)，和美國領先指標成長率 (x_3)，合併臺灣經濟成長率 (y) 自我相關落後期數。[8] 落後期 $\mathbf{k}_1,\mathbf{k}_2 = 1,2,3,4$，臺灣經濟成長率自我落後結構有 15 種組合 $\mathbf{k}_1 \in \{(1,2),(1,3),\ldots.(1,2,3,4)\}$。另外，在給定 $\mathbf{k}_1$ 下，x_1, x_2, x_3 共有 12 個落後變數。從組合算數，這樣的變數交錯有 4,095 組

$$\sum_{k=1}^{12} C_k^{12} = 4,095,$$

[8] 6.2 節的結果並不理想，所以 6.3 節我們重新擷取國外數據以找出最適合預測臺灣經濟成長率的變數組合。其次，第 6.2 章的 OECD 與 G7 雖有包括美國，但趨勢和波動幅度並不相同。

為了簡化我們的運算，12 種狀況中，前 11 種每一抽樣 6 個落後變數組合，最後一種 C_{12}^{12} 全取，總計 66+1=67 種模型。為了簡化過多的計算，此處只選擇計算較方便的 SVM。15 組自我落後組合，每組自我落後結構的 X 有 67 種隨機組合，總計算 15×67=1,005 種投入變數組合，4 核心平行運算，耗時 30 小時 (細節見第 8 章結論說明)。

結果簡單摘要如表四，可知如果數據大而無當，對預測結果未必有所助益。經過這樣的運算，與前面純粹的自我相關模型相比，添加解釋變數的確能夠改善預測表現，只不過要從大量的變數中選出有用的，需要大規模計算。

本文旨在找出預測績效最佳的機器學習模型及變數組合，Makridakis (1993) 針對預測指標穩定性的研究，指出 MAPE 相對穩定；Hyndman and Koehler (2006) 則建議採 scaled error 的 MAE。我們簡化結果，就以 RMSE 和 MAPE 列出現階段得到的最佳結果：RMSE=0.51，MAPE=0.157。但受限於經濟成長率為季資料，資料長度短，自由度有限，因此計算力弱。倘能改善資料長度問題，應能有更佳預測表現。由此可見，機器學習的原則係資料驅動，此與建構一個經濟模型下的因果關係不同。選擇變數之前，並沒有什麼特別的事先經濟考量，僅從資料的相關性來判斷，因此不限於經濟理論模型所討論的控制變數。此外，機器學習也透過資訊型態配適出合適的模型及參數，故可有最佳的預測績效。

表四、**SVM** 混合落後自我相關並增加其他變數模型的多步預測正確性

MAPE	**SVM** 演算變數組合
0.157	$F(y_{t-1},y_{t-3};x_{2t-1})$
0.164	$F(y_{t-1},y_{t-2};x_{1t-2},x_{1t-4},x_{2t-3})$
0.176	$F(y_{t-1},y_{t-2},y_{t-3},y_{t-4};x_{1t-3},x_{1t-4},x_{2t-1})$
0.180	$F(y_{t-1},y_{t-3},y_{t-4};x_{1t-1},x_{1t-2},x_{1t-4},x_{2t-1},x_{2t-2})$
0.181	$F(y_{t-1};x_{1t-3},x_{1t-4})$
0.190	$F(y_{t-1},y_{t-2},y_{t-3};x_{1t-1},x_{2t-1},x_{3t-1},x_{2t-3},x_{2t-4},x_{3t-2})$
0.201	$F(y_{t-1},y_{t-4};x_{1t-2},x_{1t-3},x_{1t-4},x_{2t-1},x_{2t-3},x_{3t-1})$
0.206	$F(y_{t-1},y_{t-2},y_{t-4};x_{1t-1})$
0.21	$F(y_{t-2};x_{1t-2},x_{1t-3},x_{2t-2},x_{2t-3},x_{2t-4})$
0.24	$F(y_{t-2},y_{t-3};x_{1t-1},x_{1t-3},x_{2t-2})$
RMSE	**SVM** 演算變數組合
0.51	$F(y_{t-1},y_{t-3};x_{2t-2})$
0.562	$F(y_{t-2},y_{t-3},y_{t-4};x_{1t-3},x_{2t-1},x_{2t-2})$
0.591	$F(y_{t-1},y_{t-2};x_{1t-2},x_{1t-4},x_{2t-3})$
0.592	$F(y_{t-1},y_{t-2},y_{t-3},y_{t-4};x_{1t-4},x_{2t-1},x_{2t-3})$
0.66	$F(y_{t-1};x_{1t-1},x_{1t-4},x_{2t-1})$
0.67	$F(y_{t-2};x_{1t-1},x_{1t-3},x_{2t-2})$
0.69	$F(y_{t-1},y_{t-2},y_{t-3};x_{1t-1},x_{2t-1},x_{2t-2},x_{2t-2},x_{3t-1},x_{3t-2})$
0.73	$F(y_{t-1},y_{t-2},y_{t-4};x_{1t-1})$
0.733	$F(y_{t-1},y_{t-4};x_{1t-2},x_{1t-3},x_{1t-4},x_{2t-1},x_{2t-3},x_{3t-1})$
0.778	$F(y_{t-2},y_{t-4};x_{1t-3},x_{2t-2},x_{2t-4})$

資料來源：本研究自行整理。

本文也執行 1,005 組 autoML 的混合運算，耗時 46 小時。表五的預測績效指標顯示其預測績效更好，MAPE 或 RMSE 最小值的下降幅度相當明顯。在下一節我們將說明，計算這 1,005 組的預測平均值，將可呈現出更好的預測結果。

表五、autoML 混合落後自我相關預測正確性

最小指標	**autoML 演算變數組合**
MAPE=0.071	$F(y_{t-2}, y_{t-3}, y_{t-4}; x_{1t-1}, x_{2t-2}, x_{2t-4}, x_{3t-4})$
RMSE=0.29	$F(y_{t-1}, y_{t-2}; x_{1t-1}, x_{1t-3}, x_{1t-4}, x_{2t-1}, x_{2t-2}, x_{3t-1}, x_{3t-3}, x_{3t-4})$

資料來源：本研究自行整理。

6.4 預測平均法 (Forecasting Average)

機器學習係在訓練過程中，根據驗證方法選擇最佳結果，然而實際上很難知道什麼是最好的預測。計量經濟學有一個模型平均法 (Model Average)，也就是將多個模式的預測予以平均，如 Hansen (2007)、Hansen and Racine (2012) 和 Liao and Tsay (2020)。在機器學習文獻中，稱之為委員會方法 (Committee approach)，見 Bishop (2009) 及 Hastie et al. (2009)。多模型預測平均法的構思如下：因為模型產生的預測，相對於真實值不是高估就是低估，因此取其平均或中位數，就很可能會提高精確度。

為簡化起見，此節不擬從最適多步預測平均 (Optimal Multistep Forecasting Averaging) 的文獻方向討論理論、檢定與組合法，我們只計算兩種機器學習的簡單平均。第 1 部份以 6.1 節的 15 種組合設定所產生的 45 個預測結果，計算 cross-sectional 平均和中位數作為預測值，再跟真實的經濟成長率對照計算精確度。從表六可知道，autoML 的 15 個預測結果的平均 (autoML.mean) 或中位數 (autoML.median)，可以大幅增加預測精確度。LSTM 和 SVM 表現較差，但是當 45 個模型放一起時 (All.mean 和 all.median)，整體績效還是提升很多。

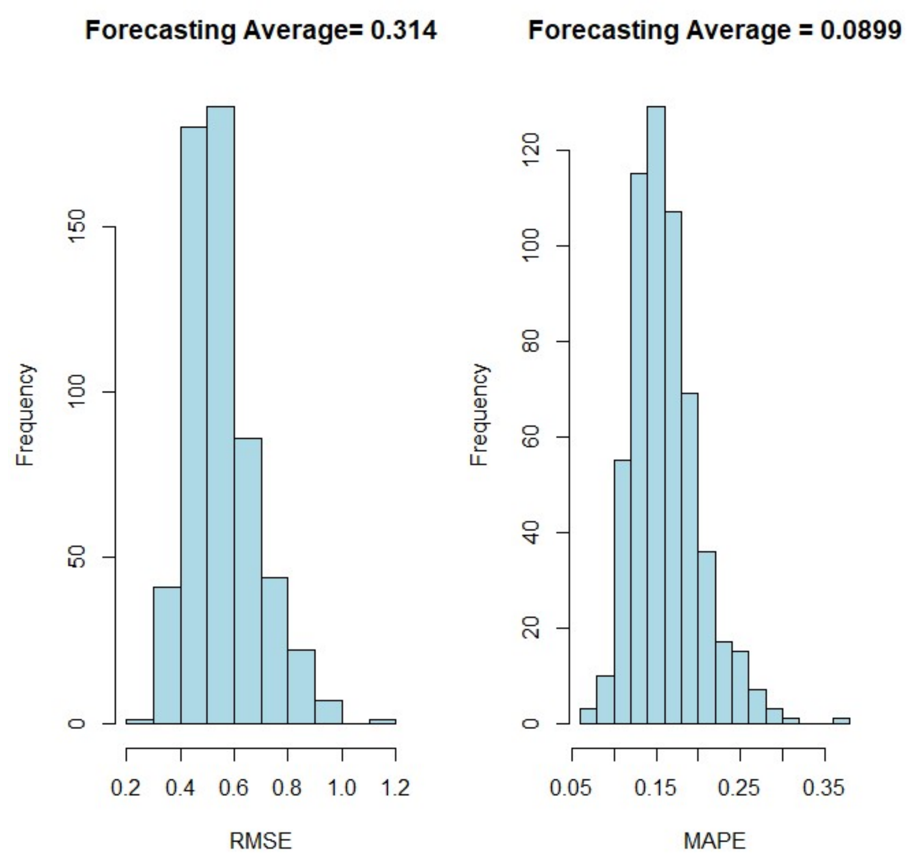
表六、依據 **(6.1)** 的模型平均預測精確度指標

	RMSE	**MAE**	**MAPE**	**ACF1**	**Theil's U**
autoML.mean	0.4329	0.3626	0.1343	0.3153	0.0075
autoML.median	0.4699	0.3843	0.1414	0.3314	0.0080
LSTM.mean	0.9061	0.7330	0.2630	0.3711	0.0132
LSTM.median	0.9657	0.7973	0.2872	0.3931	0.0149
SVM.mean	1.1591	1.0059	0.3815	-0.0255	0.0189
SVM.median	1.0337	0.8937	0.3374	-0.0710	0.0170
All.mean	0.5393	0.3696	0.1392	0.0551	0.0078
All.median	0.5417	0.4046	0.1541	-0.0752	0.0084

資料來源：本研究自行整理。

第 2 部份則根據 6.3 節的方式，我們也完成了相當耗時的 autoML。如圖三十所示，我們計算 1,005 組預測值的橫斷面平均(cross-sectional average)，得出 RMSE 與 MAPE，並將之繪成直方圖 (histogram)。[9]

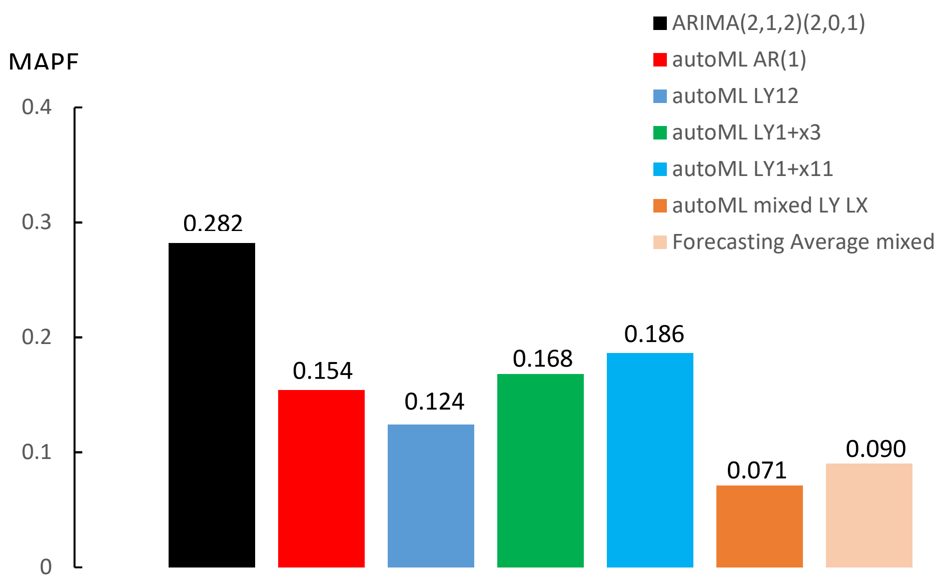
[9] 圖三十係以此樣本外預測的平均值來計算預測績效，而表五呈現的則是在各種其他變數組合下，篩選出預測績效最佳的變數組合，故表五的 MAPE 及 RMSE 必定小於圖三十。



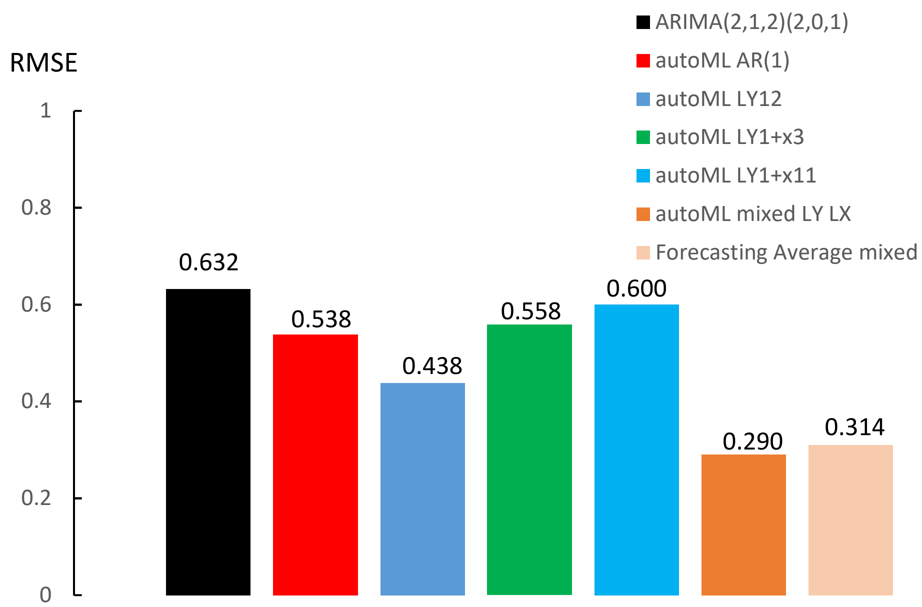
資料來源：本研究自行整理。

圖三十、1,005 組 autoML 的預測與預測平均之表現

綜言之，將多個機器學習模式產生的多步預測，用預測平均法組合成單一預測，實務上應該是很理想的一種作法，文獻上有加權平均的方式，應可以持續發展。再者，以機器學習方法改善時間序列模型預測的可行性，仍需要一個紮實的理論支撐，本節只是嘗試設計一個作法，初步判讀其可行性。若就資訊科學的角度思考，只要硬體演算能力夠強大，用 autoML 對資料直接進行大規模計算，找出最佳資料結構，再使用模型平均法，應是改善時間序列預測更為有效的方式。最後，我們在圖三十一列出績效最小值的綜合直方圖。這個綜合結果指出，添加解釋變數不一定對預測有幫助，這再次說明預測平均法或許是一個綜合的好方式。



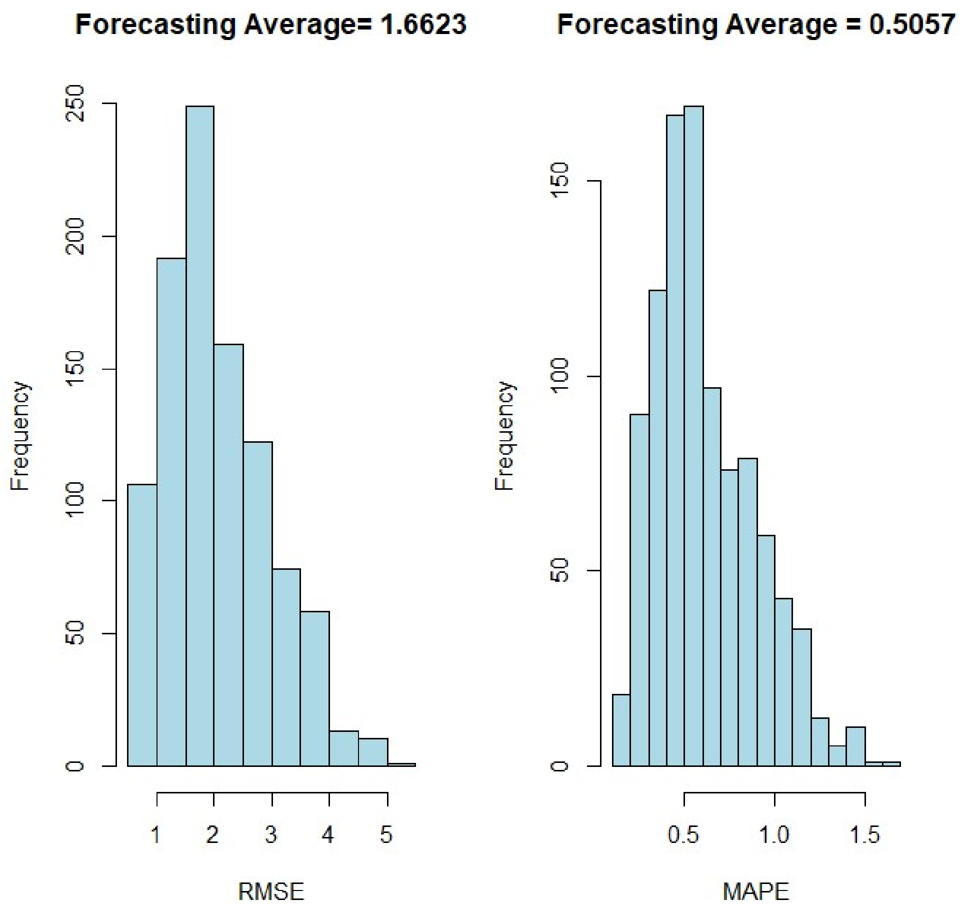
（A）



（B）

圖三十一、綜合比較

然而，預測平均僅適用於預測狀況比較好的情況。例如圖三十二用於 SVM 的結果就不甚理想。



圖三十二、**1,005** 組 **SVM** 的預測與預測平均表現

7　以機器學習診斷時間序列模型的預測誤差

本計畫最後一個項目，是研究如何以機器學習方法，改善傳統標準時間序列模式，例如 ARIMA。這個研究動機很簡單，因為時間序列模式在複雜的機器學習模式 (如 AutoML 和 LSTM) 下，往往在決定最佳模式時的演算會相當費時，如果用一個相對簡易的模型 (SVM) 去改善最佳 ARIMA 的預測，或許是一個可以考慮的嘗試。此處是一個嘗試作法，沒有理論基礎，也沒有在計算時將兩類模型整合。許多理想的綜合作法，或有待後繼者賡續探索。

我們執行的架構流程如下：

第 1 步：模型預測誤差的產生

估計 4 個單變量時間序列模式

$$y_t = f(y_{t-L}) + e_t.$$

如表七所示，第 1 個模式 ARIMA(2,1,2)(2,0,1)，其階次是採用自動配適的結果，第 2 個模式是添加季節虛擬變數，第 3 與第 4 個模式則如本報告第 3 章第 1 節的解釋。由多個精確度指標來看，第 1 個模式 ARIMA(2,1,2)(2,0,1)的誤差最小，因此使用其殘差進行後續分析。

表七、時間序列模型的多步預測正確性

Models	**Accuracy**				
	RMSE	**MAE**	**MAPE**	**ACF1**	**Theil's U**
ARIMA(2,1,2)(2,0,1)	0.6315	0.5105	0.2082	0.5033	0.0121
ARIMA(2,1,2)(2,0,1)+seasonDummy	0.9570	0.7976	0.3183	0.3849	0.0179
baggedModel.ETS	0.9386	0.7436	0.3153	0.5414	0.0194
baggedModel.autoArima	0.6501	0.5499	0.2213	0.4699	0.0126

資料來源：本研究自行整理。

第 2 步：確認預測誤差是否可由機器學習模式提取可預測成份

因為殘差項 $e_t = y_t - E\left(y_t \middle| x_{t-L}\right)$，$e_t > 0$ 模型預測低估，反之為預測高估。我們依照正負條件，把 ARIMA(2,1,2)(2,0,1)殘差項高估與低估分別編碼成 $\{0,\ 1\}$ 二元類別變數如下：

$$ID_t = \begin{Bmatrix} e_t \geq 0 \\ e_t < 0 \end{Bmatrix},$$

$$ID_t = tree(x_{t-L}) + \varepsilon_t$$

然後用分類演算法檢視基本預測能力。分類演算法有決策樹 (decision tree) 與隨機森林 (random forest)。這個二元分類預測的作法，初步確認時間序列預測誤差的高估與低估是否具可預期性 (Predictability)。我們採用如表八的隨機森林混淆矩陣 (confusion matrix)，假設 Positive Class= 1。

表八、隨機森林的混淆矩陣

隨機森林混淆矩陣		Actual	
		高估 0	低估 1
Predicted	高估 0	C=3	B=2
	低估 1	A=1	D=4

資料來源：本研究自行整理。

由混淆矩陣顯示出的可預測性指標如下所示：

1. 正確率 (Accuracy)：總樣本中主對角線預測正確率，即 (C+D)/(A+B+C+D)。
2. 精確率 (Precision)：Positive=1 的預測值(A+D)與真實值(D)相符的比重，D/(A+D)。
3. $\text{Kappa}(\kappa) = \dfrac{\text{觀察符合率} - \text{機率符合率}}{1 - \text{機率符合率}}$

透過機率符合率和觀察符合率，綜合評估模型對資料的可預測性。[10]

4. 敏感度 (Sensitivity)：殘差項為正，並被預測為低估的比率，D/(B+D)。
5. 特異度 (Specificity)：殘差項為負，且被預測為高估的比率，C/(A+C)。

估計結果如下：

Accuracy = 0.7
Kappa = 0.4
Precision = 0.67
Sensitivity (Recall of ID=1) = 0.6
Specificity (Recall of ID=0) = 0.8

正確率 (Accuracy) 為 70%，精確率 (Precision) 為 67%，兩指標均略低。Kappa 係作為預測修正的權重，因為它具備了如下性質：-1≤Kappa≤1，若 Kappa＜0，說明機率符合率大於觀察符合率；Kappa=0，說明診斷試驗的結果完全由機率因素導致；Kappa＞0 說明觀察符合率大於因為機率符合的程度。既有研究總結得出，如果 0＜Kappa≤0.40，則說明診斷試驗的可重複性差；如果 0.40＜Kappa＜0.75，則說明具有中、高度可重複性；如果 Kappa≥0.75，那麼該診斷試驗就具有極好的可重複性。臨床醫生和研究者們根據這一指標綜合評估診斷試驗的可重

[10] Kappa coefficient 是一種統計量化指標，平衡類別之間正確性的一種指標，簡單說就是要把大者恆大的影響消除掉的指標。假設表八“高估”的樣本數遠大於低估時，只需把資料都判給“高估”一類，整體的正確率都很高/錯誤率很低。計算中的觀察符合率係指預測正確的機率，為總樣本中主對角線預測正確率，即百分比混淆矩陣的對角線元素相加；機率符合率則是把每個類別(實際第 i 類別的總和乘上預測第 i 類別的總和) 做加總，即

$$\frac{(A+C)}{(A+B+C+D)}\times\frac{(B+C)}{(A+B+C+D)}+\frac{(B+D)}{(A+B+C+D)}\times\frac{(A+D)}{(A+B+C+D)}$$

複性，排除機遇因素干擾。[11]本研究殘差項低估的 Kappa 為 0.4，即診斷試驗的可重複性差。後面的預測修正以 Kappa 為權重。

最後，以敏感度及特異度的指標，Youden's J index (Sensitivity+Specificity-1) 聯手衡量預測的總貢獻，J index 越接近 1 越好，而本文所得到的值為 0.4。

第 3 步：以機器學習模型直接預測殘差 e_t，修正 ARIMA 經濟成長率預測值

$$e_t = g(x_{t-L}) + \xi_t .$$

為節省演算時間，此步驟 g 函數僅採取 SVM 的預測值。如下表，SVM-EAP 之 EAP 表示誤差調整的預測 (Error Adjusted Prediction)，3x 和 11x 代表上式函數 *g()*內的其他解釋變數 x 的數量為 3 個或 11 個。解釋變數 x 數量為 3 時，係指 3 個外國經濟成長率；數量為 11 則是因為再添加 8 個臺灣總經變數的變化率。下表是用 SVM 預測樣本外 12 季 ARIMA 的預測誤差，然後把預測的誤差以 Kappa 加權後，直接修正 ARIMA 預測。

根據表九，經過預測誤差調整後的預測績效多較原來 ARIMA 模型要差，只有少量狀況在預測誤差調整後，預測績效改善，例如 SVM-EAP 3x L=1，或是 SVM-EAP 11x L=4,5,7,8，但僅小幅度改善。唯有在 SVM-EAP 11x L=2 時，亦即 11 個解釋變數落後兩期時，預測績效改善幅度最佳。

---

[11] 混淆矩陣中若有類別間樣本不平均，使得預測時發生大者恆大，也就是只要把資料都判給數量最多的一類，整體的正確率都很高的情況，而 Kappa 指標就是在判別是否有此情況。臨床應用上，如 Landis and Koch (1977) 以混淆矩陣應用在神經學的試驗上，來分析不同地區病患的病徵差異，同樣使用 Kappa 指標判斷不同地區間不平均的情況。近年研究機器學習模型廣泛應用在睡眠階段的診斷，也有運用 Kappa 指標 (Peker, 2016；Kuo and Chen, 2020)。

表九、修正 **ARIMA** 並增加其他變數模型之正確性

	RMSE		MAE		MAPE		ACF1		Theil's U		Kappa
ARIMA(2,1,2)(2,0,1)	**0.6315**		**0.5105**		**0.2082**		**0.5033**		**0.0121**		**NA**
SVM-EAP 3x L=1	0.6087	-	0.4870	-	0.1991	-	0.4735	-	0.0113	-	0.3333
SVM-EAP 3x L=2	0.6650	+	0.5085	-	0.2137	+	0.4818	-	0.0132	+	0.3333
SVM-EAP 3x L=3	0.6714	+	0.5278	+	0.2191	+	0.4965	-	0.0131	+	0.1667
SVM-EAP 3x L=4	0.6842	+	0.5424	+	0.2245	+	0.5065	+	0.0134	+	0.1667
SVM-EAP 3x L=5	0.6491	+	0.5174	+	0.2130	+	0.4948	-	0.0126	+	0.1667
SVM-EAP 3x L=6	0.6767	+	0.5344	+	0.2218	+	0.4970	-	0.0133	+	0.1667
SVM-EAP 3x L=7	0.5894	-	0.5125	+	0.2002	-	0.5082	+	0.0108	-	-0.1667
SVM-EAP 3x L=8	0.6315		0.5105		0.2082		0.5033		0.0121		0.0000
SVM-EAP 11x L=1	0.6823	+	0.5385	+	0.2236	+	0.5514	+	0.0133	+	0.1667
SVM-EAP 11x L=2	0.5014	-	0.4490	-	0.1656	-	0.3847	-	0.0086	-	-0.3333
SVM-EAP 11x L=3	0.6315		0.5105		0.2082		0.5033		0.0121		0.0000
SVM-EAP 11x L=4	0.5784	-	0.4936	-	0.1922	-	0.4752	-	0.0108	-	-0.3333
SVM-EAP 11x L=5	0.5904	-	0.5092	-	0.1996	-	0.4436	-	0.0111	-	-0.3333
SVM-EAP 11x L=6	0.6211	-	0.5159	+	0.2073	-	0.4951	-	0.0119	-	-0.1667
SVM-EAP 11x L=7	0.6168	-	0.5075	-	0.2044	-	0.4798	-	0.0117	-	-0.1667
SVM-EAP 11x L=8	0.6289	-	0.5090	-	0.2066	-	0.4680	-	0.0120	-	-0.1667

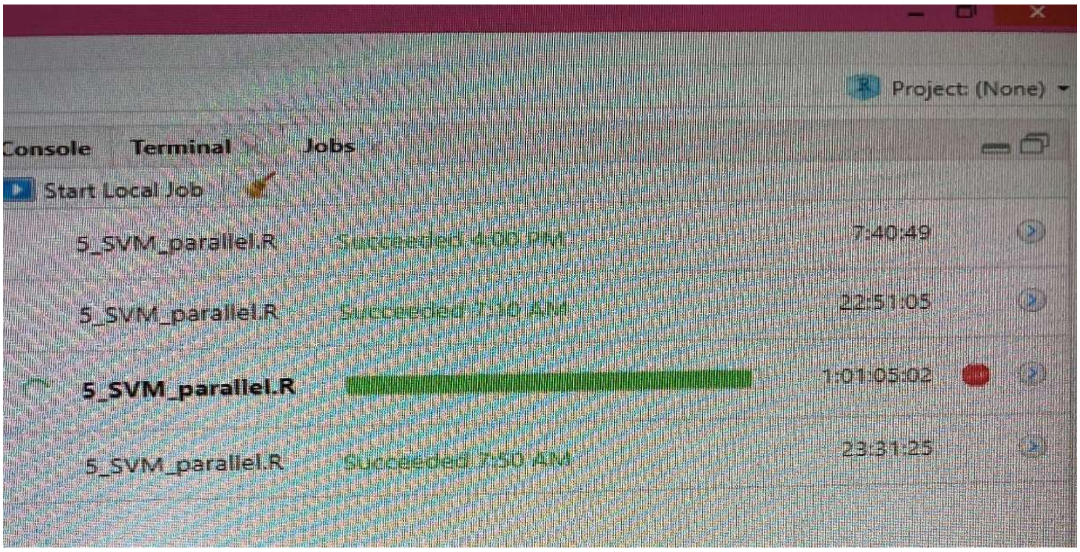
資料來源：本研究自行整理。

註：+ 表示經 SVM-EAP 修正後，預測誤差高於時間序列模式；
- 表示經 SVM-EAP 修正後，預測誤差低於時間序列模式。

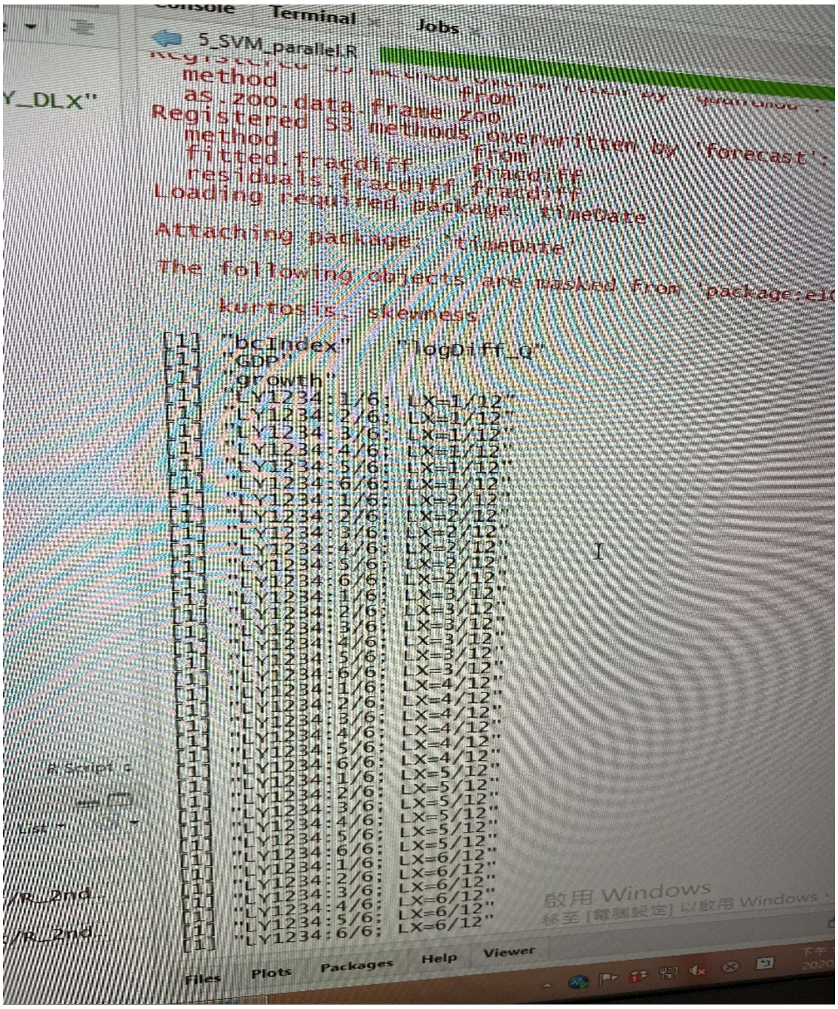
8　結論

本案嘗試所謂的機器學習演算法，能否改善時間序列預測的表現，利用臺灣經濟成長率為範例，採用 SVM，自動化機器學習 (autoML) 和深度學習的 LSTM 三個模式。投入變數 (input variables) 則有三種：y (被解釋變數) 從 AR(1) 到混合多階，以及添加大量解釋變數的混合多階落後組合。至於以機器學習改善時間序列預測這一部份，本案結論是謹慎看待，未來仍可能有改善空間。進一步說明如次：

1. 根據既有機器學習的演算經驗，多半需要先進行大量訓練，然後再進行預測。我們發現產生多組預測雖屬可行，但是十分耗時。如以圖三十三的二張實際操作截圖為例，SVM 的混合計算，採用平行運算依然用了 1 天半 (30 小時) 計算完 1 千多組。autoML 的狀況更為嚴重，因為下載的元件只支援單集群 (cluster) 演算，在平行處理上受到限制，而 LSTM 亦同。



(A)　SVM　平行運算耗時



(B) SVM 平行運算的運作狀況

資料來源：本研究自行整理。

圖三十三、SVM 平行運算的耗時狀況

2. 本案最佳的方法是自動化機器學習，不論是從自我落後結構的 15 種組合，或混合落後的 1,005 組，預測平均法皆可提高其精準度，這也是本研究案對於機器學習時間序列預測的建議作法。過多的投入變數恐產生大而無當的成本，且效益也不會比較好。能夠事先訓練好一個小集合，並產生多個預測的平均，應該很有助益。

3. 多種落後期變數的混合，雖然須測試大量的投入變數組合使計算耗時，但根據本文與附錄三的說明，利用 PCA 或 ICA 將資料降維，無法得到

較好的結果，因此這份研究計畫把重心放在原始數據的演算。由預測平均法的表現看來，減少運算量，提高預測表現，應該是一個可行方法。

4. SVM 和 LSTM 則容易產生持續性的高估或低估，這或許是因為樣本數太少所致，當樣本數夠多時，這兩個方法應該可以表現更好。
5. 用機器學習修正時間序列模型的預測誤差 (如第 7 章在 ARIMA 計算過程添加機器學習模式) 雖屬可行，但仍有改善空間。

我們已在本研究案做了一些嘗試，可惜經濟成長季資料觀察值僅 232 個，對於號稱大數據的學習演算法而言，訓練和交叉驗證的設計都受到極嚴重限制，學習不足使其預測優勢難以被確認，這是值得注意的地方。

最後，使用間接混頻的方式 (如 MIDAS)，結合機器學習對高頻月資料 (如工業生產指數) 進行預測，亦即用機器學習訓練高頻預測模型，再用經濟計量方法預測經濟成長。[12]本計畫就此給予高度評價，應對整合機器學習到計量經濟學有廣大貢獻，可做為未來研究方向。

[12] 感謝徐士勛教授的建議。

參考文獻

吳易樺，黃朝熙，劉子衙 (2014)。時間序列模型對我國產業成長預測之優劣比較，應用經濟論叢 96，35-68。

林向愷，黃裕烈，管中閔 (1998)。景氣循環轉折點認定與經濟成長率預測，經濟論文叢刊 26 (4)，431-457。

徐士勛，管中閔，羅雅惠 (2005)。以擴散指標為基礎之總體經濟預測，臺灣經濟預測與政策 36 (1)，1-28。

陳宜廷，徐士勛，劉瑞文，莊額嘉 (2011)。經濟成長率預測之評估與更新，經濟論文叢刊 39 (1)，1-44。

葉錦徽，程英賓，王景南 (2012)。經驗解構法與臺灣經濟成長之預測，經濟論文 40 (4)，559-598。

Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, *29*(5-6), 594-621.

Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications*, *113*, 457-480.

Bergmeir, C., Hyndman, R. J., & Benítez, J. M. (2016). Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation. *International Journal of Forecasting*, *32*(2), 303-312.

Beyca, O. F., Ervural, B. C., Tatoglu, E., Ozuyar, P. G., & Zaim, S. (2019). Using machine learning tools for forecasting natural gas consumption in the province of Istanbul. *Energy Economics*, *80*, 937-949.

Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

Bolhuis, M., & Rayner, B. (2020). *Deus ex Machina? A Framework for Macro Forecasting with Machine Learning* (No. 20/45). International Monetary Fund.

Bontempi, G., Taieb, S. B., & Le Borgne. Y. A. (2013). Machine learning strategies for time series forecasting. In Marie-Aude Aufaure and Esteban Zimányi (eds.) *Business Intelligence*, pp. 62-77, Spring-Verlag.

Cepni, O., Guney, I. E., & Swanson, N. R. (2020). Forecasting and nowcasting emerging market GDP growth rates: The role of latent global economic policy uncertainty and macroeconomic data surprise factors. *Journal of Forecasting*, *39*(1), 18-36.

Chakraborty, C., & Joseph, A. (2017). Machine learning at central banks. Staff working paper No. 674, Bank of England.

Chatzis, S. P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., & Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications*, *112*, 353-371.

Chen, S. W. & Lin, J. L. (2000). Modelling business cycles in Taiwan with time varying Markov switching models. *Academia Economic Papers*, *28*(1), 17-42.

De Livera, A. M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American statistical association*, *106*(496), 1513-1527.

De Stefani, J., Caelen, O., Hattab, D., & Bontempi, G. (2017). Machine learning for multi-step ahead forecasting of volatility proxies. In MIDAS@ PKDD/ECML,17-28.

Diebold, F. X., & Shin, M. (2019). Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives. *International Journal of Forecasting*, *35*(4), 1679-1691.

Ediger, V. Ş., & Akar, S. (2007). ARIMA forecasting of primary energy demand by fuel in Turkey. *Energy Policy*, *35*(3), 1701-1708.

Feng, H., & Liu, J. (2003). A SETAR model for Canadian GDP: Non-linearities and forecast comparisons. *Applied Economics*, *35*(18), 1957-1964.

Feng, L., & Zhang, J. (2014). Application of artificial neural networks in tendency forecasting of economic growth. *Economic Modelling*, *40*, 76-80.

Franses, P. H., & Van Dijk, D. (2000). *Non-linear Time Series Models in Empirical Finance*, Cambridge University Press.

Gilliland, M. (2020). The value added by machine learning approaches in forecasting. *International Journal of Forecasting*, *36*(1), 161-166.

Gogas, P., Papadimitriou, T., Matthaiou, M., & Chrysanthidou, E. (2015). Yield curve and recession forecasting in a machine learning framework. *Computational Economics*, *45*(4), 635-645.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223-2273.

Hafezi, R., Shahrabi, J., & Hadavandi, E. (2015). A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price. *Applied Soft Computing*, *29*, 196-210.

Hall, A. S. (2018). Machine learning approaches to macroeconomic forecasting. *The Federal Reserve Bank of Kansas City Economic Review*, *103*(63), 2.

Hamzaçebi, C., Akay, D., & Kutay, F. (2009). Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert Systems with Applications*, *36*(2), 3839-3844.

Hansen, B. E. (2007). Least squares model averaging. *Econometrica*, *75(4)*, 1175-1189.

Hansen, B. E., & Racine, J. S. (2012). Jackknife model averaging. *Journal of Econometrics*, *167*(1), 38-46.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

Huang, C. H. (1999). Phases and characteristics of Taiwan's business cycles: A Markov switching analysis. *Taiwan Economic Review*, *27*(2), 515-540.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., & Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *454*(1971), 903-995.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. 2nd edition, Available at https://otexts.com/fpp2/.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679-688.

Hyndman, R. J., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., & Yasmeen, F. (2018). forecast: Forecasting functions for time series and linear models. Available at https://rdrr.io/cran/forecast/.

Iworiso, J., & Vrontos, S. (2020). On the directional predictability of equity premium using machine learning techniques. *Journal of Forecasting*, *39*(3), 449-469.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013) *An Introducton to Statistical Learning*. Springer Science & Business Media.

Karpathy, A. (2015). The Unreasonable Effectiveness of Recurrent Neural Networks. Available at http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Kline, D. M. (2004). Methods for multi-step time series forecasting neural networks. In *Neural Networks in Business Forecasting* (pp. 226-250). IGI Global.

Koenker, R. & Xiao, Z. (2006) Quantile Autoregression. *Journal of the American Statistical Association*, 101(475), 980-990.

Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2017). Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1), 841-851.

Kuo, C. E., & Chen, G. T. (2020). Automatic Sleep Staging Based on a Hybrid Stacked LSTM Neural Network: Verification Using Large-Scale Dataset. *IEEE Access, 8*, 111837-111849.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 159-174.

Liao, J. C., & Tsay, W. J. (2020). Optimal multistep VAR forecast averaging. *Econometric Theory*, 1-28.

Makridakis, S. (1993). Accuracy measures: theoretical and practical concerns. *International Journal of Forecasting*, *9*(4), 527-529.

Makridakis, S., & Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International Journal of forecasting*, *16*(4), 451-476.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*(1), 54-74.

Medeiros, M. C., Vasconcelos, G. F., Veiga, Á., & Zilberman, E. (2019). Forecasting Inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics*, 1-22.

Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting*, *36*(1), 86-92.

Morlidge, S. (2014). Do forecasting methods reduce avoidable error? Evidence from forecasting competitions. *Foresight: The International Journal of Applied Forecasting*, (32), 34-39.

Olah, C. (2015). Understanding LSTM Networks, Available at https://colah.github.io/posts/2015-08-Understanding-LSTMs/. ◦

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, *42*(4), 2162-2172.

Peker, M. (2016). An efficient sleep scoring system based on EEG signal using complex-valued machine learning algorithms. *Neurocomputing*, 207, 165-177.

Richardson, A., & Mulder, T. (2018). Nowcasting New Zealand GDP using machine learning algorithms. CAMA Working Paper No. 47/2018. Available at SSRN: https://ssrn.com/abstract=3256578.

Risse, M. (2019). Combining wavelet decomposition with machine learning to forecast gold returns. *International Journal of Forecasting*, *35*(2), 601-615.

Saad, E. W., Prokhorov, D. V., & Wunsch, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on neural networks*, *9*(6), 1456-1470.

Serinaldi, F. (2011). Distributional modeling and short-term forecasting of electricity prices by generalized additive models for location, scale and shape. *Energy Economics*, *33*(6), 1216-1226.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *36*(1), 75-85.

Stock, J. H., & Watson, M. W. (1998). *A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series* (No. w6607). National Bureau of Economic Research.

Stock, J. H., & Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, *20*(2), 147-162.

Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, *54*(8), 799-805.

Tiffin, A. (2016). Seeing in the dark: a machine-learning approach to nowcasting in Lebanon. IMF Working Paper No. 16/56. Available at SSRN: https://ssrn.com/abstract=2770291.

Tiffin, A. (2019). Machine learning and causality: the impact of financial crisis on growth. IMF Working Paper No. 19/228. Available at SSRN: https://ssrn.com/abstract=3496706

Torres, J. F., Galicia, A., Troncoso, A., & Martínez-Álvarez, F. (2018). A scalable approach based on deep learning for big data time series forecasting. *Integrated Computer-Aided Engineering*, *25*(4), 335-348.

Tsay, R. S. (2010). *Analysis of Financial Time Series*, John Wiley & Sons Inc.

Ülke, V., Sahin, A., & Subasi, A. (2018). A comparison of time series and machine learning models for inflation forecasting: empirical evidence from the USA. *Neural Computing and Applications*, *30*(5), 1519-1527.

Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer science & business media.

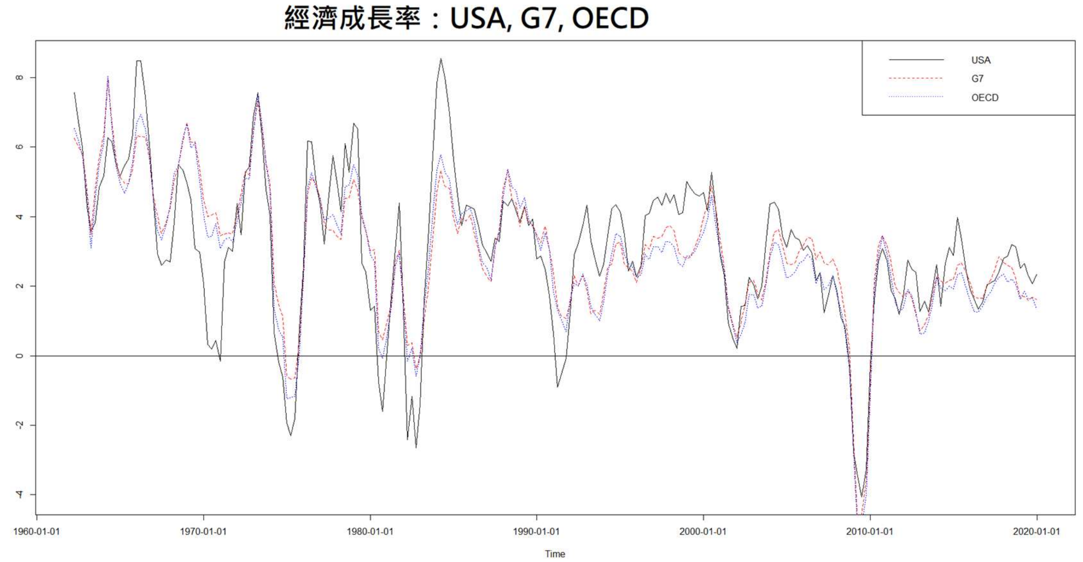
Wei, L. Y. (2016). A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting. *Applied Soft Computing*, *42*, 368-376.

Wood, S. N., & Augustin, N. H. (2002). GAMs with integrated model selection using penalized regression splines and applications to environmental modelling. *Ecological modelling*, *157*(2-3), 157-177.

Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017a). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, *11*(2), 68-75.

Zhao, Y., Li, J., & Yu, L. (2017b). A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, *66*, 9-16.

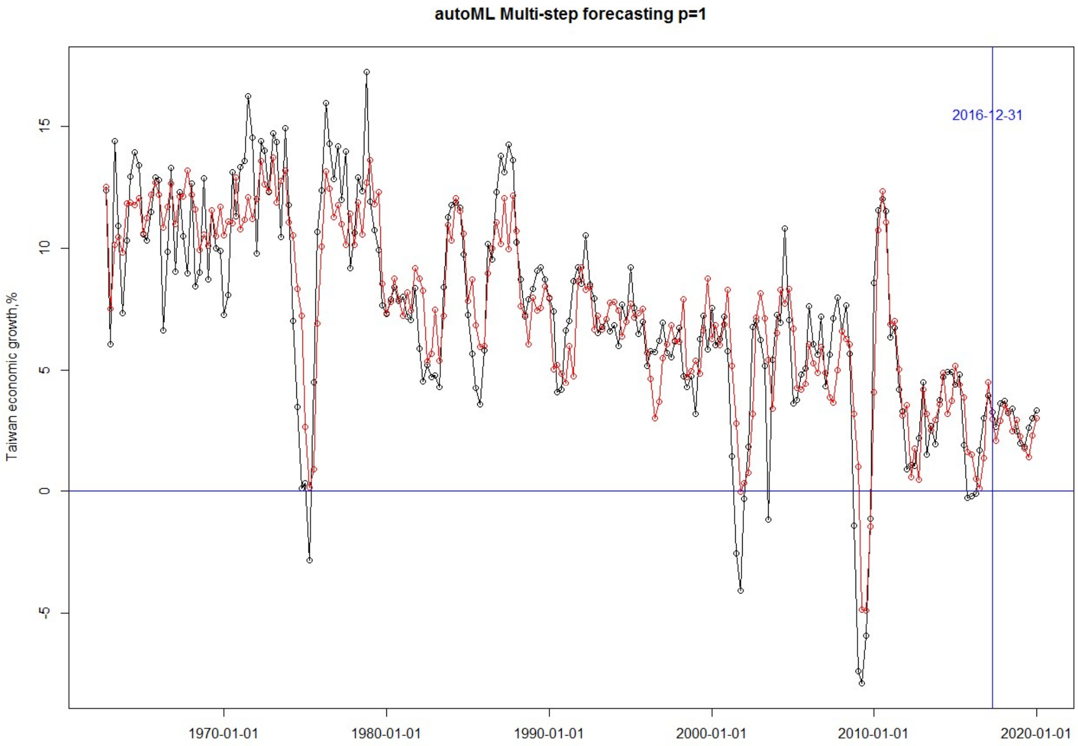
附錄一 **6.2** 節相關附圖

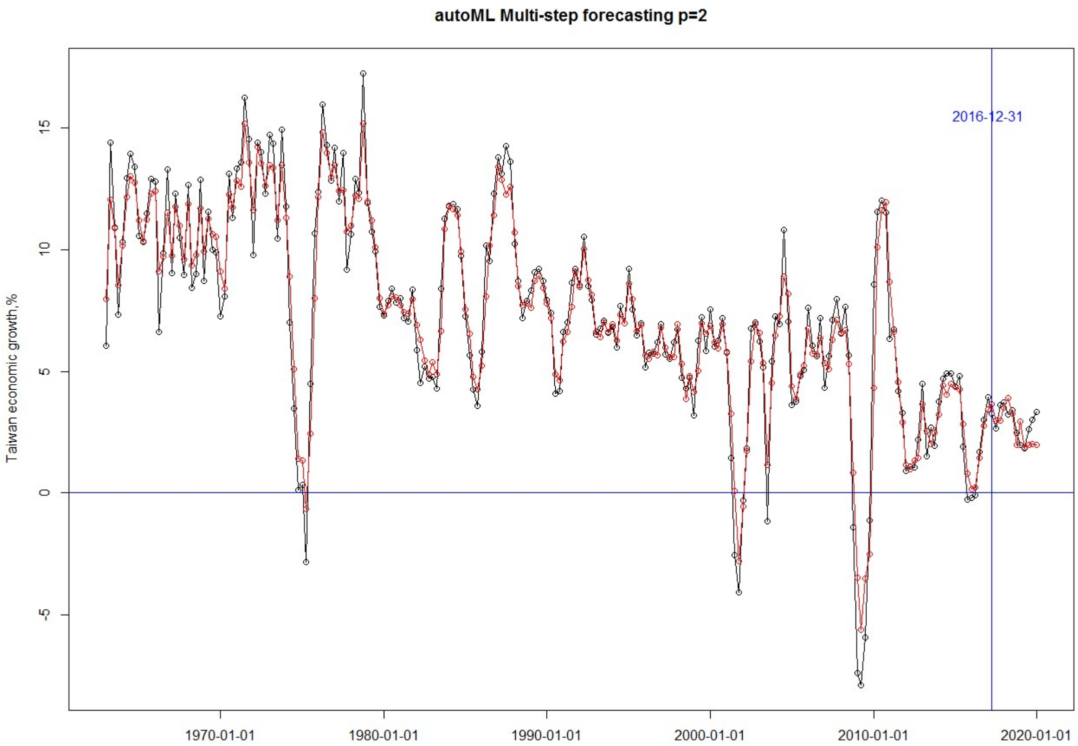


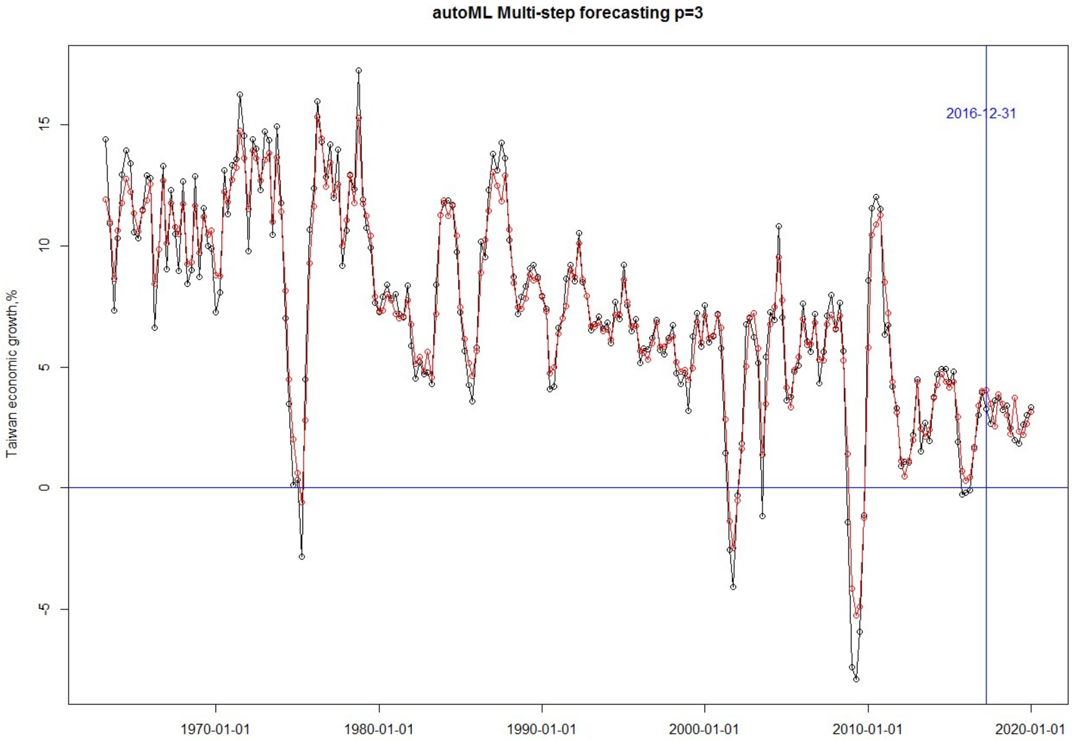
資料來源：本研究自行整理。

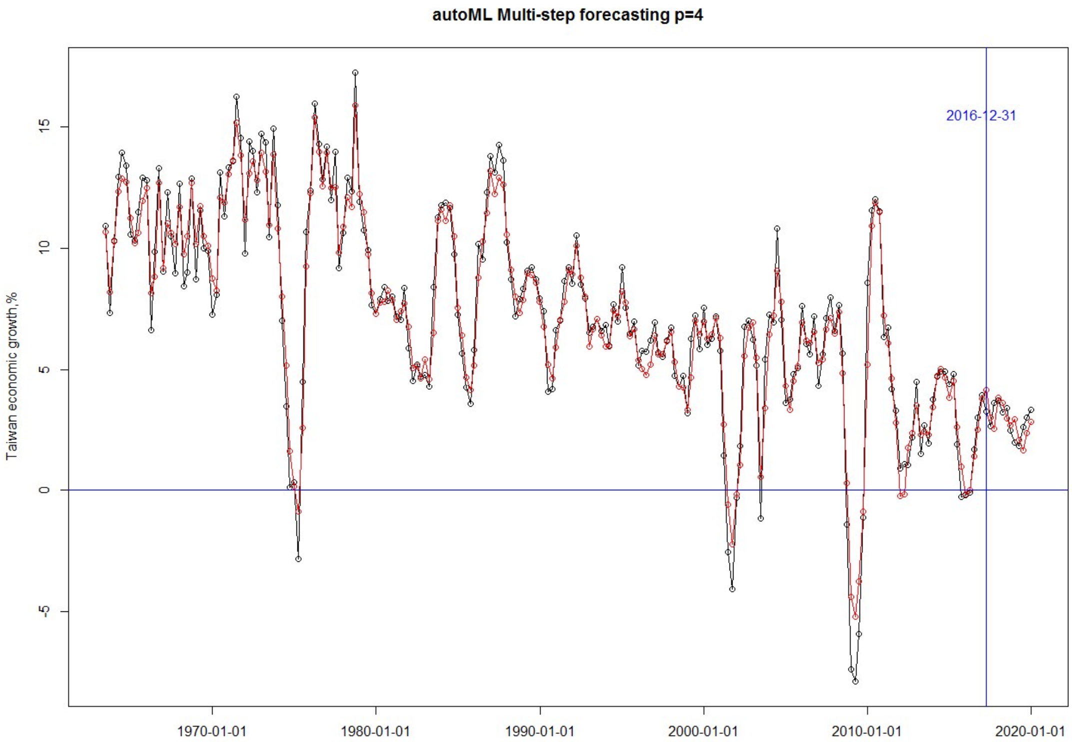
因為 autoML 會自動產生時間特徵，包括季節結構和時間趨勢，我們就沒有另外做設定比較。其餘 SVM 和 LSTM 各有 16 張圖。

A1-1 autoML 4 個模式預測表現圖



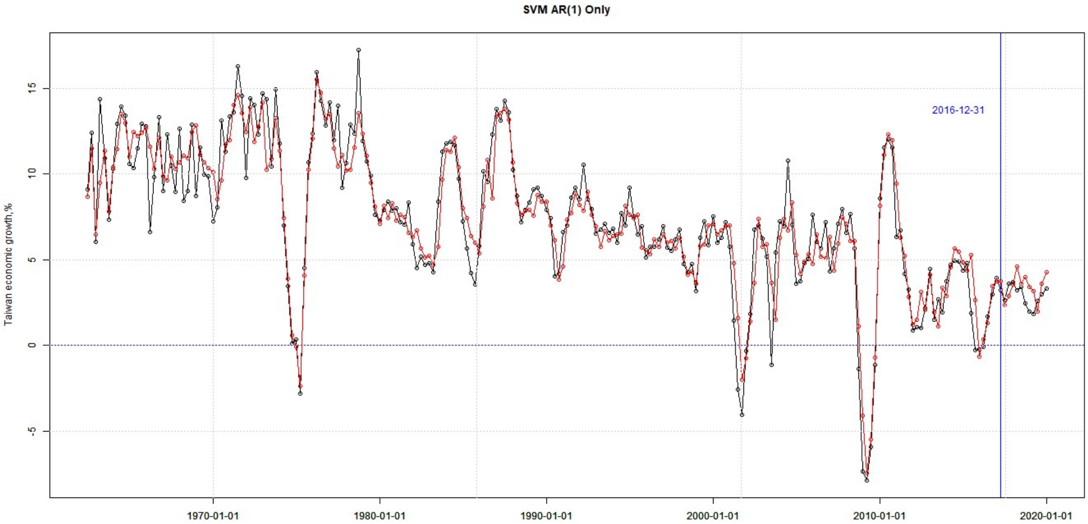


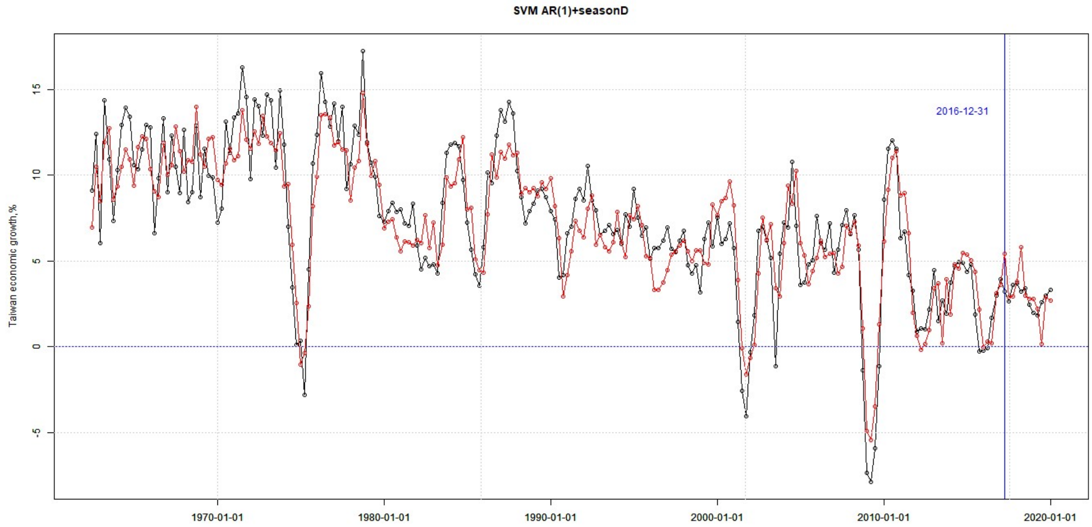


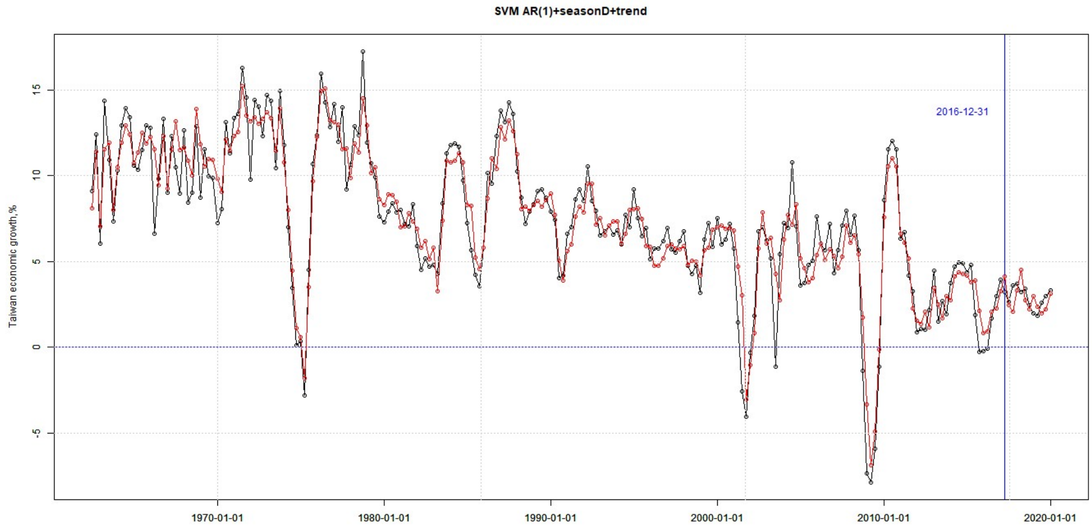


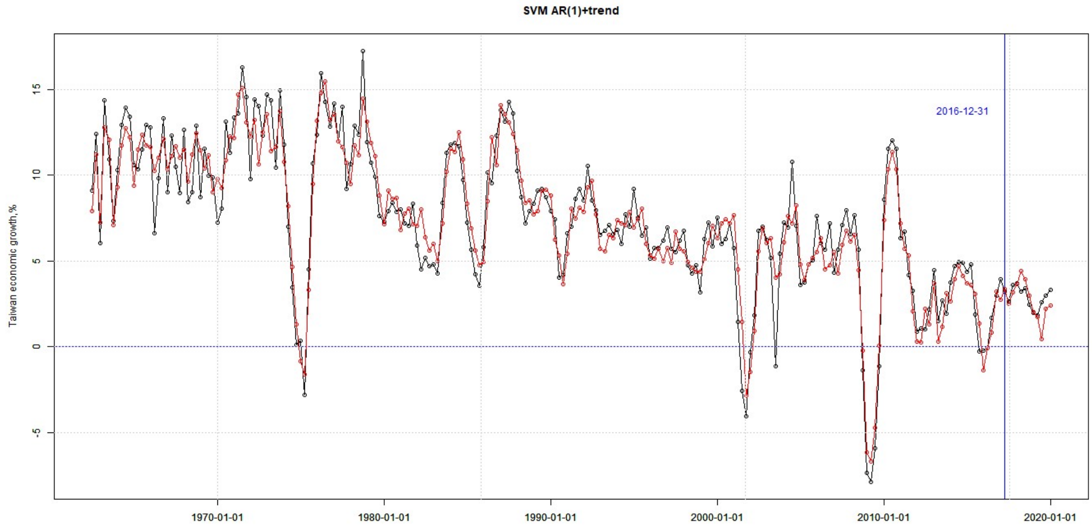
資料來源：本研究自行整理。

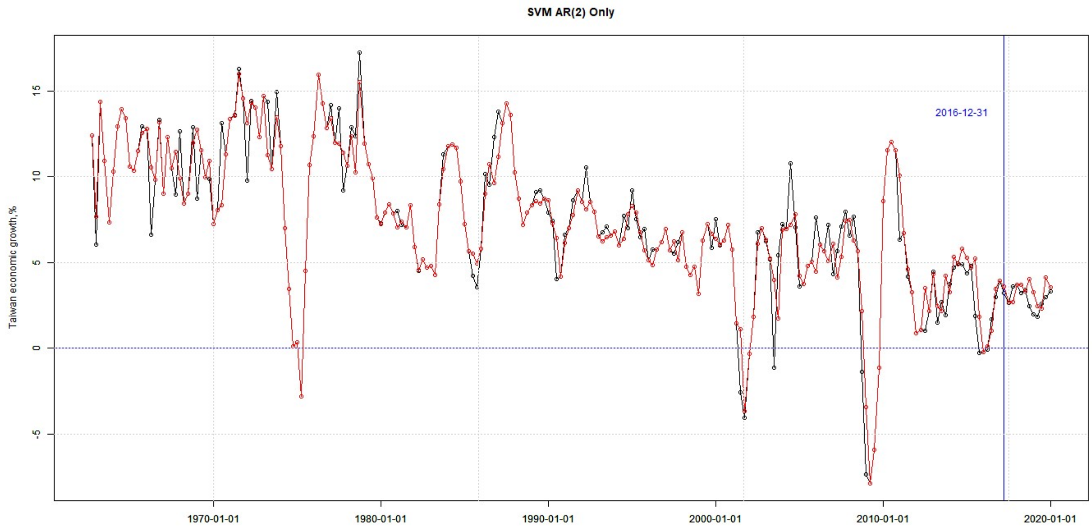
A1-2 SVM 16 個模式預測表現圖

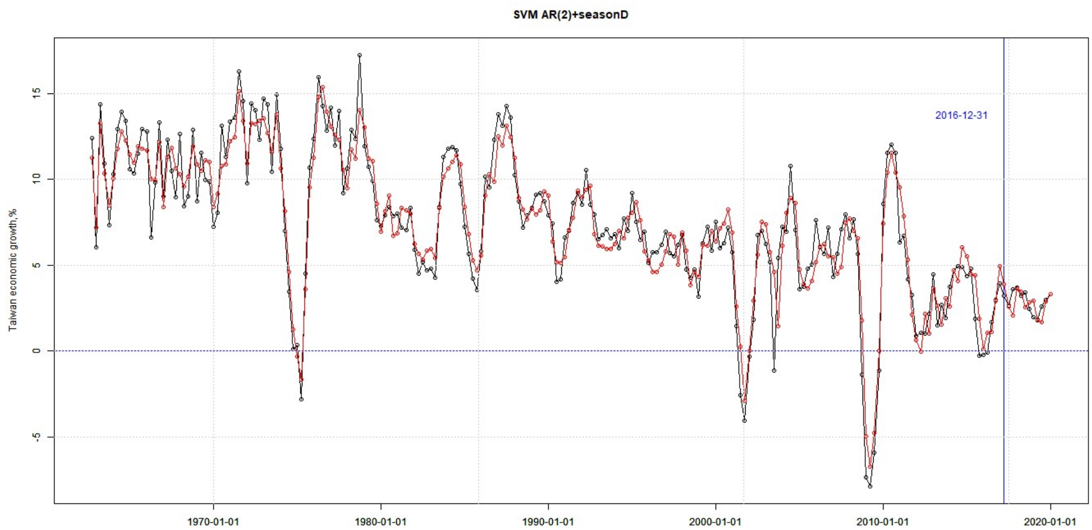


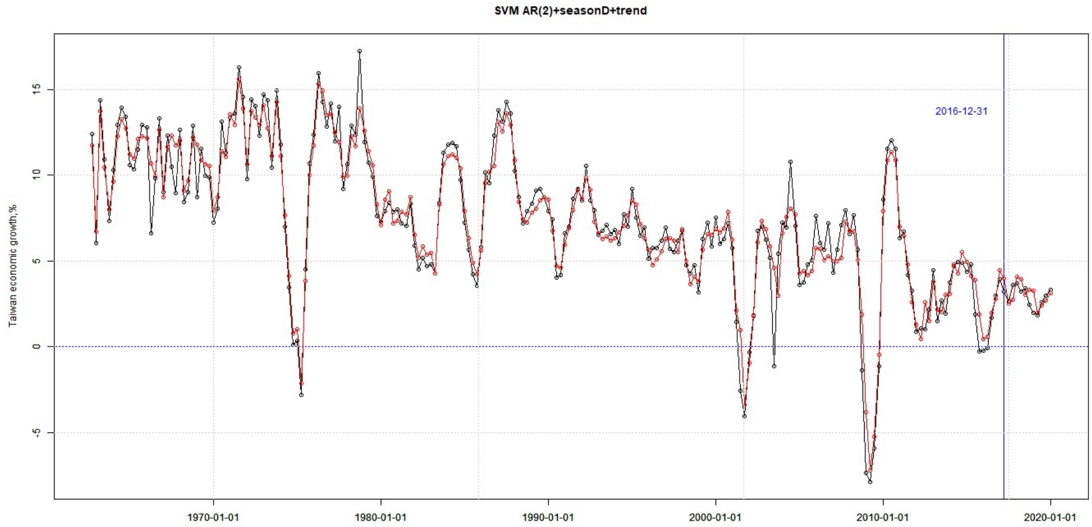

SVM AR(1)+seasonD
Taiwan economic growth, %
2016-12-31
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

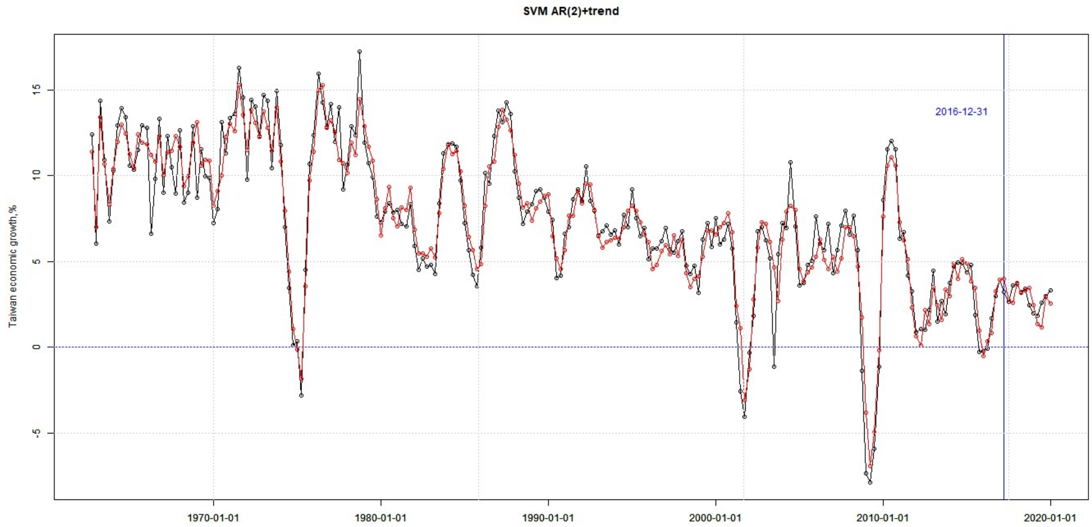

SVM AR(1)+seasonD+trend
Taiwan economic growth, %
2016-12-31
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

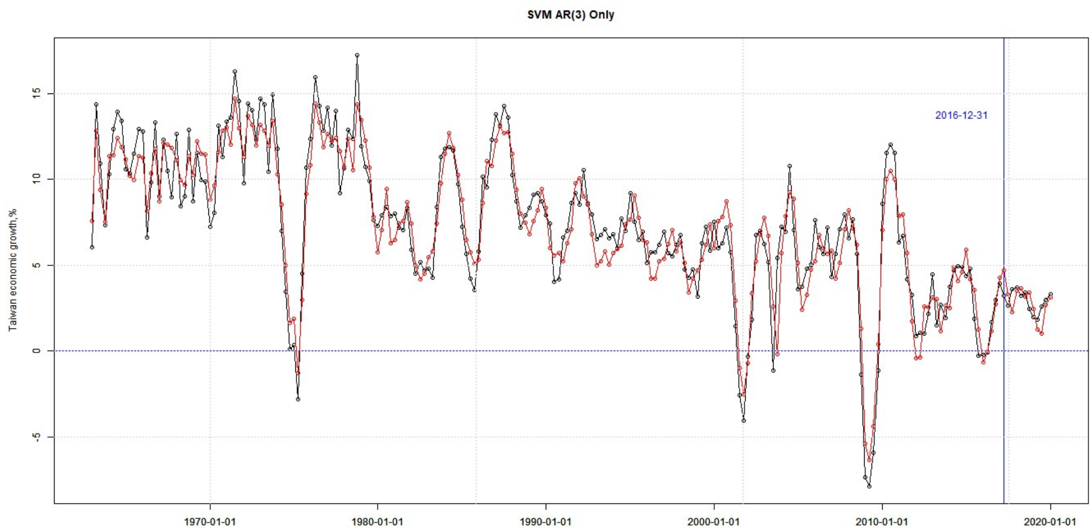

SVM AR(1)+trend
Taiwan economic growth, %
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

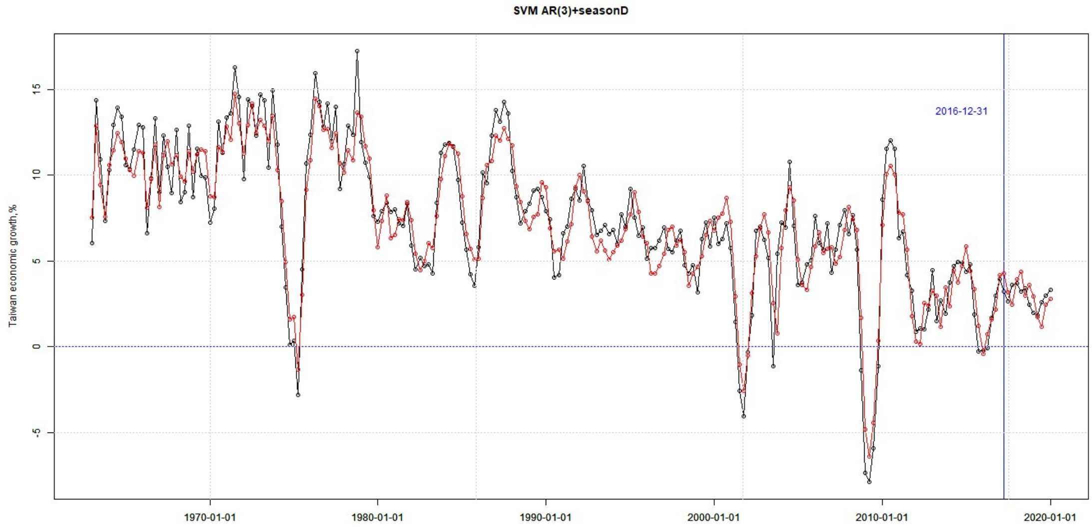

SVM AR(2) Only
Taiwan economic growth, %
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

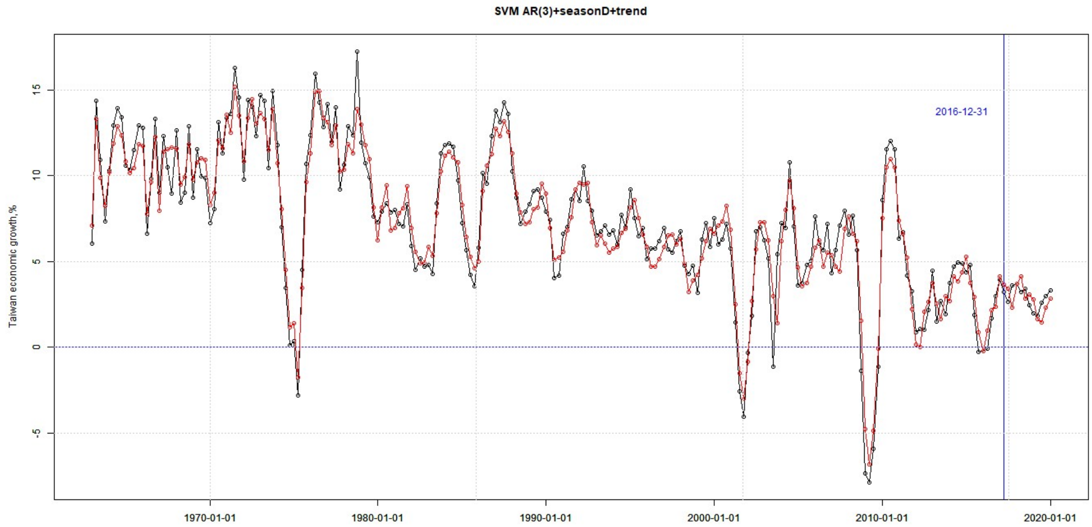

SVM AR(2)+seasonD
Taiwan economic growth, %
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

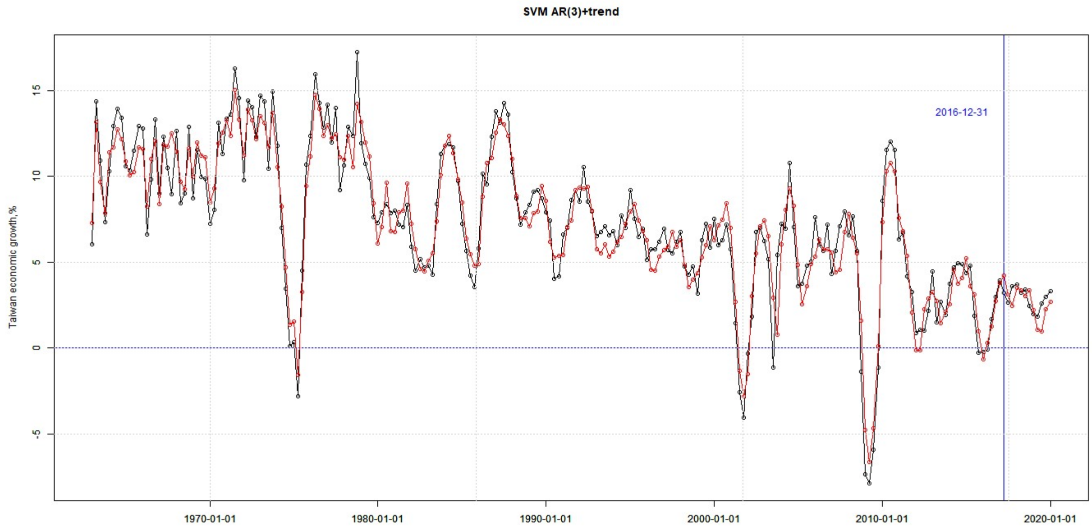

SVM AR(2)+seasonD+trend
2016-12-31
Taiwan economic growth, %
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

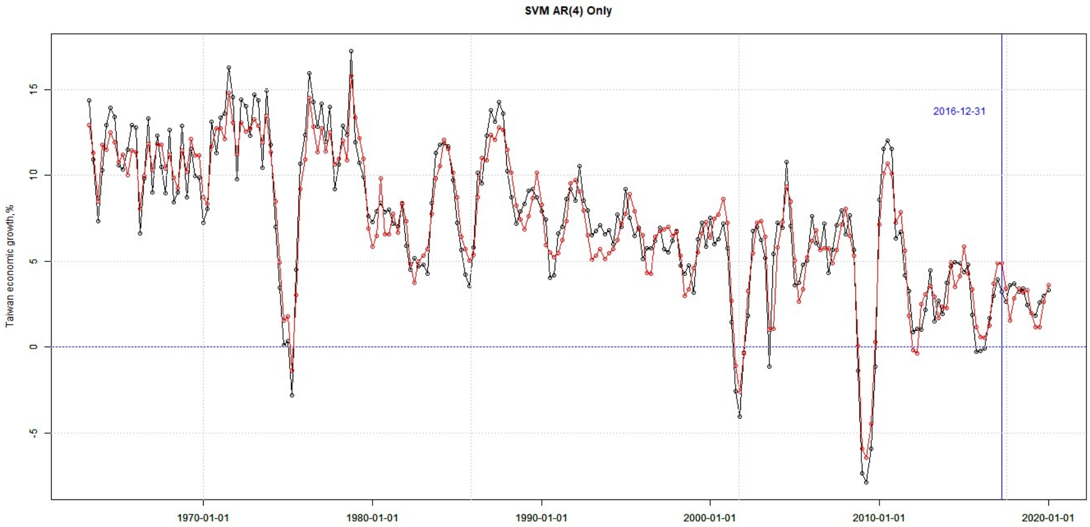

SVM AR(2)+trend
2016-12-31
Taiwan economic growth, %
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

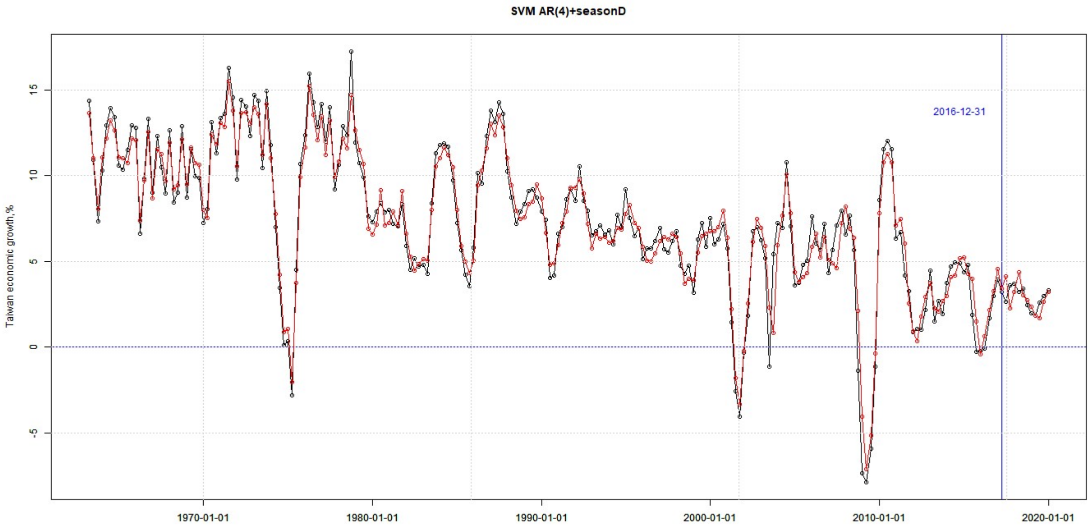

SVM AR(3) Only
2016-12-31
Taiwan economic growth, %
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

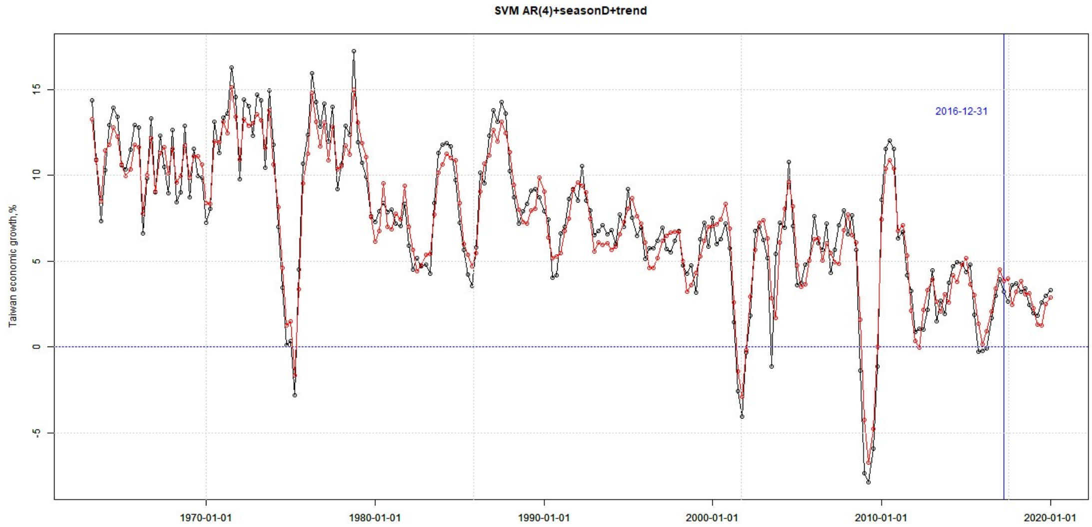

SVM AR(3)+seasonD
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

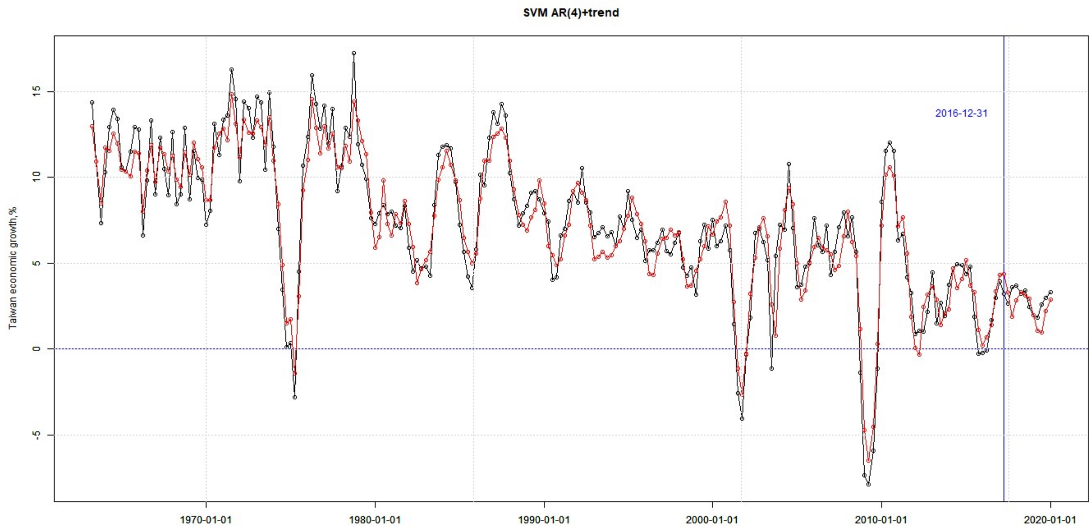

SVM AR(3)+seasonD+trend
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01


SVM AR(3)+trend
Taiwan economic growth,%
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01


SVM AR(4) Only
Taiwan economic growth, %
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

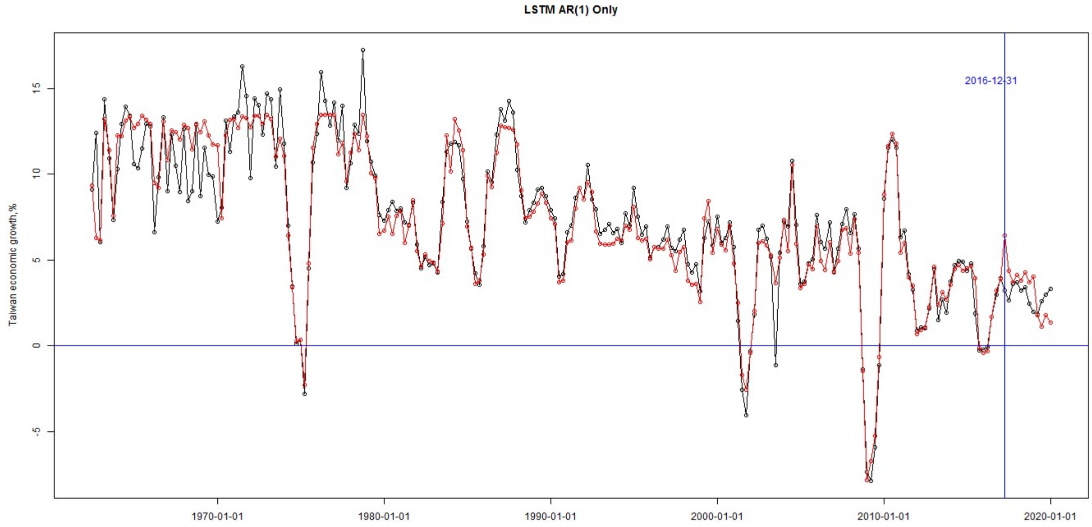

SVM AR(4)+seasonD
Taiwan economic growth, %
2016-12-31
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

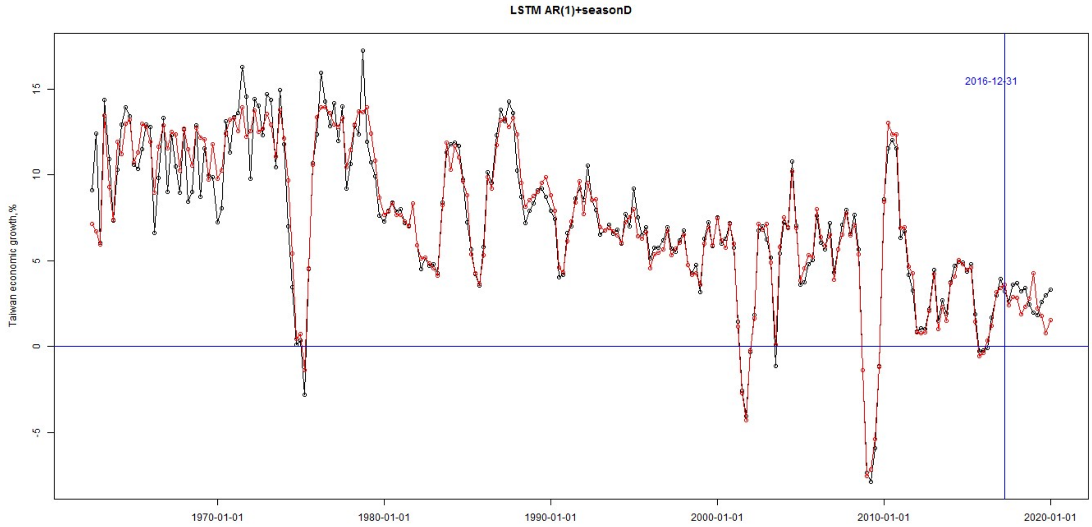




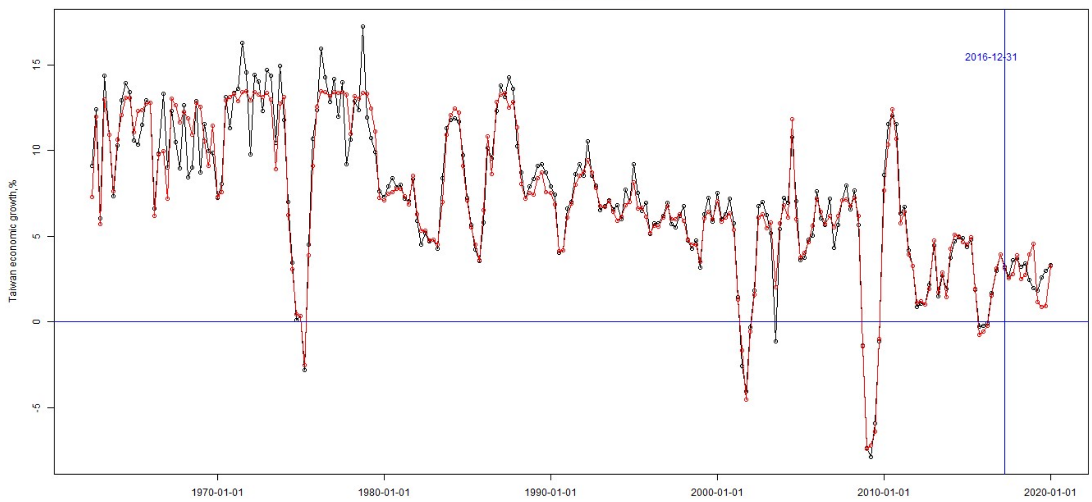
資料來源：本研究自行整理。

A1-3 LSTM 16 個模式預測表現圖

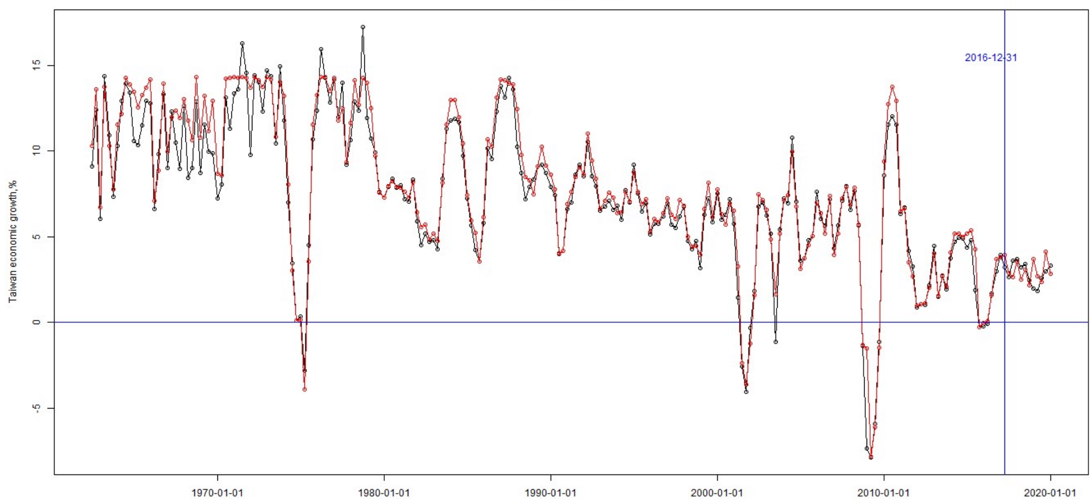




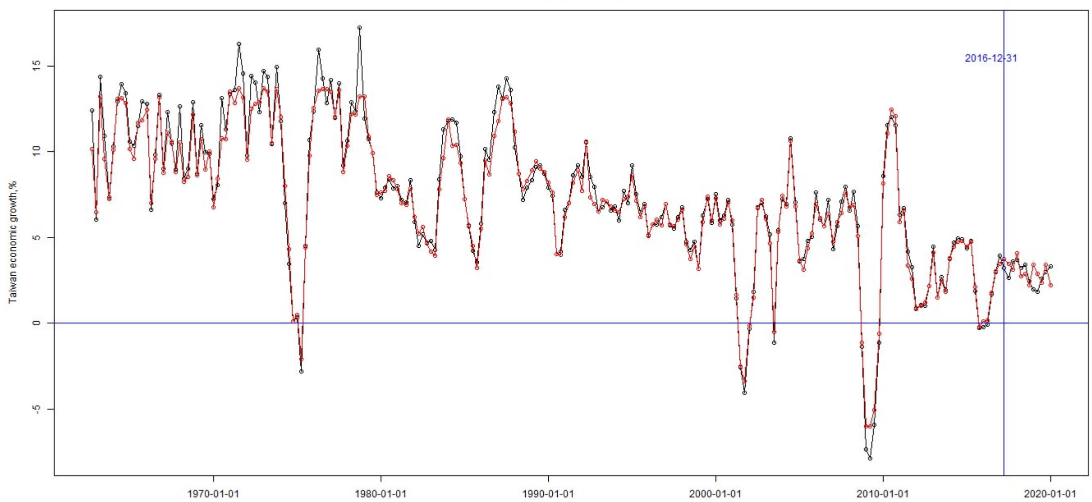
LSTM AR(1)+seasonD+trend

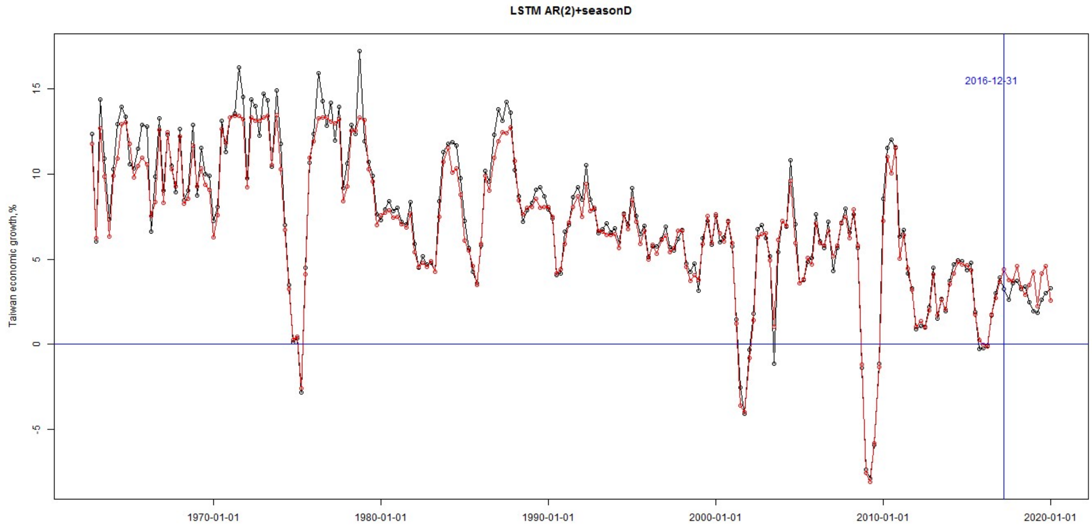


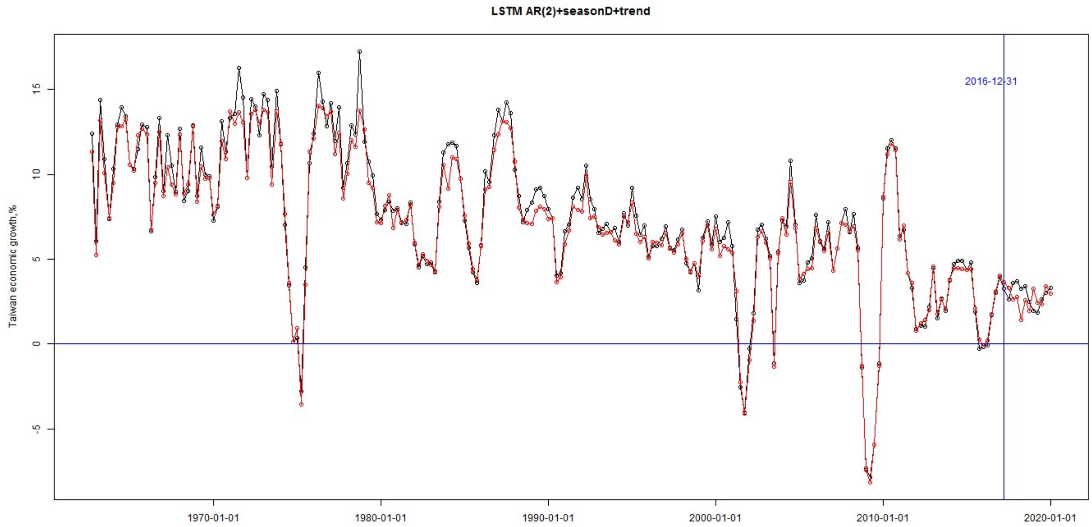
LSTM AR(1)+trend

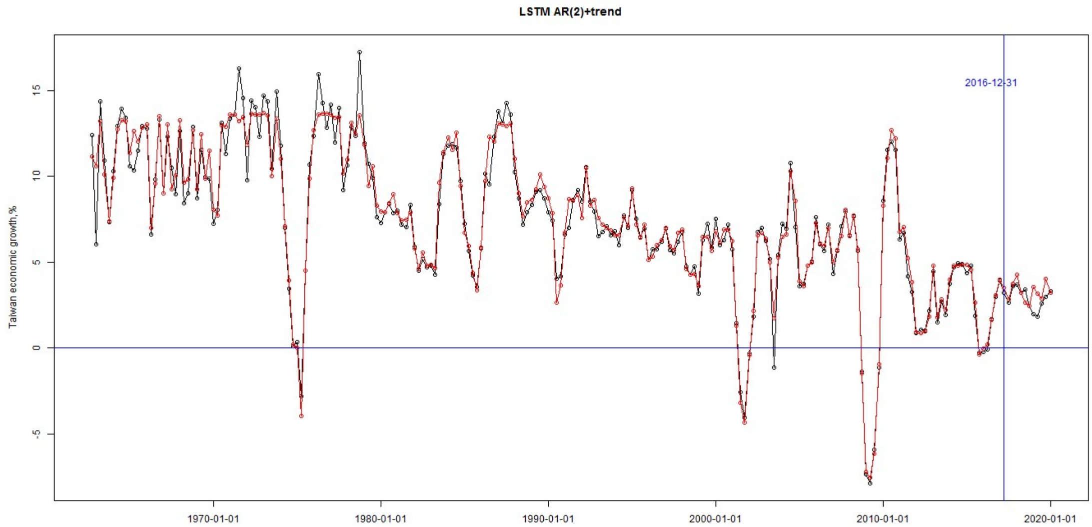


LSTM AR(2) Only

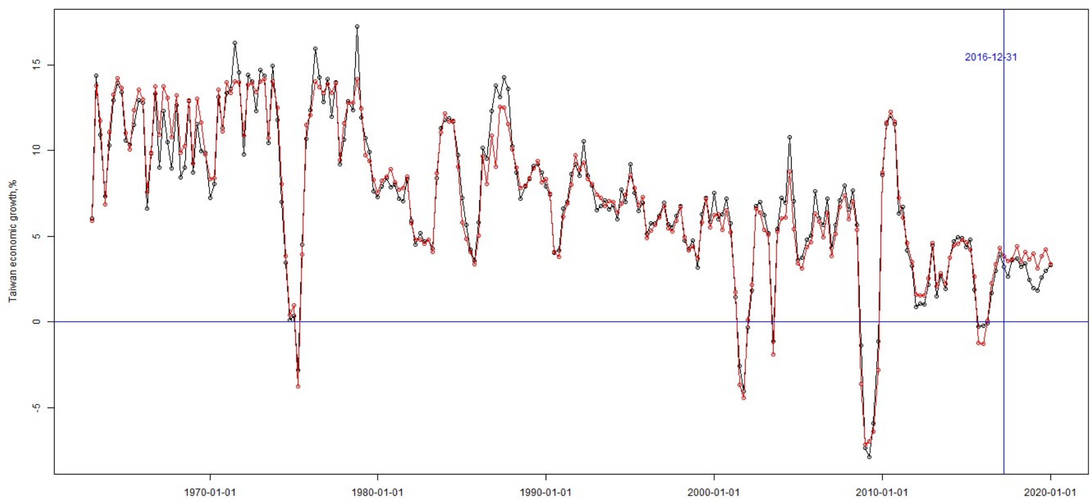



LSTM AR(2)+seasonD
2016-12-31
Taiwan economic growth, %
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

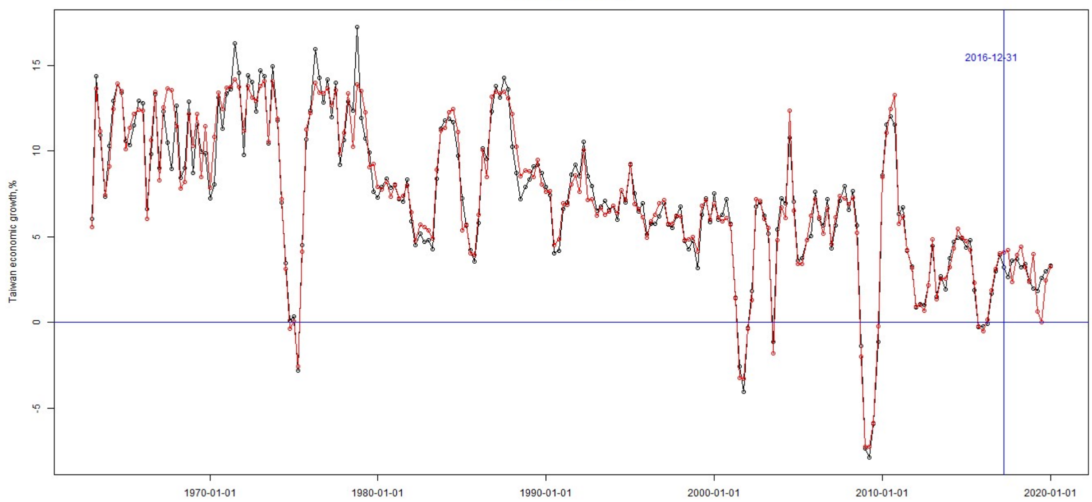

LSTM AR(2)+seasonD+trend
2016-12-31
Taiwan economic growth, %
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01


LSTM AR(2)+trend
2016-12-31
Taiwan economic growth, %
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01

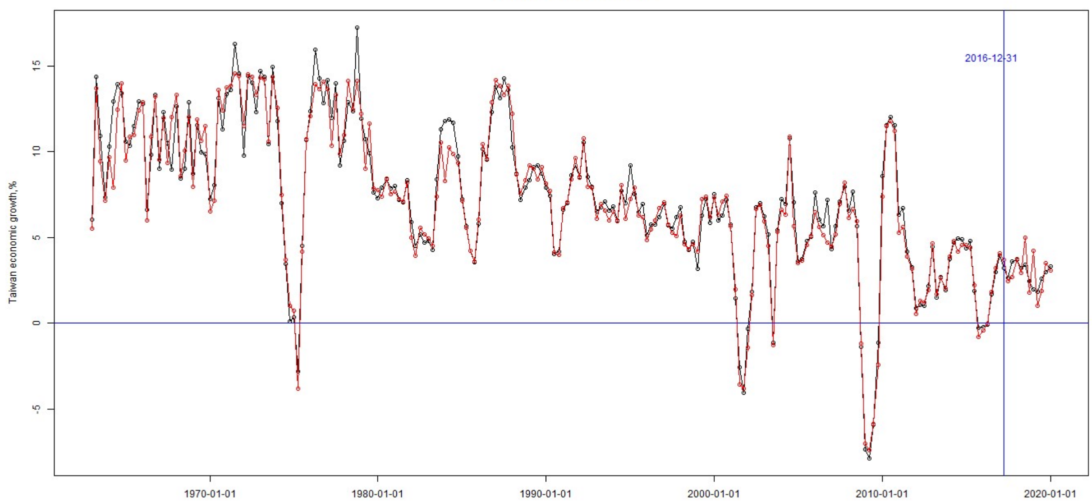
LSTM AR(3) Only

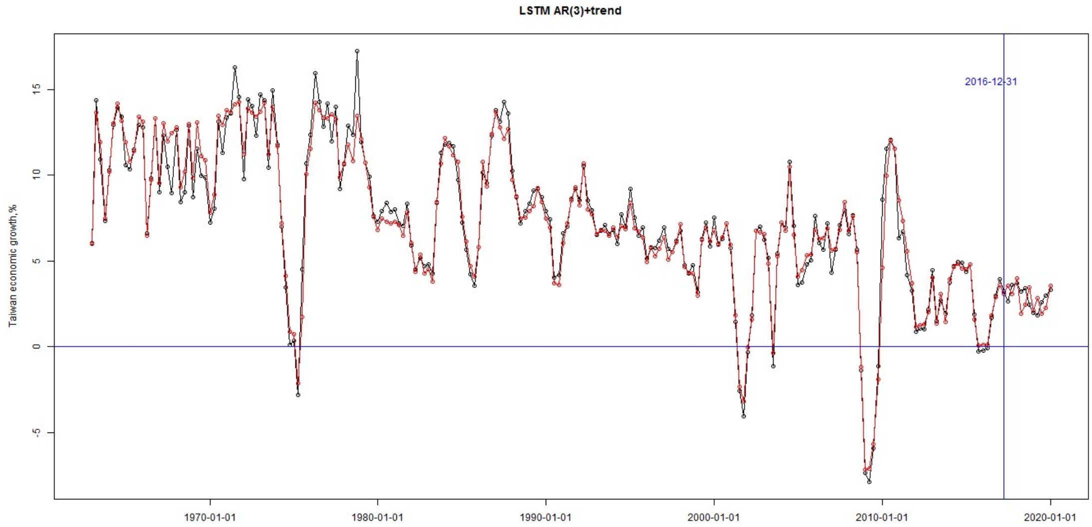


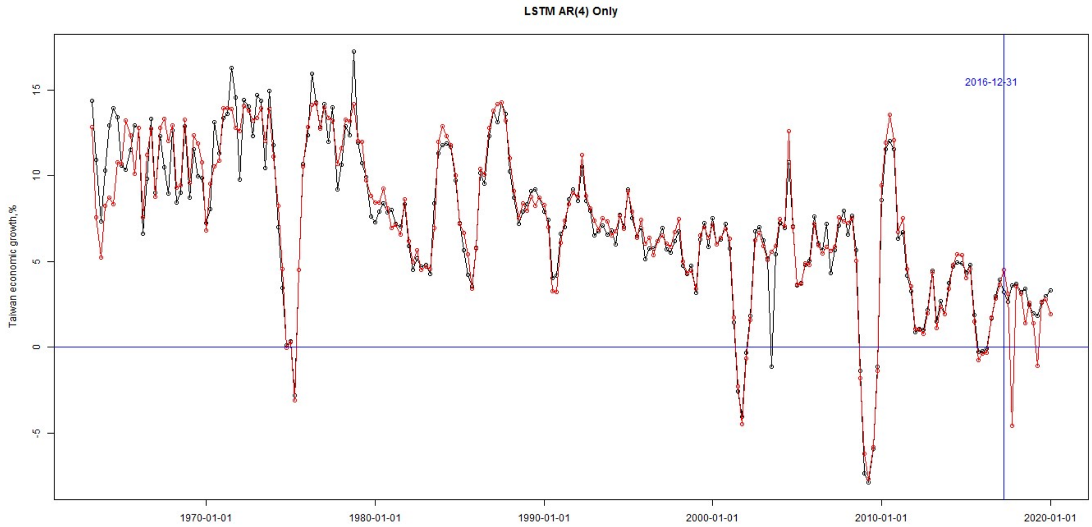
LSTM AR(3)+seasonD

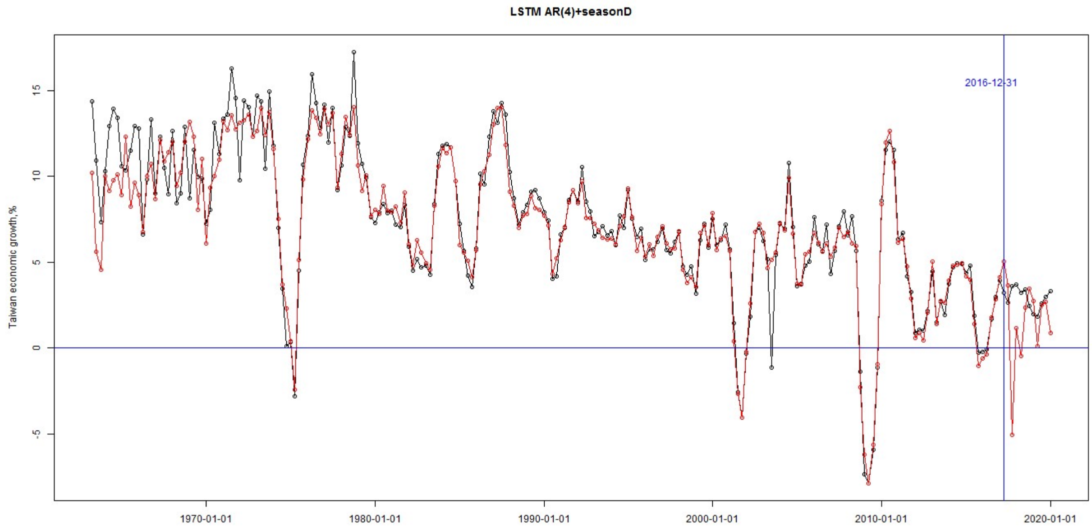


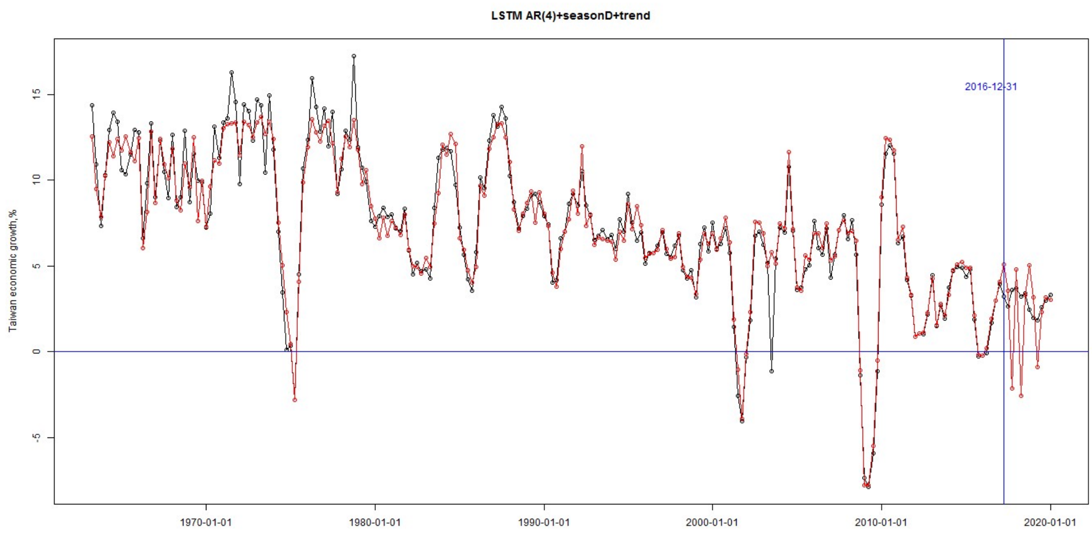
LSTM AR(3)+seasonD+trend

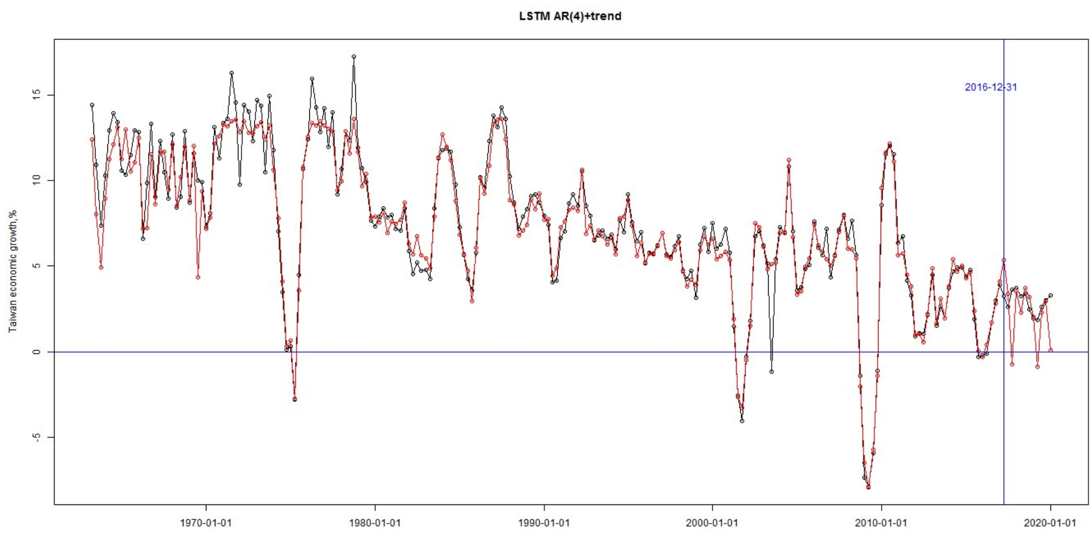



LSTM AR(3)+trend
2016-12-31
Taiwan economic growth,%
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01


LSTM AR(4) Only
2016-12-31
Taiwan economic growth,%
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01


LSTM AR(4)+seasonD
2016-12-31
Taiwan economic growth,%
15
10
5
0
-5
1970-01-01
1980-01-01
1990-01-01
2000-01-01
2010-01-01
2020-01-01





資料來源：本研究自行整理。

附錄二 對 **2020Q1-2021Q4** 未來八季預測

因應外部審查人及委託單位之建議，本附錄亦實證列出未來經濟成長率之預測結果。本文執行 3 個模式：SVM，autoML 和傳統時間序列模型。在 1962Q1-2019Q4 做模型訓練，然後以遞迴多步(Recursive multistep)產生 8 季的未來預測。基於此，所有的模型皆可以如此產生未來預測的參考，甚或將之平均。所有的機器學習模型，都是從歷史資料訓練的學習，對於第一次發生的事件衝擊，難以有效預測；如 2020 前兩季受冠狀肺炎疫情影響，是過去所沒有的外生衝擊(external shocks)，再厲害的演算法也無法預測到這種衝擊。因此，以下預測與實際必有大幅度偏差，我們執行的結果整理如下。當考慮額外解釋變數時，我們考慮 6.2 節使用的 5 個國內變數，分別為商品及服務輸出、商品及服務輸入、工業生產指數與台幣兌美元匯率(分為季平均與季底值)。以上 5 筆解釋變數資料皆轉化為年增率。

A2-1. 無額外解釋變數

(1) SVM AR(P) 的 **4** 個模式的經濟成長率預測

首先，表 A 2-1.1 指出，單單只有 AR(1)，樣本內訓練的結果是最差的。只要有添加季節啞變數或時間趨勢，預測配適就會有大幅度改善。整個 SVM 從 AR(2)到 AR(4)的結果，列於表 A 2-1.2 到表 A 2-1.4。

表 A 2-1.1 經濟成長率預測：**SVM AR(1)**

	AR(1) Only	**seasonDummy**	**trend**	**seasonDummy+trend**
樣本外預測值，**%**				
2020Q1	3.500	3.131	3.182	2.861
2020Q2	3.686	3.413	3.108	2.957
2020Q3	3.866	3.884	3.070	2.759
2020Q4	4.041	3.356	3.054	2.720
2021Q1	4.209	3.171	3.054	2.463
2021Q2	4.368	3.463	3.065	2.533
2021Q3	4.517	3.906	3.082	2.471
2021Q4	4.655	3.379	3.105	2.495
樣本內訓練績效				
RMSE	2.377	2.3823	2.1735	2.1522
MAE	1.7562	1.817	1.5912	1.5674
MAPE	0.6015	0.5498	0.5308	0.5766
ACF1	0.0581	0.0879	0.2139	0.1327
Theil's U	0.0079	0.0087	0.0072	0.0072

資料來源：本研究自行整理。

表 A 2-1.2 經濟成長率預測：SVM AR(2)

	AR(2) Only	seasonDummy	trend	seasonDummy+trend
樣本外預測值，%				
2020Q1	3.933	3.805	2.998	2.687
2020Q2	4.614	3.763	2.679	3.292
2020Q3	5.119	4.319	2.460	2.467
2020Q4	5.337	4.487	2.337	2.328
2021Q1	5.329	4.542	2.283	2.299
2021Q2	5.208	4.286	2.268	2.937
2021Q3	5.063	4.598	2.273	2.212
2021Q4	4.948	4.686	2.283	2.156
樣本內訓練績效				
RMSE	2.2152	2.1068	1.9705	1.8933
MAE	1.7012	1.5358	1.4391	1.4407
MAPE	0.5151	0.4899	0.501	0.4745
ACF1	-0.0425	0.0217	-0.0082	-0.0239
Theil's U	0.0086	0.0058	0.0106	0.0065

資料來源：本研究自行整理。

表 A 2-1.3 經濟成長率預測：SVM AR(3)

	AR(3) Only	seasonDummy	trend	seasonDummy+trend
樣本外預測值，%				
2020Q1	3.664	3.991	2.146	2.990
2020Q2	3.941	3.724	1.529	3.173
2020Q3	4.162	4.173	0.965	2.898
2020Q4	4.345	4.041	0.895	2.631
2021Q1	4.502	4.364	1.014	2.205
2021Q2	4.637	3.975	1.281	2.580
2021Q3	4.754	4.304	1.411	2.563
2021Q4	4.854	4.148	1.421	2.470
樣本內訓練績效				
RMSE	2.047	2.0029	1.9071	1.7676
MAE	1.5369	1.4177	1.5051	1.2147
MAPE	0.5057	0.5084	0.4624	0.4512
ACF1	0.0384	0.0244	0.0145	-0.0117
Theil's U	0.0085	0.0059	0.0073	0.0042

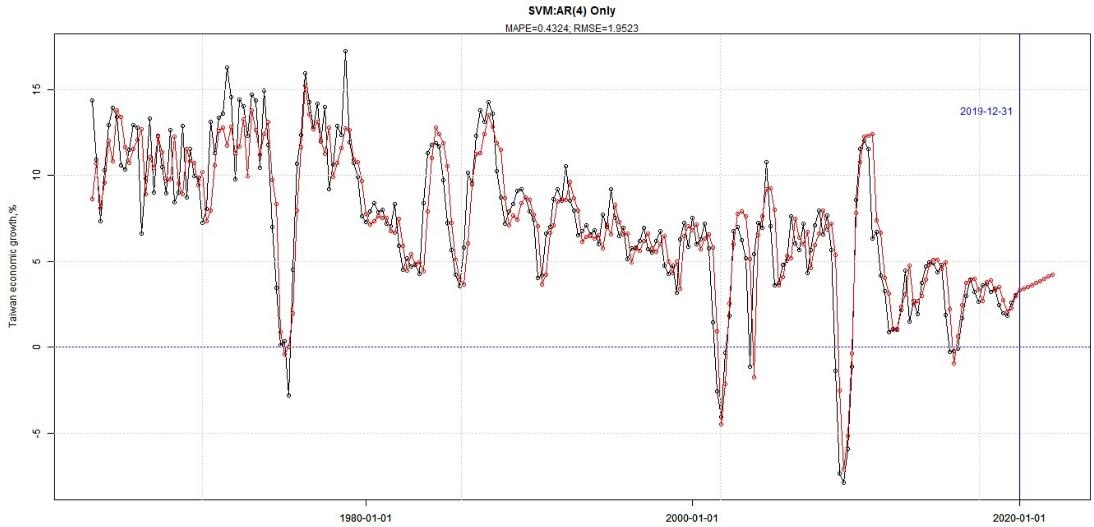
資料來源：本研究自行整理。

表 A 2-1.4 經濟成長率預測：SVM AR(4)

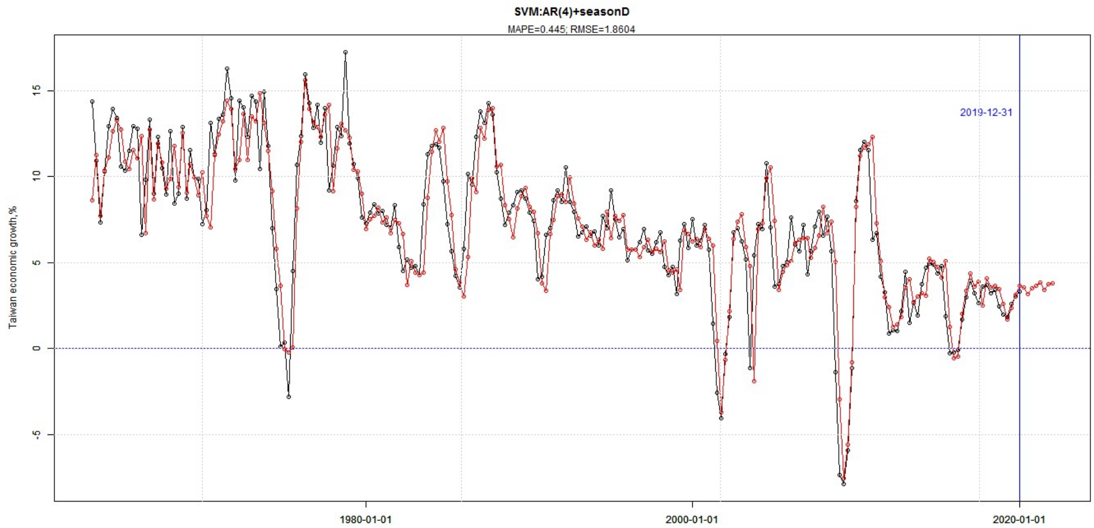
	AR(4) Only	seasonDummy	trend	seasonDummy+trend
樣本外預測值，%				
2020Q1	3.414	3.562	2.511	3.238
2020Q2	3.515	3.163	1.930	2.827
2020Q3	3.620	3.529	1.435	2.720
2020Q4	3.737	3.664	1.126	2.728
2021Q1	3.857	3.863	1.185	2.743
2021Q2	3.981	3.433	1.358	2.528
2021Q3	4.109	3.725	1.525	2.652
2021Q4	4.243	3.815	1.631	2.718
樣本內訓練績效				
RMSE	1.9523	1.8604	1.7736	**1.5922**
MAE	1.4017	1.2308	1.3617	0.9047
MAPE	0.4324	0.445	0.4732	**0.3187**
ACF1	0.1001	0.0248	0.0623	0.0206
Theil's U	0.0064	0.0048	0.0064	0.0029

資料來源：本研究自行整理。

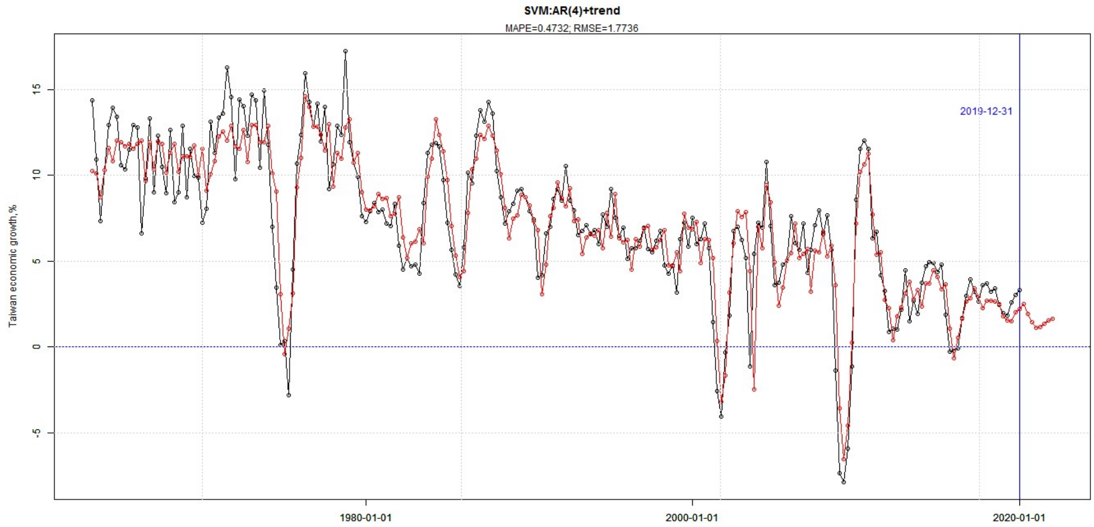
整體看起來，模型到 AR(4)時，所有的指標都是相對最小，加上季節和趨勢的虛擬變數則是四個模式的最小。因此，我們將表 A 2-1.4 的預測結果繪於圖 A 2-1.1 (A)到(D)的四張圖作為視覺參考。



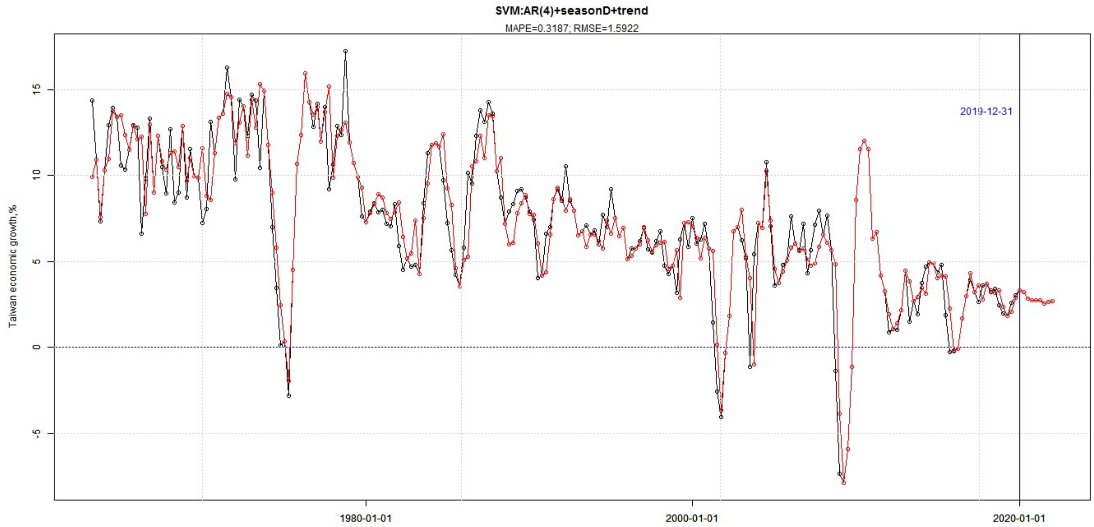
(A) AR(4)



(B) AR(4)+季節虛擬變數



(C) AR(4)+確定趨勢



(D) AR(4)+季節虛擬變數+確定趨勢

資料來源：本研究自行整理。

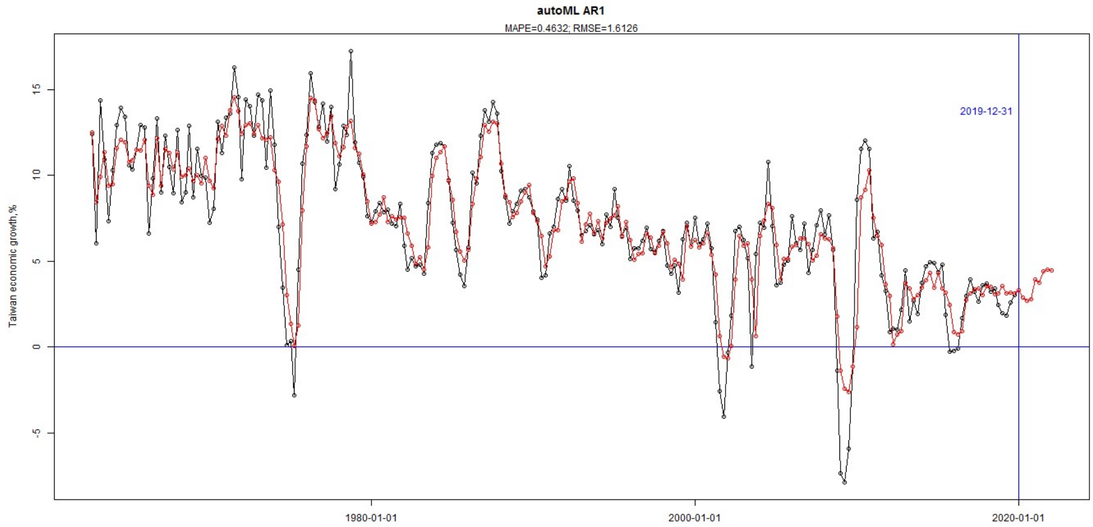
圖 A 2-1.1 以 SVM AR(4)的四個模式之預測 2020Q1-2021Q4

(2) autoML 經濟成長率的 AR(P)預測

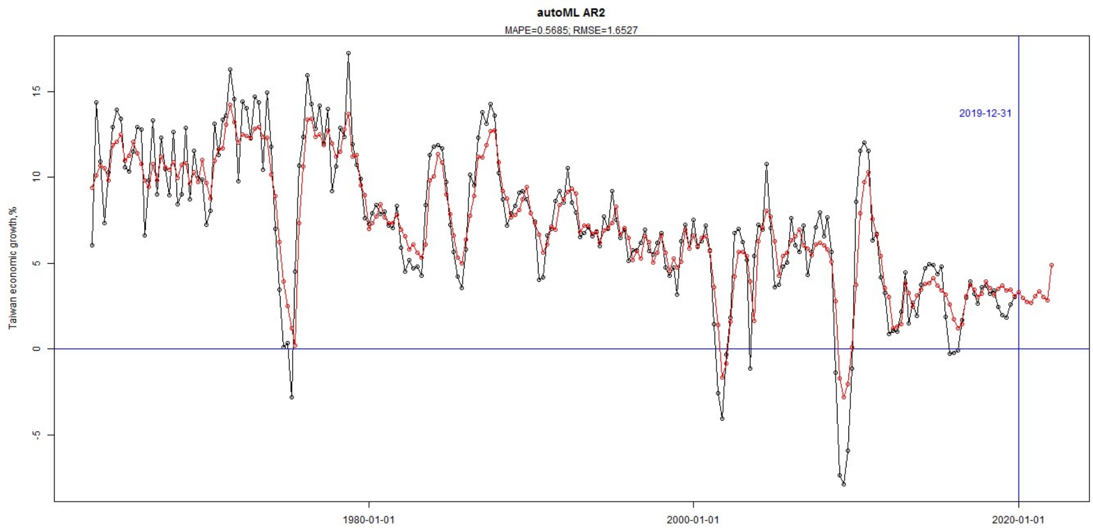
表 A 2-1.5 經濟成長率預測：autoML AR(P)

樣本外預測, %				
	P=1	P=2	P=3	P=4
2020Q1	2.868	2.974	3.230	3.361
2020Q2	2.718	2.765	2.979	3.192
2020Q3	2.792	2.714	3.465	3.157
2020Q4	3.953	3.092	3.514	3.104
2021Q1	3.728	3.379	3.774	3.455
2021Q2	4.408	3.028	4.308	3.026
2021Q3	4.505	2.818	2.692	3.286
2021Q4	4.446	4.915	4.762	4.795
樣本內訓練績效				
RMSE	1.613	1.653	**0.283**	2.536
MAE	1.149	1.230	0.186	1.832
MAPE	0.463	0.568	**0.065**	0.983
ACF1	0.220	0.257	-0.238	0.535
Theil's U	0.0075	0.0100	0.0015	0.0211

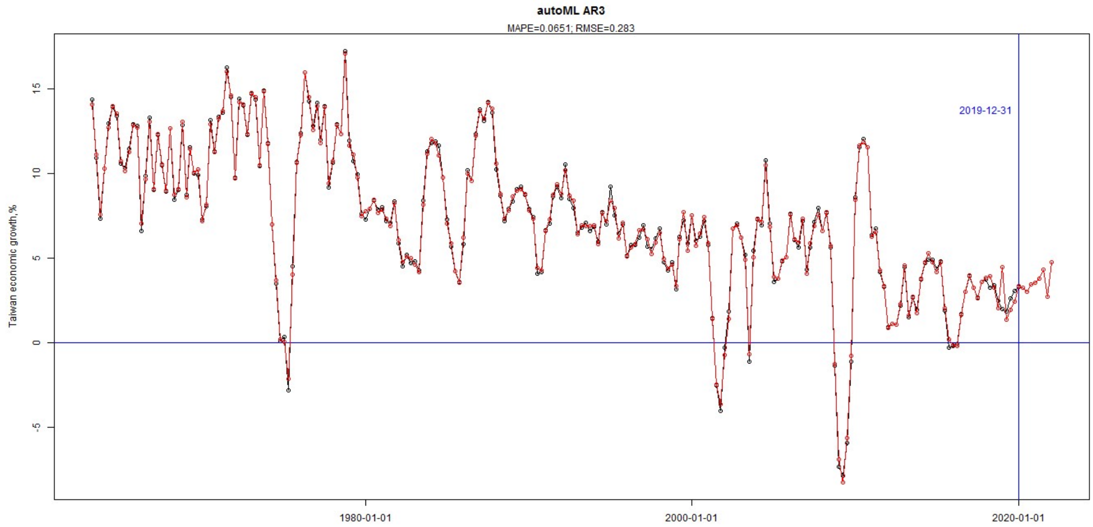
資料來源：本研究自行整理。



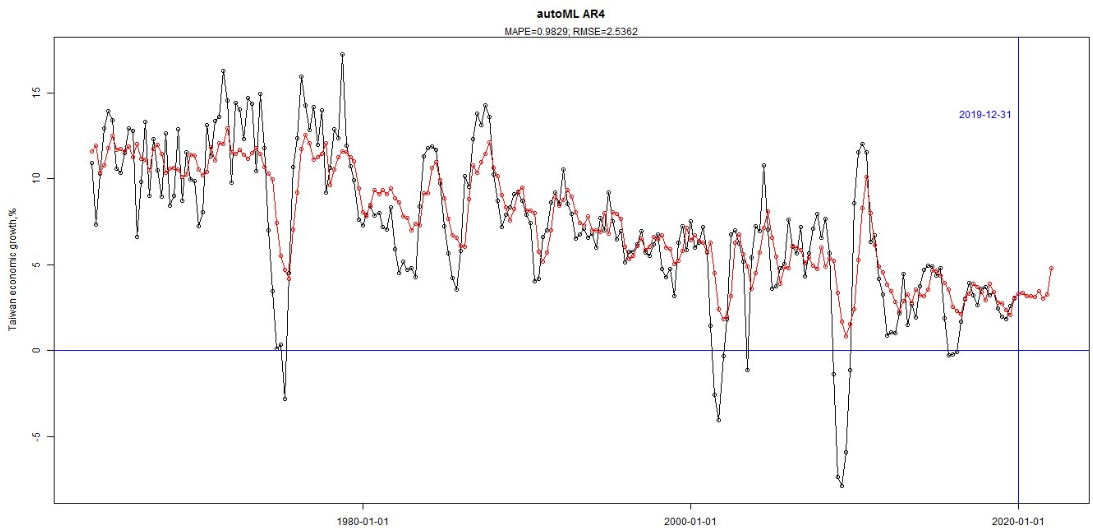
(A) autoML AR(1) Only



(B) autoML AR(2) Only



(C) 精確指標最好的結果：autoML AR(3) Only



(D) autoML AR(4) Only

資料來源：本研究自行整理。

圖 A 2-1.2 以 autoML 模式預測 2020Q1-2021Q4

(3) 傳統時間序列的經濟成長率預測

表 A 2-1.6 列出傳統時間序列的結果，以下模式是基於自動配適最佳 ARIMA 的結果為基礎，再添加其餘虛擬變數。

表 A 2-1.6 統計時間序列樣本內訓練績效, 1962Q1-2019Q4

時間序列模型	RMSE	MAE	MAPE	ACF1	Theil U
ARIMA(2,1,2)(2,0,1)	2.185	1.631	0.494	-0.034	0.012
ARIMA(2,1,2)(2,0,1)+trend	2.138	1.621	0.572	-0.025	0.012
ARIMA(2,1,2)(2,0,1)+seasonDummy	2.178	1.631	0.483	-0.041	0.011
ARIMA(2,1,2)(2,0,1)+trend+seasonDummy	2.075	1.568	0.465	-0.055	0.009
ARIMA(2,1,2)(2,0,1)+Fourier	2.138	1.621	0.572	-0.025	0.012
ARIMA(2,1,2)(2,0,1)+Fourier+trend	2.075	1.568	0.465	-0.055	0.009
bagged.ETS	2.074	1.491	0.581	0.412	0.010
bagged.autoARIMA	1.970	1.452	0.622	0.366	0.013
NNETAR	1.697	1.268	0.421	0.083	0.007
BATS	2.161	1.605	0.499	-0.030	0.011
ARFIMA	2.207	1.635	0.514	-0.031	0.010
LSTAR	2.195	1.630	0.458	0.065	0.009
SETAR	2.198	1.637	0.515	0.059	0.008
GAMs	2.152	1.627	0.554	0.075	0.007

資料來源：本研究自行整理。

表 **A 2-1.7** 統計時間序列經濟成長率樣本外預測值, %

時間序列模型	2020Q1	2020Q2	2020Q3	2020Q4	2021Q1	2021Q2	2021Q3	2021Q4
ARIMA(2,1,2)(2,0,1)	2.26	2.68	2.24	3.04	2.26	2.68	2.24	3.04
ARIMA(2,1,2)(2,0,1) + trend	2.58	2.58	2.58	2.58	2.58	2.58	2.58	2.58
ARIMA(2,1,2)(2,0,1) + seasonDummy	2.60	2.69	2.31	2.81	2.60	2.69	2.31	2.81
ARIMA(2,1,2)(2,0,1) + trend + seasonDummy	2.65	1.98	2.44	2.10	2.65	1.98	2.44	2.10
ARIMA(2,1,2)(2,0,1) + Fourier	2.58	2.58	2.58	2.58	2.58	2.58	2.58	2.58
ARIMA(2,1,2)(2,0,1) + Fourier + trend	2.63	1.98	2.43	2.10	2.63	1.98	2.43	2.10
bagged.ETS	3.09	3.09	3.09	3.09	3.09	3.09	3.09	3.09
bagged.autoARIMA	3.26	3.26	3.15	3.13	3.11	3.02	3.09	3.12
NNETAR	3.24	2.97	3.07	3.08	3.36	3.61	3.43	3.01
BATS	3.66	3.57	3.50	3.34	3.12	3.04	2.96	2.95
ARiMA	3.66	4.53	5.19	5.75	6.29	6.67	6.96	7.18
LSTAR	3.65	4.51	5.18	5.74	6.29	6.68	6.97	7.20
SETAR	3.66	4.53	5.19	5.75	6.29	6.67	6.96	7.18
GAMs	4.06	4.67	5.17	5.65	6.03	6.31	6.39	6.23

資料來源：本研究自行整理。

A2-2. AR(P) + 5 個解釋變數

添加解釋變數 X 的樣本外預測，傳統 ARIMA 模式的表現沒有 5-1 要好，例如，RMSE=1.96, MAPE=0.66。而且添加的解釋變數，讓多數模式無法執行，這小節就略。

(1) SVM 經濟成長率的 AR(P) + 5 個解釋變數的 AR(P)

首先，表 A 2-2.1 指出，單單只有 AR(1)，樣本內訓練的結果是最差的。只要有添加季節虛擬變數或時間趨勢，預測配適就會有大幅度改善。

表 **A 2-2.1** 經濟成長率預測：**SVM P=1**

	AR(1) Only	seasonDummy	trend	seasonDummy+trend
樣本外預測值，%				
2020Q1	4.089	4.191	3.157	3.693
2020Q2	4.659	4.568	2.498	3.326
2020Q3	6.164	5.457	3.603	2.923
2020Q4	4.453	3.792	3.212	1.131
2021Q1	3.767	3.191	2.654	1.425
2021Q2	4.804	3.370	2.987	2.585
2021Q3	4.821	3.270	2.215	1.905
2021Q4	6.052	6.173	3.230	4.385
樣本內訓練績效				
RMSE	1.779	1.648	1.722	1.602
MAE	1.369	1.187	1.383	1.274
MAPE	0.532	0.451	0.527	0.500
ACF1	0.069	0.023	0.136	0.174
Theil's U	0.0056	0.0037	0.0075	0.0066

資料來源：本研究自行整理。

表 A 2-2.2 經濟成長率預測：SVM P=2

	AR(2) Only	seasonDummy	trend	seasonDummy+trend
樣本外預測值，%				
2020Q1	3.748	3.994	2.801	2.838
2020Q2	5.346	5.359	3.338	3.425
2020Q3	5.195	4.293	3.287	3.158
2020Q4	3.114	1.780	1.903	0.476
2021Q1	3.930	4.551	2.273	2.222
2021Q2	5.399	5.110	3.547	2.725
2021Q3	5.131	4.479	3.048	2.576
2021Q4	8.605	8.843	4.779	5.184
樣本內訓練績效				
RMSE	1.653	1.708	1.496	1.400
MAE	1.373	1.460	1.190	1.132
MAPE	0.470	0.527	0.407	0.391
ACF1	0.058	0.104	0.078	0.096
Theil's U	0.0057	0.0071	0.0046	0.0058

資料來源：本研究自行整理。

表 A 2-2.3 經濟成長率預測：SVM P=3

	AR(3) Only	seasonDummy	trend	seasonDummy+trend
樣本外預測值，%				
2020Q1	3.521	3.773	3.383	3.230
2020Q2	4.215	4.638	3.399	3.491
2020Q3	3.761	5.298	2.937	3.935
2020Q4	2.935	2.001	1.826	0.480
2021Q1	3.434	4.847	2.289	2.392
2021Q2	4.733	7.048	3.218	4.580
2021Q3	5.282	6.678	3.473	4.347
2021Q4	6.912	9.703	4.714	6.105
樣本內訓練績效				
RMSE	1.166	1.457	1.105	1.158
MAE	0.537	1.244	0.481	0.935
MAPE	0.169	0.404	**0.153**	0.321
ACF1	0.185	0.201	0.127	0.131
Theil's U	0.0024	0.0093	0.0022	0.0062

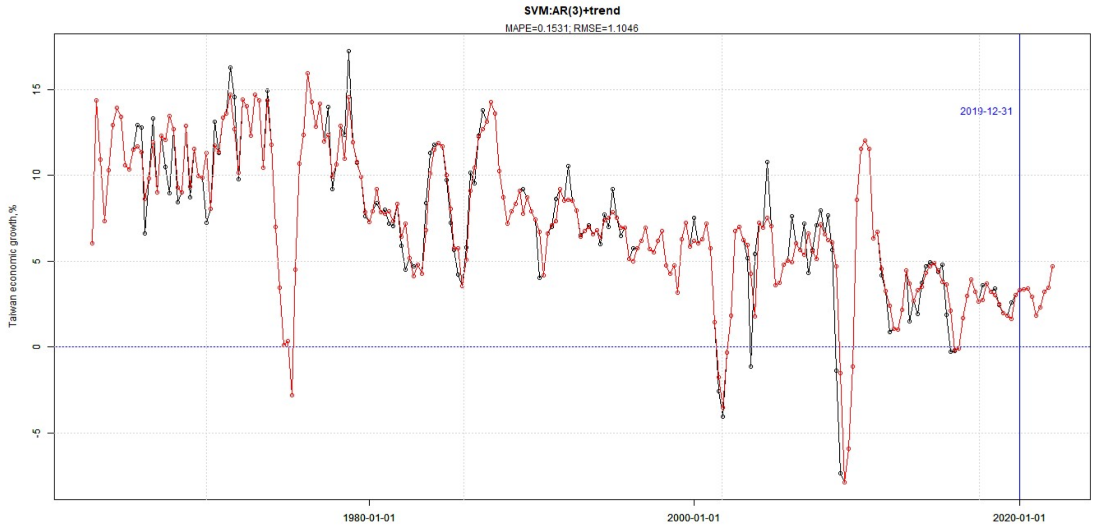
資料來源：本研究自行整理。

表 A 2-2.4 經濟成長率預測：SVM P=4

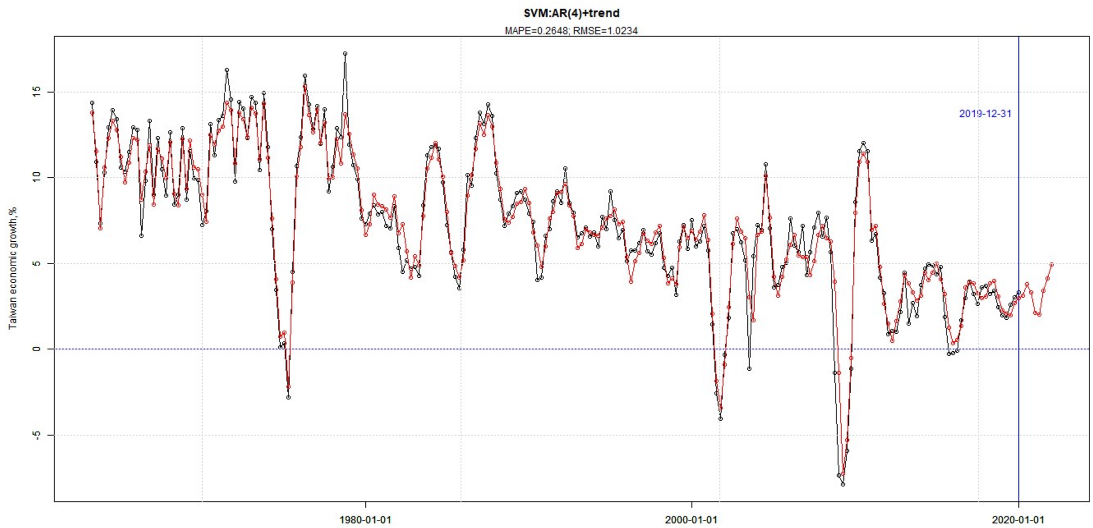
	AR(4) Only	seasonDummy	trend	seasonDummy+trend
樣本外預測值，%				
2020Q1	4.272	3.757	3.141	3.200
2020Q2	5.289	4.564	3.796	3.403
2020Q3	5.098	5.360	3.304	3.809
2020Q4	4.581	2.757	2.108	0.674
2021Q1	4.843	4.312	2.035	1.836
2021Q2	6.724	6.752	3.415	3.741
2021Q3	8.017	7.366	4.126	4.129
2021Q4	9.264	10.222	4.950	6.498
樣本內訓練績效				
RMSE	1.069	1.315	**1.023**	1.138
MAE	0.784	1.143	0.746	0.968
MAPE	0.275	0.403	0.265	0.337
ACF1	0.172	0.238	0.212	0.221
Theil's U	0.004	0.0086	0.0039	0.0062

資料來源：本研究自行整理。

整體看起來，AR(3)和 AR(4)的狀況類似較好，我們將 AR(3)和 AR(4)最小 MAPE/RMSE 的預測結果繪於圖 A 2-2.1 (A)與(B)兩張圖。



(A) P=3，最小 MAPE



(B) P=4，最小 RMSE

資料來源：本研究自行整理。

圖 A 2-2.1 以 SVM 模式之預測 2020Q1-2021Q4

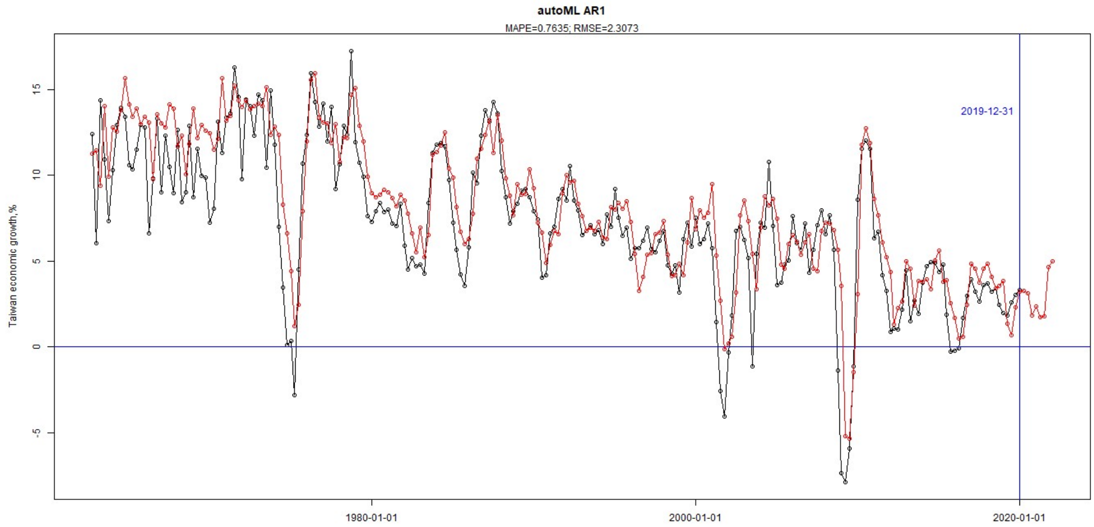
(2) autoML AR(P) + 5 個解釋變數 AR(P)的經濟成長率預測

4 個落後期皆預測 2021 後兩季的復甦；然而，訓練表現最好 P=2 模型，在過程中會有數季負成長，然後反彈。

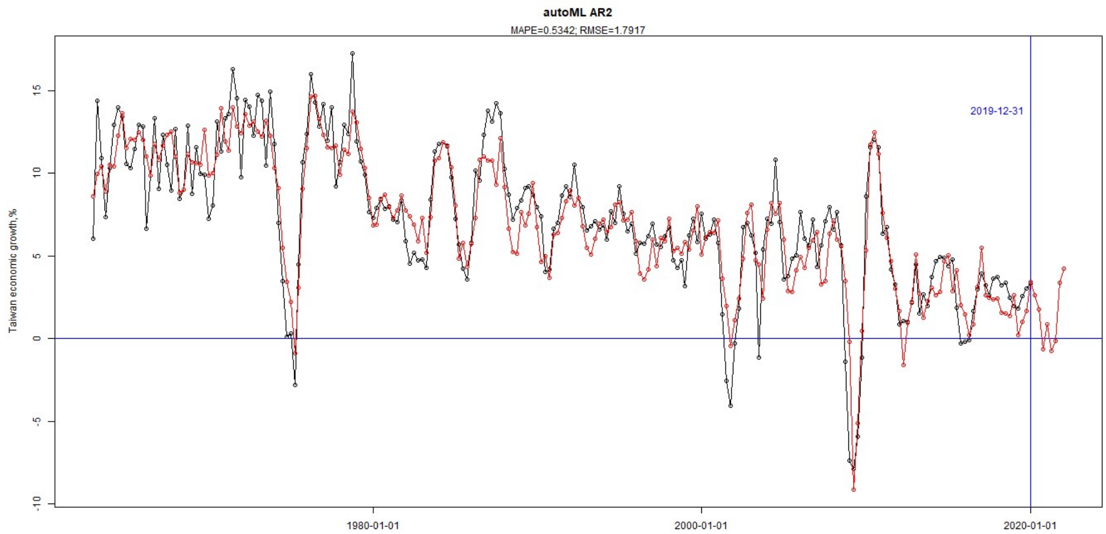
表 A 2-2.5 經濟成長率預測：autoML

樣本外預測, %				
	P=1	P=2	P=3	P=4
2020Q1	3.291	2.604	4.022	3.625
2020Q2	3.142	1.780	3.171	2.631
2020Q3	1.828	-0.635	1.451	0.520
2020Q4	2.366	0.885	2.092	0.935
2021Q1	1.724	-0.714	3.868	0.284
2021Q2	1.808	-0.158	2.872	0.904
2021Q3	4.633	3.363	4.191	2.864
2021Q4	4.975	4.216	5.365	4.737
樣本內訓練績效				
RMSE	2.307	**1.792**	2.362	2.022
MAE	1.680	1.417	1.815	1.527
MAPE	0.764	**0.534**	0.646	0.612
ACF1	0.206	0.169	0.463	0.387
Theil's U	0.0162	0.0087	0.0094	0.0060

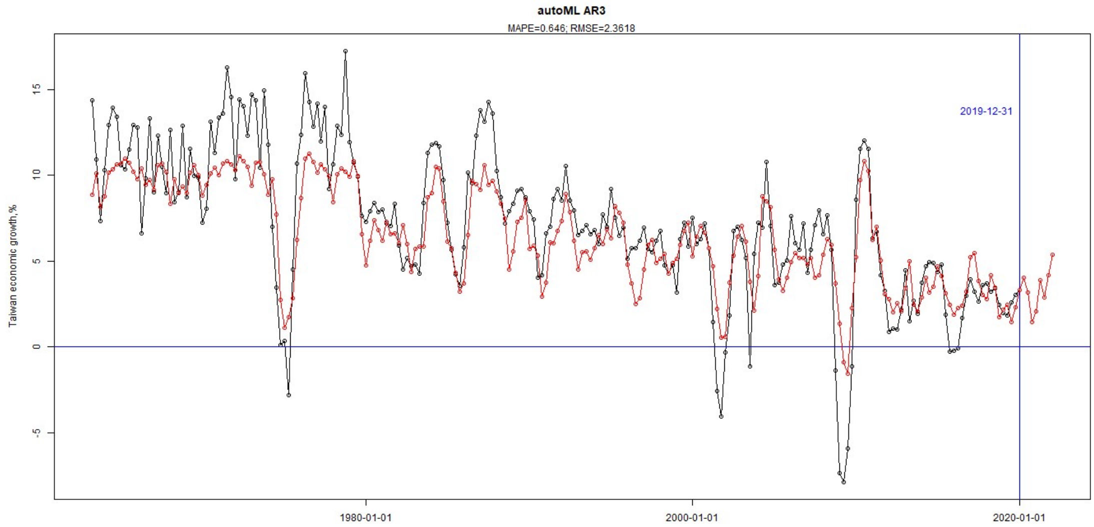
資料來源：本研究自行整理。



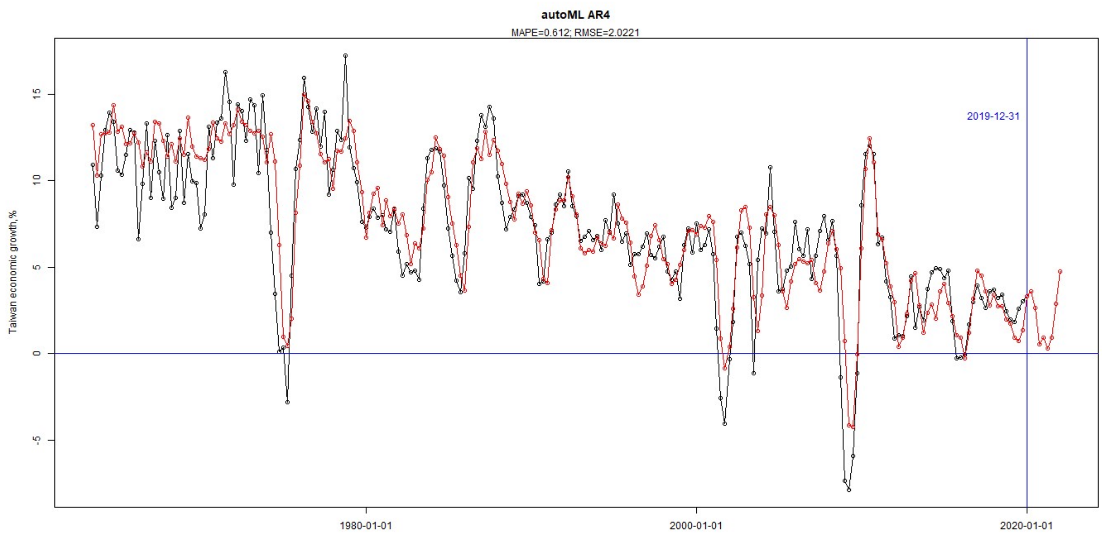
(A) P=1



(B) P=2，指標 MAPE/RMSE 最小



(C) P=3



(D) P=4

資料來源：本研究自行整理。

圖 A 2-2.2 autoML

(3) 傳統時間序列的經濟成長率預測

表 A 2-2.6 列出傳統時間序列的 16 種設定的樣本內訓練精確度結果。表中的 p 是其他變數 X 的 AR(P)。被預測變數 Y 的結構由自動配適最佳 ARIMA 的結果決定。添加其餘虛擬變數有確定趨勢(trend)和季節虛擬變數(seasonD)。由 MAPE 和 RMSE 看來，傳統時間序列和 autoML 類似，但不比 SVM 好。

表 **A 2-2.6** 統計時間序列樣本內訓練績效**, 1962Q1-2019Q4**

時間序列模型	**RMSE**	**MAE**	**MAPE**	**ACF1**	**Theil U**
ARIMA(2,1,2)(2,0,1), p=1	2.054	1.581	0.559	-0.0786	0.0091
ARIMA(2,1,2)(1,0,1), p=2	2.005	1.565	0.521	-0.0597	0.0091
ARIMA(3,1,0)(2,0,1), p=3	2.091	1.614	0.466	0.0313	0.0082
ARIMA(3,1,2)(1,0,1), p=4	1.846	1.423	0.435	0.0088	0.0055
ARIMA(2,1,0)(1,0,1).seasonD, p=1	2.211	1.713	0.596	0.0248	0.0097
ARIMA(0,1,2)(1,0,1).seasonD, p=2	2.081	1.620	0.588	-0.0102	0.0075
ARIMA(2,1,0)(1,0,1).seasonD, p=3	2.092	1.612	0.476	0.0226	0.0077
ARIMA(1,1,2)(0,0,1).seasonD, p=4	1.899	1.469	0.494	0.0425	0.0074
ARIMA(0,0,3)(1,0,0).trend, p=1	1.926	1.460	0.452	-0.0598	0.0067
ARIMA(1,0,0)(1,0,1).trend, p=2	2.029	1.556	0.559	-0.0239	0.0084
ARIMA(1,0,0)(2,0,1).trend, p=3	1.972	1.525	0.480	-0.0025	0.0063
ARIMA(1,0,0)(1,0,0).trend, p=4	1.893	1.443	0.470	0.0052	0.0068
ARIMA(1,0,0)(1,0,1).seasonD.trend, p=1	2.059	1.561	0.604	-0.0316	0.0077
ARIMA(3,0,0)(0,0,1).seasonD.trend, p=2	2.016	1.550	0.561	0.0178	0.0069
ARIMA(1,0,0)(1,0,1).seasonD.trend, p=3	1.967	1.520	0.491	-0.0029	0.0063
ARIMA(1,0,0)(1,0,0).seasonD.trend ,p=4	1.887	1.439	0.484	0.0048	0.0069

資料來源：本研究自行整理。

表 A 2-2.7 則是各模型產生的八季預測。

表 **A 2-2.7** 統計時間序列經濟成長率樣本外預測值, %

時間序列模型	2020Q1	2020Q2	2020Q3	2020Q4	2021Q1	2021Q2	2021Q3	2021Q4
ARIMA(2,1,2)(2,0,1), p=1	4.618	4.004	0.949	1.699	2.884	2.435	5.489	3.814
ARIMA(2,1,2)(1,0,1), p=2	4.441	3.915	0.862	1.070	2.190	2.120	5.296	4.053
ARIMA(3,1,0)(2,0,1), p=3	4.743	4.376	0.804	1.754	2.954	2.631	5.493	4.240
ARIMA(3,1,2)(1,0,1), p=4	4.977	4.097	1.303	1.671	2.368	2.788	5.353	3.991
ARIMA(2,1,0)(1,0,1).seasonD, p=1	4.886	4.926	1.566	2.228	3.557	3.359	6.570	4.520
ARIMA(0,1,2)(1,0,1).seasonD, p=2	4.191	4.071	0.737	0.864	2.022	2.228	5.572	3.942
ARIMA(2,1,0)(1,0,1).seasonD, p=3	4.698	4.604	1.165	1.585	2.905	2.814	5.717	4.288
ARIMA(1,1,2)(0,0,1).seasonD, p=4	5.151	4.000	1.213	0.846	1.740	2.949	5.762	4.385
ARIMA(0,0,3)(1,0,0).trend, p=1	11.679	10.640	8.102	8.655	9.422	9.114	11.506	10.144
ARIMA(1,0,0)(1,0,1).trend, p=2	10.688	10.483	7.232	7.750	8.497	8.423	11.326	10.040
ARIMA(1,0,0)(2,0,1).trend, p=3	11.094	10.920	7.662	8.242	9.015	8.841	11.425	10.322
ARIMA(1,0,0)(1,0,0).trend, p=4	12.137	11.629	8.755	8.926	9.267	10.143	12.334	11.053
ARIMA(1,0,0)(1,0,1).seasonD.trend, p=1	10.920	10.700	7.686	8.155	9.104	8.882	11.634	9.802
ARIMA(3,0,0)(0,0,1).seasonD.trend, p=2	10.863	10.716	7.328	7.683	8.585	8.509	11.756	9.803
ARIMA(1,0,0)(1,0,1).seasonD.trend, p=3	11.127	11.024	7.863	8.115	8.974	8.911	11.591	10.189
ARIMA(1,0,0)(1,0,0).seasonD.trend ,p=4	12.120	11.711	8.986	8.829	9.155	10.211	12.524	10.950

資料來源：本研究自行整理。

總結本附錄結果，當未添加 X 時，autoML 的樣本內訓練績效比 SVM 要好；當添加 X 時，SVM 的樣本內訓練績效比 autoML 要好。autoML 在未添加 X 時的樣本內配適度比添加 X 時更好，但可能存在過度配適的問題。整體訓練結果，SVM 應該是最好的。

若以 SVM 最好的結果(SVM AR(3)+trend+5 個解釋變數)為主，對未來的預測應如圖 A 2-2.1 (或表 A 2-2.3)所示，2020 年第 3、4 季經濟成長率降至 2.9%與 1.8%，2021 年則逐季走升，各季約介於 2.3%-4.7%。

最後，機器學習的特性是由資料驅動(data-driven)，花時間討論應該放哪些解釋變數有意義，哪些不該放，沒有太大的意義。對時間序列模型，一筆不適當的資料可能就是全域(globally)沒有用。但是，對於機器學習，局部(locally)有用的資料，就可以參考，機器學習只是盡可能發現任何資料有用於預測的部份，再將之拼湊起來。機器學習的實務態度，應該是去嘗試各種數據，然後由資料驅動去預測與解釋。

如同外部審查委員徐教授指出，用傳統時間序列模型的態度看機器學習，硬把機器學習塞進時間序列的框架，其實是需要改變的研究心態。

附錄三 獨立成分分析

本附錄說明研究內文 6.3 節中為何捨去獨立成分方法而採逐次清算。根據合約「我們盡可能透過蒐集大量數據來訓練與學習，以提高分類預測精確性。會視數據量，採用獨立成分分析(Independent Component Analysis, ICA)方法，萃取少量因子做為預測變數」。計畫書的關鍵是「會視數據量，採用 ICA」。研究中最後沒有採用 ICA 降維的原因在於，ICA 萃取的成分因子，在降維程度並不明顯。我們使用 3 筆外國的變數與 9 筆國內的經濟數據以萃取因子。3 筆外國變數分別為美國、OECD 和七大工業國(G7)的 GDP 成長率(yoy)。台灣的經濟數據包括: 1.實質民間消費年增率、 2.實質政府消費年增率、 3.固定資本形成年增率、4.商品與服務輸出年增率、5.商品與服務輸入年增率、6.貿易條件利得年增率、7.實質國外需求年增率、8.淨國外要素所得年增率、9.工業生產指數年增率。

我們做了兩種成分法：主成分分析(Principal Component Analysis, PCA)和獨立成分分析(ICA)；分別說明如下。

A1. PCA (Hastie et al. 2009, pp.534-552)

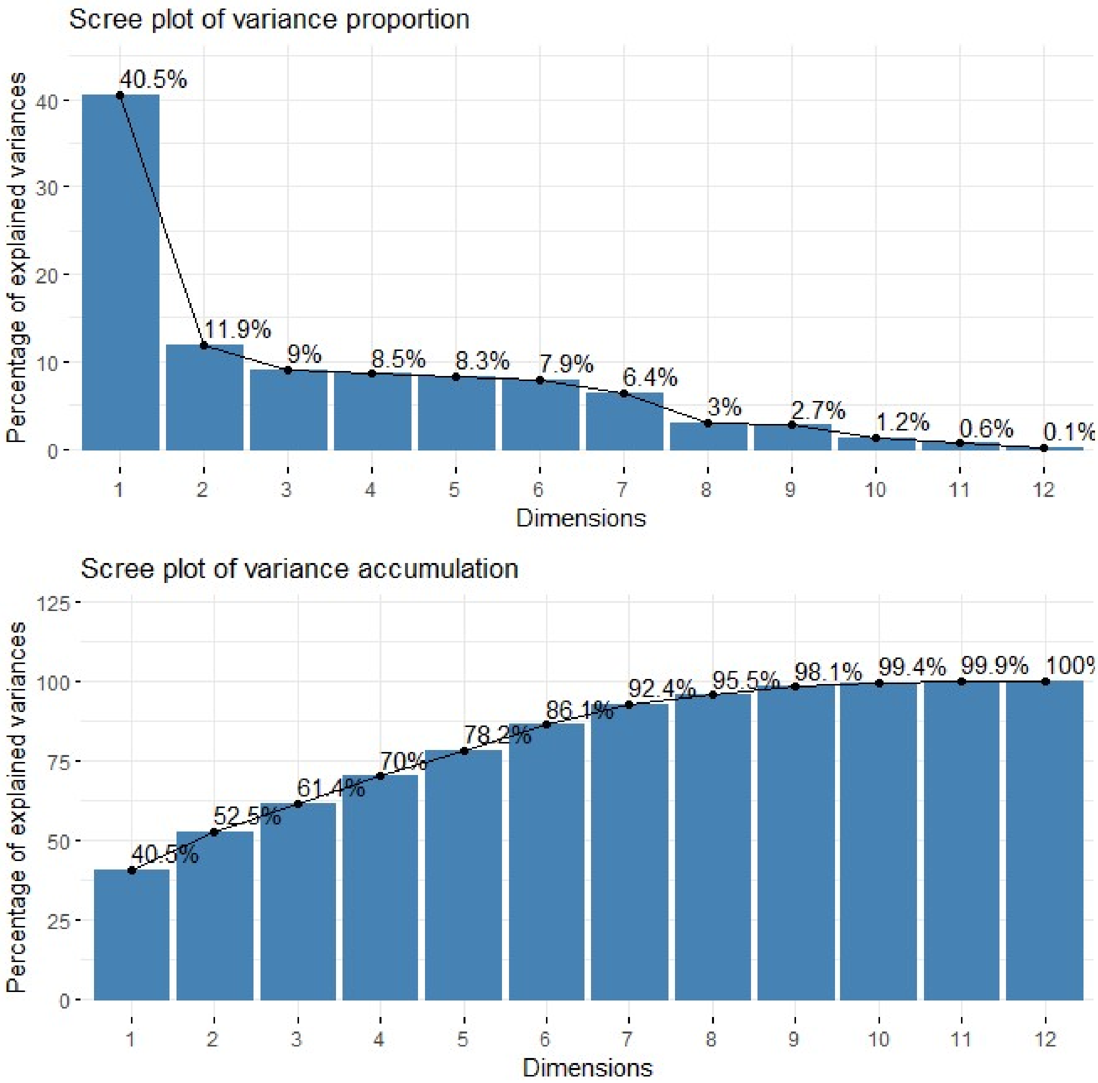
因為經濟成長率是變動率，所以我們先對 12 個解釋變數依照當季對應上年同季計算變動率，然後輸入這 12 筆資料矩陣。主成分就是 12 個變數的線性組合，主成分方法從 12 筆資料最多產生 12 個互相獨立的主成分變數。我們取的成分個數依照對變異數解釋比率為主：例如，取前三個主成分，這三個主成分解釋了 12 筆資料 80%的變異，這其實就是變異數分析(ANOVA)的原理。資料繪製的陡坡圖(scree plot)如下。如圖 A3.1 上方是 12 個主成分的個別變異數貢獻，下方則是累積百分比。我們一般用累積百分比選擇少量主成分。如果定在 90%，則選擇 6～7 個主成分。這樣的數量並沒有達到顯著的降維效果。對此文所需要的演算，我們希望主成分數量是不超過 3 個。原因如下：

$$\sum_{1}^{3\times4} C_i^{3\times4} = 4,095$$

$$\sum_{1}^{4\times4} C_i^{4\times4} = 65,535$$

$$\sum_{1}^{5\times4} C_i^{5\times4} = 1,048,575$$

依 6.3 節的資料維度展開，選 3 個外國變數的 4,095 組，已經是非常龐大的數據結構，添一個，就會暴增，如果取 5 個就破百萬，CPU 演算成本非常高，如果用 GPU 或有顯著改善。因此，這個 PCA 降維措施並未產生太好的結果。



資料來源：本研究自行整理。

圖 A3.1 PCA 因子陡坡圖(scree plot)

A2. ICA (Hastie et al. 2009, pp.557-570)

跟主成份分析需要決定選多少個主成份(利用「解釋變異」)的概念一樣，一般在使用 ICA 時，也必須知道原始訊號的數量。ICA 和 PCA 的差異在於「還原」。ICA 將 12 筆資料來自 12 筆被雜訊包裝的資料，因此 ICA 的演算重點在於去除雜訊。

在實務應用中，一般可以仰賴領域(先驗)知識來決定。例如，可以確定原始訊號的個數，跟混合訊號一樣。但如果沒有相關經驗或知識的話，就很難判斷 ICA 中的來源訊號要選幾個。這時候，就得依照某些方法選擇 component 的數量。其中，基於 information-theoretic criteria (ITC)的資訊理論，是一個不錯且實用的方法。ICA 不似 PCA 可以排序，所以我們將萃取訊號畫為圖 A 3.2，和 PCA 一樣，ICA 因子彼此獨立。在 12 個 ICA 中，依據和經濟成長型態類因子選取，排除 ICA2 和 ICA12，剩下 10 個中，依據型態差異，至少超過 5 個可以採用。

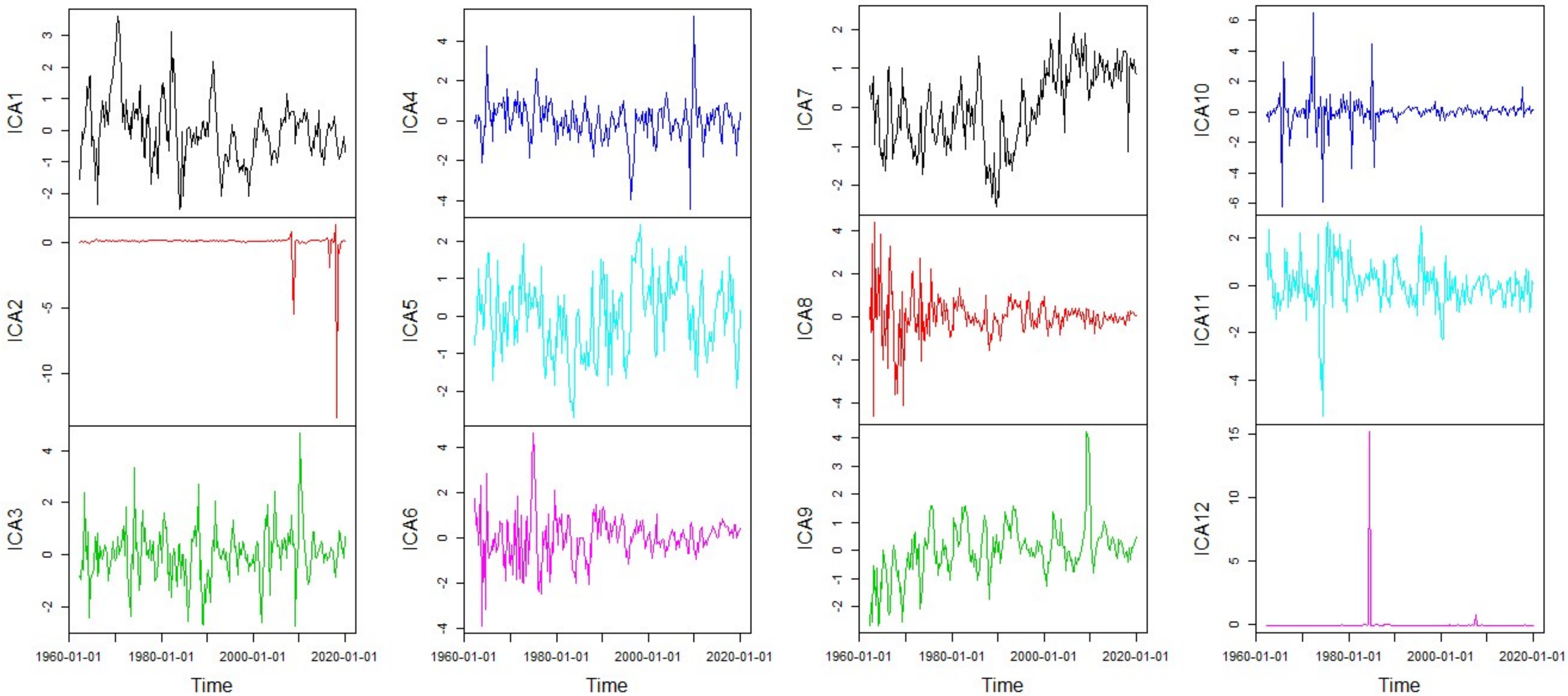
本案選和經濟成長率相關係數絕對值大於 0.25 的 ICA，然後嘗試 Case 2 的運算，表現並不比前面原 Case 2 理想(圖二十八)。下表 A 3.1 和圖 A 3.2 是這方法的部份結果。表 A 3.1 的預測結果繪在圖 A 3.3。

表 A 3.1 SVM ICA 的預測精確度指標

	RMSE	MAE	MAPE	ACF1	Theil's U
SVM h=1	1.402	1.180	0.404	0.295	0.0219
SVM h=2	1.799	1.394	0.499	0.215	0.0273
SVM h=3	2.671	2.174	0.823	0.0065	0.0463
SVM h=4	2.803	2.405	0.933	0.265	0.0582

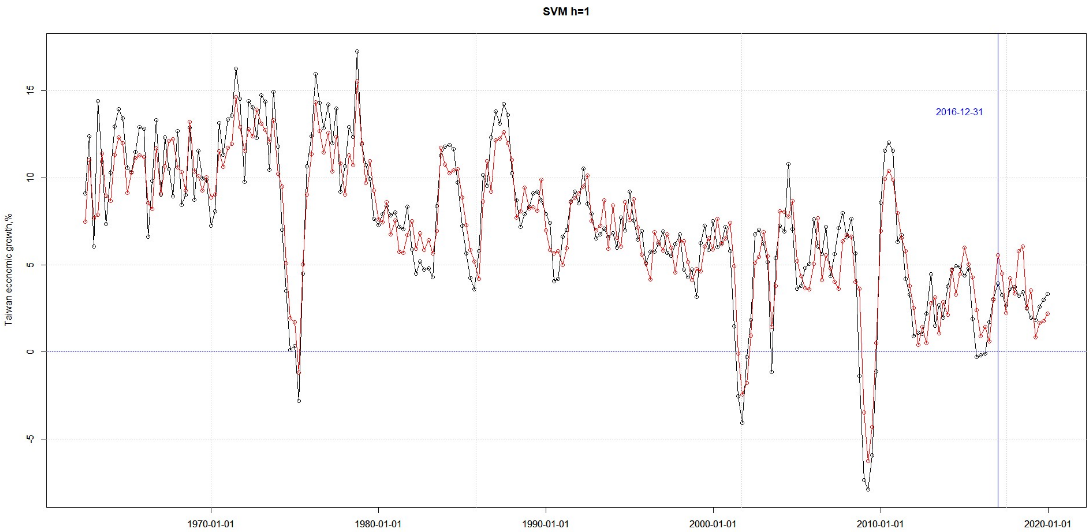
資料來源：本研究自行整理。

最後，ICA 或 PCA 因為是組合成分，失去經濟意義直觀，而且對未來預測也十分困難，ICA 或 PCA 依賴歷史資料，所以未來的參考值必需先預測 X。綜合以上的分析，我們就沒有將 ICA/PCA 的結果正式納入報告章節。

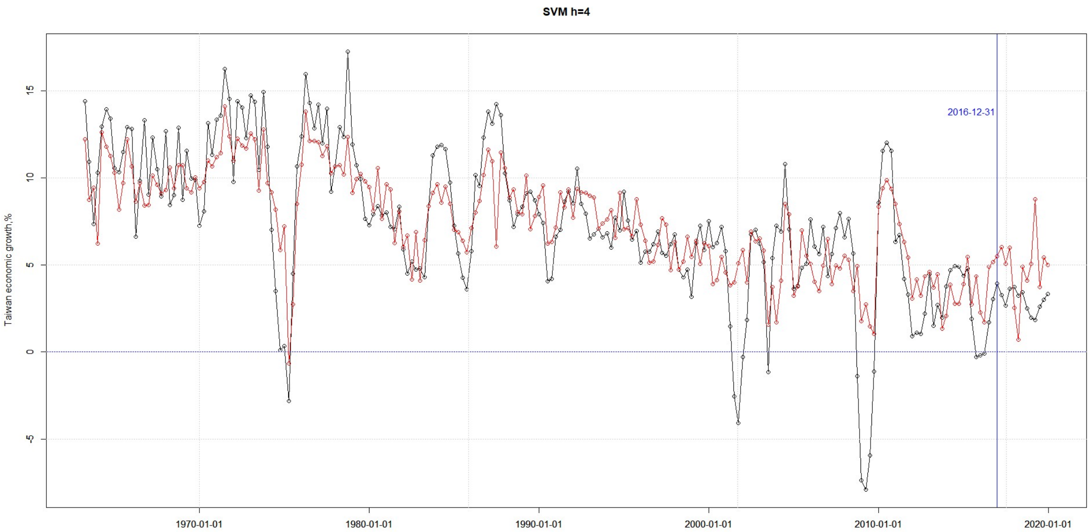


資料來源：本研究自行整理。

圖 A 3.2 由 12 筆資料萃取之 ICA 因子



(A) h=1



(B) h=4

資料來源：本研究自行整理。

圖 A 3.3 利用 ICA 的 SVM 預測圖

附件一 R 程式在資料串流的說明

本計畫隨結案報告移交一個 R 的壓縮檔 R.rar，解開後會有四個資料夾：

DEMO

R_0_DownloadOpenData

R_1st

R_2nd

DEMO 內為技術移轉的程式，裡面的程式對應技術移轉文件說明。

R_1st 和 R_2nd 分別為期中和期末之後的兩次實證分析程式。包含預測 2020-2021 的內容。

另外，根據合約所述，本案將提供下載全球開放數據的串接方法作為技術轉移。這是透過 R 程式 (資料夾 **R_0_DownloadOpenData** 內的程式檔 **downloadData.R**) 執行資料的網際網路下載。這個途徑是由 Open Data 網站 (https://stock-ai.com/)透過購買一年的 R Key (如下 key="IRxZmPatcN")。

downloadData.R 程式碼說明

只要輸入資料代碼（如下 quarterlyIndex 或 monthlyIndex 內的字碼，如 twRGDPNTDYoY 為臺灣依支出面計算的經濟成長率），此研究案使用的所有資料，皆可以輕易更新，全球財經數據的代碼，皆可以由(https://stock-ai.com/)以 Open Data 的方式取得。因為擷取多筆資料時，資料頻率必須一樣以利疊取資料。以下程式範例即是分兩個頻率的數據，如果需要擷取日資料，同理類推。當所需數據擷取完畢後，可將之存在 PC 可作為交換格式，一種通用交換格式為 csv，如下：

```
write.csv(QrtData, file="D:/myfile.csv")
```

```
library(timeSeries)
## This code creates real economic growth as GDP by enpenditure
key="lRxZmPatcN"
path="https://stock-ai.com/history-data-download?symbol="
quarterlyIndex=c("twRGDPNTDYoY","G7NGDPYOY","OECDNGDPYOY","USNGDPYOY")
# variable name of Stock-AI
#twRGDPNTDYoY= Economic growth of Taiwan by expenditure (2016=base year)
#G7NGDPYOY= Economic growth of G-7
#OECDNGDPYOY= Economic growth of OECD
#USNGDPYOY=Economic growth of USA

    QrtData=NULL
    for(i in 1:length(quarterlyIndex)) {
    url=paste0(path,quarterlyIndex[i], "&export=csv&startDate=1962-01-
    01&endDate=2019-10-01&key=", key,"&export=.csv")
    filname=paste0(quarterlyIndex[i],".csv")
    temp <- tempfile(tmp=getwd(),fileext = ".csv")
    download.file(url,destfile=filname)
    temp0 <- read.csv(filname)
    timeID=temp0[,1]
    tmp=as.matrix(temp0[,-1])
    rownames(tmp)=timeID
    QrtData=cbind(QrtData,tmp)
    }
    colnames(QrtData)=c("Taiwan","G7","OECD","USA")

QrtData=as.timeSeries(as.ts(QrtData))

write.csv(QrtData, file="D:/myfile.csv")

#=========================================
monthlyIndex=c("TTOECDCLI","G7OECDCLI","NAFTAOECDCLI","CBCLI","CBCCNI","CB
CLGI","FBUOECDCLI")

#TTOECDCLI = Monthly leading indicator of OECD (All)
#G7OECDCLI = Monthly leading indicator of OECD (G7)
```

```
#NAFTAOECDCLI = Monthly leading indicator of OECD (NAFTA)
#CBCLI = Monthly Leading indicator (US CBC, SA)
#CBCCNI = Monthly Coincidence indicator (US CBC, SA)
#CBCLGI = Monthly Lag indicator (US CBC, SA)
#FBUOECDCLI = Monthly leading indicator of OECD (Europe Big 4)
#TWC01 = Monthly unemployment rate (Taiwan)

  bcIndex=NULL
  for(j in 1:length(monthlyIndex)) {

    url=paste0(path, monthlyIndex[j], "&export=csv&startDate=1961-01-
  01&endDate=2019-12-01&key=", key,"&export=.csv")
    filname=paste0(monthlyIndex[j],".csv")
    temp <- tempfile(tmp=getwd(),fileext = ".csv")
    download.file(url,destfile=filname)
    temp0 <- read.csv(filname)
    timeID=as.character(temp0[,1])
    tmp=as.matrix(temp0[,-1])
    rownames(tmp)=timeID
    tmp=as.timeSeries(tmp)
    bcIndex=cbind(bcIndex,tmp)

  }

  colnames(bcIndex)=c("OECD_all","G7","NAFTA","US_LD","US_coin","US_lag","E
  urope_Big4")
```

附件二 期中報告意見回覆

徐士勛老師 期中報告意見回覆書

1. 文獻回顧方面，2.2 節關於機器學習的應用文獻多是針對相對高頻的貨幣或是金融變數，是否已經有一些使用機器學習預測經濟成長率的相關文獻可供參考？

回覆：謝謝徐老師之意見，Feng 和 Zhang (2014)使用機器學習方法為中國浙江省第三產業對該省 GDP 之貢獻率進行預測，該方法之預測表現比傳統時間序列模型更好，而 Gogas et al. (2015)在預測美國實質 GDP 衰退也肯定機器學習方法之表現，本案在文獻回顧加入上述論文(請見第 7-8 頁)，加強本研究使用機器學習方法預測臺灣經濟成長率之動機。

2. 在實證分析中，研究團隊使用了 1962Q1 - 2016Q3 共 219 筆歷史季資料進行模型估計（訓練），然後針對 2016Q4 – 2019Q2 共 11 筆資料進行樣本外的預測評估。這樣的樣本內外筆數的考量為何？是否比例太過懸殊？在近 230 筆歷史資料下，保留後 1/4 至 1/3 期間進行樣本外預測績效評估應仍可行，這樣或許可進一步驗證各模型面對可能經歷些結構性改變下的樣本外的預測能力。

回覆：因為 230 季的觀察值，用(7:3)分割對傳統時間序列模型是足夠，但是，對目前冠上學習演算法之名的 AutoML 或 LSTM，整個 230 都嫌少，因為時間序列的分段，會訓練不足。後來採用預測樣本外 12 季，也是遷就之下的選擇。Chakraborty and Joseph (2017) 的英格蘭銀行 WP，樣本外用 8 季，以 60

季開始，用逐次增 1 的方法，recursively update。用季資料做機器學習建模預測，受限於觀察值太少。我們選用的 AutoML 和 LSTM 都會因為樣本過小出現訓練不足的 Error 警訊，目前的分割雖然 LSTM 沒有錯誤警訊，但是 autoML 計算還是很勉強。

3. 報告中的多步預測設計似乎與相關文獻常用的遞迴（recursive）法針對未來 h 期的預測 (h-step ahead forecasts) 績效評估不同。 這部分是否調整再請研究團隊斟酌。

回覆：本案的多步和通用作法一樣。原先的多步預測，是先估計 AR(1)再用 recursive 方法動態計算多步樣本外預測，和常用作法唯一不同在於我們受限於樣本數，無法執行所謂的 evaluation on a rolling forecasting origin。

在研究方法部份，已經把多步預測的兩個方法(recursive & direct)做了完整說明。本案後半部將會以直接多步 direct multistep 為主。

4. 表二與表三的結果呈現，建議可增加時間序列模型與機器學習模型兩大類的標示，以方便讀者辨識比較。

回覆：謝謝徐老師之意見。

5. 參考文獻缺漏或錯字：(p.11) Alysha et al. (2011)、 (p.14) Vapnik(2000)、 (p.41) 陳宜廷，徐「士」勛，劉瑞文，莊額嘉 (2011)。

回覆：以上缺漏之參考文獻以及錯字皆做好修正。

陳宜廷老師 期中報告意見回覆書

1. Rationale of using these methods in this empirical context?

 回覆：機器學習方法選擇對沒有預設，原則上就是訓練時，模型越多越好，資料越多越好。所以目前 Google 開發的 autoML 是很通用的方法。此案，採用 SVM 是它在各領域的表現優良且計算相對簡易；LSTM 則是在深度學習中有很好的表現。嚴格上講，6 個模式的自動化機器學習(auto.ML)才具有所謂的機器學習的實務性質。

2. Implementation of these methods? Should we need validation after training & before prediction?

 回覆：基於樣本偏少，方法上沒有另外設計 validation 期。autoML 原先有 2 年共 8 季 (2015 與 2016) validation，但是，警示連連因此取消。

3. Sampling variations of RMSE/MAE/MAPE…. (Simulation? Other targets?)

 回覆：For large sample size, we can check the variations of RMSE/MAPE under different sample sizes. Nor do we use rolling method, hence, we do not examine the performance changes of accuracy indices. I have to say, sampling variation is widely used in the standard k-fold CV. However, in our case, short time series is hard to do this. We merely calculate sample prediction error by actual–prediction, and then compute accuracy indices. No simulation, no rolling.

4. Chen et al. (2011, TER): forecasting updating.

回覆：內文修改新增：「陳宜廷等 (2011) 在模型設置時納入了消費、貨幣和國外部門的歷史訊息，發現可以改善主計處發布的一季前臺灣經濟成長率之預測能力，惟仍未能解決景氣循環時對訊息過度反應而產生的預測誤差。」

5. Chen (2019, TEFP): synthetic control is for generating counterfactual not for forecasting.

回覆：已做移除。

期中報告「經濟研究處」彙整意見回覆書

一、　文章內容相關意見

(一)樣本期間設定及模型預測方法

1. 計畫雖說明樣本外需包含轉折點，惟作者將 1962Q1~2016Q3 當作樣本內，2016Q4-2019Q2 為樣本外預測期間，樣本外僅有 11 個樣本點，且轉折點並不明顯，請補充說明為何採用這樣的樣本切點。理論上，非線性模型對大幅偏離常態的時期會有較佳的預測能力，因此，若樣本自 1962Q1 起，是否可以嘗試樣本外預測包含全球金融危機期間(有較大的轉折點)或許較能比較出不同模型的預測能力？

回覆：

(1) 樣本內和樣本外的分法原因是資料個數。因為我們的研究重點在比較機器學習預測，資料量不足機器學習的演算特性難以顯現，尤其是自動化機器學習方法(autoML)。目前的 230 筆季資料對 SVM 是可以，但是對其他方法就不好。為勉強綜合比較三個機器學習方法的預測表現，我們擷取如此子樣本。

(2) 原先使用是依照「生產面」計算的經濟成長率，在上述時間本有一個往下的轉折。後經央行指示改採用「依支出面」計算的經濟成長率，爾後全部重新做過。依照支出面計算的經濟成長率，在後段都沒有太明顯的轉折，我們也就不強調轉折。

2. 圖 1 為自 1962Q1 以來的臺灣經濟成長率，從該圖即可發現臺灣經濟成長率歷經幾個重大轉折。文獻上認為由於臺灣經濟在 1990 年前後成長速度有相當大的差異，1990 年前為高度經濟成長期；1990 年後為溫和持續成長期。此外，

2000 年後，臺灣經濟成長速度又再次下降。作者於表 1 列出不同時期的臺灣經濟成長率偏峰態等資訊，不知為何分段點選在 2016Q4?是否可以從上述之時間切點，透過不同時期的經濟成長率樣貌，選擇合適的模型，或許可讓模型樣本外預測能力更好？

回覆：表 1 列出的統計摘要是敘述統計，沒有分段參考。機器學習是 data-driven，事先不給予太多資料的看法，更不需給予特定分段。機器學習方法論，在數據量適宜之下，必須設計多種分段進行大量學習演算；此案現行分段只是遷就數據的權宜作法，沒有特別意義。

3. Trend 與 seasonal dummy 的設定未明確說明，seasonal dummy 係設定 1 至 4 季的共 3 個虛擬變數嗎? Trend 時間趨勢又係如何設定? 是否每個模型用的季節與趨勢拆解的方法是一樣的？此外，請補充說明季節拆解是用虛擬變數的方式或是用 AR 的形式 (P.21 圖 2 季節拆解不像是虛擬變數的方式)？此外在樣本外期間的 trend 係如何設定?

回覆：

(1) 模型中的 trend 和 seasonal dummy 皆用虛擬變數。Trend 為一等差序列 (1,2,3,……T)， seasonal dummy 配合截距，是 1-4 季三個虛擬變數沒錯；

(2) 季節拆解不是虛擬變數，我們用 STL 軟體內建的 STL (Seasonal Decomposition of Time Series by Loess)；

(3) trend 是規律的序列，樣本外直接延伸。本案模型沒有做季節與趨勢拆解，都是用虛擬變數。

4. H20.autoML 與 LSTM 模型皆僅加入 AR(1)，而 SVM 模型則進一步考慮 seasonal dummy 與 trend，是否可在 H20.autoML 與 LSTM 也考慮 seasonal dummy 與 trend，使不同演算法間有相同的比較基準。

回覆：SV 會特別這樣講，是因為 SVM 是淺層機器學習，因此對於添加變數的表現，我們會特別列出來。autoML 和 LSTM 的模式，對於時間特徵都有內含，因此不會另外解釋，請參考圖六。在期末報告也是全部用一樣的投入變數。

5. 在工具箱簡介方面，能否請研究團隊如何將相關工具運用於預測經濟成長率進行說明，如 LSTM 係先說明 RNN 概念，並用語言模型舉例說明當期與前期訊息連接，但用於經濟成長率預測又當如何解釋這種關係? 又如 P.16 H_2O 圖示中的特徵為何？建議加強範例的說明，以協助讀者了解該技術的用途。

回覆：已在期末報告中述及，如需再添增說明，請再提出。

6. 建議作者以淺顯易懂的表格方式彙整各個模型／技術的優劣處。 並建議作者加強機器學習技術與經濟成長預測的關聯性，以協助讀者了解為何利用機器學習技術能提升預測準確度。尤其在介紹模型時(不同機器學習方法)，建議皆略為說明如何代入變數，據以得到預測值。

回覆：已在期末報告中述及，如需再添增說明，請再提出。

7. P.24 表 2 中的諸多模型設定似未在文中說明。如 NNETTS 模型的相關設定為

何? ARIMA(0,1,0)(1,0,1)+Fourier 模型的 Fourier 並未介紹，而 BaggedModel 有 ETS 與 autoArima 的設定，兩者的差異似未在文中說明。另外表 2 或表 3 出現 ARFIMA 及 QAR 等不同方法，惟文中並未對應說明。

回覆：在 3.1 統計方法說明章節，已分別說明，請參見第 13-20 頁。

8. 本文樣本外預測的做法似乎是先以 1962Q1-2016Q3 為樣本內，估計各模型的參數後，單步預測係代入 2016Q4-2019Q2 的實際值，而如果多步預測則係代入預測值。為何不是用文獻常見的遞迴估計或滾動視窗估計方法進行預測?

回覆：期中報告的多步是採用 recursive 遞迴，但沒有滾動。此案除了小樣本，且機器學習演算耗時，所以就不用滾動視窗。另外，滾動視窗一般不是 ML 使用的方法，因為就機器學習，滾動多期和最後全期，用全樣本訓練即可。最後，我們沒有檢視每個階段的樣本變異性，例如，MAPE 的變化；所以，就小樣本和研究設計，就不採取滾動。

9. 本文的樣本外預測期間為 2016Q4-2019Q2，共 11 季，樣本外期間是否過短，不利比較各模型預測表現的優劣及其穩健性? 應選定不同預測期數 N 的多步預測進行績效比較，才能確保結果的穩健性。或者固定 N，選取不同樣本外期間進行比較等。

回覆：對於不同預測期的組合的確是一個可行作法，但是，除了資料太少，本案的重點在比較三個機器學習對於從不同投入變數組合中提取預測訊號的能力。另外，確保穩健性 (robustness) 是統計分析對於資料中有離群值

(outliers) 或未知異值變異常用的分析技術，重點之一在統計檢定的穩健，機器學習方法並無執行相關統計檢定之需要。

10. P.40 目前本文僅用 GDP 成長率單一數列進行預測，期末報告將從更多解釋變數尋找特徵成分；解釋變數除了文內已打算納入之美國及多國之經濟成長率，在國內變數部分，建議可參採徐士勛等 (2005)、陳宜廷等 (2011) 使用之大量變數，如實質面與金融面變數。或可以機器學習方法，加上獨立成分分析，檢視能否進一步提升績效。

回覆：期末報告已納入解釋變數。我們完成了全面組合分析，故就不用獨立成分分析法降維。以示依合約負責，獨立成分分析法的處理並沒有減少，放在附錄三。(第 141-145 頁)。

11. 在計畫申請書中提到的將殘差類別化 (高估、低估) 並結合其他解釋變數建立殘差類別的分類演算，希望能納入期末報告中。

回覆：期末報告已納入，請參閱第七章。(第 89-93 頁)

(二) 資料說明與處理

1. 目前本文採用生產面 GDP 成長率 (P.2)，但一般各界在觀察經濟成長率時，都是觀察支出面資料，故建議改用支出面 GDP 成長。

回覆：期末報告已一併重新做。

2. 11 月底主計處公布的經濟成長率資料已對歷史資料回溯修正，希望期末報告時能使用修正後的數字進行測試。

回覆：期末報告已更新資料重新做過。

3. 一般而言，GDP 年增率不太會是 I(1)，但作者認為經濟成長率為非定態，建議可以進行單根檢定，以支持作者的推論。若 GDP 年增率為定態，是否還有必要進行 STL 拆解?

回覆：單根問題對機器學習無礙，時間序列性質也非此案重點，STL 的角色只是呈現資料性質，無關機器學習分析。若不妥，期末會將之刪除。

(三) 預測結果

1. P.22 說明的 ARIMA 基準模型為 ARIMA(0,1,1)(1,0,1)，惟表 2 及表 3 卻是 ARIMA(0,1,0)(1,0,1) 並不一致，請再查明。

回覆：謝謝指出，期末報告依然沒有修正，特別致歉。因為用新資料重做，表格只改了新數字，卻忘了同步改模型。用新資料的結果是 ARIMA(2,1,2)(2,0,1)，之後會一致修正。

2. P.33 表 3 可發現，預測績效結果差異頗大，如觀察 RMSE，僅有 7 個低於 1，而 ARIMA 的系列模型，則均超過 8 ；是否代表經濟成長率預測時，採用 Multi-step 預測會使預測結果產生大幅度偏差。研究團隊能否說明原因?

回覆：基於 AR(1) 的 Recursive Multistep，將預測建立在前一期的預測，所以對預測錯誤敏感。ARIMA 的模型或許恰恰反應了這樣的問題。

3. P.33 表 3 AR(1)+trend 與 AR(1)+seasonal dummy 之預測績效(如 RMSE、MAE…等)數字均同，且 ARIMA(0,1,0)(1,0,1) 之各種系列模型預測績效數字亦出現同樣情況，請再查明數字是否有誤 (或解釋可能原因)。

回覆：如果發生這樣情況，多半是添加的虛擬變數的估計參數極小且不顯著。用新資料重做後，並未發現大量相同預測績效數字。

4. P.37 為多步預測結果，由於多步預測係用前 1 期預測結果再對下一期進行預測，因此預測誤差通常會較單步為高。但以 P.37 中，SVM: AR(1)+SeasonDummy+trend 的預測表現不僅明顯優於其他模型，且預測值乎與實際值亦步亦趨，甚至其本身預測誤差如 RMSE 為 0.628 (P.33 表 3) 也低於單步的 0.65 (P.24 表 2)，此情況亦出現在 AutoML 模型，建議作者可補充說明其中原因。

回覆：期中報告的問題是做圖時間落後錯置。期末報告重新採用依支出面計算經濟成長率且重新估計所有結果，重新做了結果。

5. 本文在預測方面，單步 (one-step) 預測即為「向前一期」之預測；多步 (multistep) 即為「間接 (indirect) 預測」或「疊代 (iterative) 預測」。然而，本文建構之模型，無論時間預測模型或機器學習模型，其多為非線性模型。

惟對非線性模型而言，不可使用間接或疊代的方式進行向前兩期以上的預測，否則會產生偏誤；而是應直接建構 t 期與 t+h 期的關係：$\hat{y}_{t+h}=F(y_t)$。

回覆：謝謝指出此點！期中報告內的時間序列模式，只是列入大致比較，並不是本案重心。因此我們沒有針對特定細緻問題處理 (如非線性預測或半定態)，對於不妥的例子，期末報告會予刪除。多步預測有兩種：多個單步預測，也就是 recursive multi-step 和直接多步 (如本問題所指)。期末報告全採用直接多步 $\hat{y}_{t+h}=F(y_t)$。

6. 機器學習方法於單步預測表現劣於時間序列模型可能背後原因為何?

回覆：機器學習方法是資料驅動 (data driven)，因此，對於特定的結果，我們不去做一般性解讀。因為，統計時間序列本就不差，在 M4 的時間序列預測競賽中，深度學習鮮少勝出強化後的時間序列。主要原因在於時間序列的性質對預測的影響，機器學習並沒有特別去掌握。機器學習的時間序列動態預測尚有很大進步空間。本案只在技術上，嘗試用機器學習對時間序列資料預測進行「訓練」，預測好壞的理論原因，暫不深入。

7. 請進一步作樣本外預測績效檢定或是考慮依據 Amstad et al. (2017) 進行轉折點樣本外預測。

回覆：本案焦點放在機器學習的「訓練」方法。「轉折點樣本外預測」的檢定與本案章節內容較無關聯，且似無處安放。

二、 勘誤與補充說明

1. P.2、3、21 多處內文中出現「錯誤!找不到參照來源」，建議修正。

2. P.3 表 1 敘述統計中，最大值與最小值對應之數值應該互換。

3. P.4 圖 1 標題應為 1962Q1~2019Q2，圖上英文標題宜刪，圖 2 標題宜挪移至前一頁尾。

4. P.8 公式(2)迴歸式中的 m 第一次出現應解釋其意義。

5. P.9 公式(4)的 if $\theta_L \leq x_{t-d}$應倒過來為$x_{t-d} \leq \theta_L$。 P.9 倒數第二行，羅吉斯平滑門檻「自我回歸模型」應為「自我迴歸模型」。

6. P.10 LSTAR 模型當中羅吉斯函數之的γ為何?

7. P.16 與 P.20 提到啞變數，其餘多用虛擬變數，建議一致用虛擬變數。

8. P.22，ARMA(0,1,1)(1,0,1) 模型表現「次」之。P.22 倒數第二行圖 3 並無 ACF(1)值。

9. 報告格式：宜注意目錄頁的字型大小，以及表格編號與內文參照。

10. 參考文獻：

    (1) 陳宜廷、「徐世勛」、劉瑞文、莊額嘉應為陳宜廷、「徐士勛」、劉瑞文、莊額嘉

    (2) 內文與參考文獻不一致：如缺 Alysha et al. (2011)

    (3) 參考文獻格式問題：期刊名稱大小寫，如 International journal of forecasting, Ecological modelling,

    (4) 參考文獻中未有 Huang (1999) 一文，此外參考文獻排序應依英文字母順序，惟 Chen and Lin (2000) 一文順序排置錯誤。

(5) 陳宜廷 (2019) 係以反事實分析之「合成控制法」比較臺灣、南韓經濟成長率之差異，並無進行預測，建議刪除此文獻 (P.5)。

回覆：勘誤部份已全部完成修正。

附件三 期末報告意見回覆

期末報告 陳宜廷老師 彙整意見回覆書

一、 計畫總評

- Compares a wide variety of time-series methods and ML methods for Taiwan's GDP growth rate forecasts.
- Many issues have been taken into account.
- Extended information sets: lagged Y, lagged X and combinations.
- Forecast combination: Mean and median forecasts.
- A very detailed empirical study.
- Detailed references.
- Main conclusions: one-step forecasts vs. multi-step forecasts.
- This is really a hard work!
- I only provide some discussions & minor suggestions.

二、 問題與建議

1. Empirical performance is likely to be data-dependent. In particular, the number of time-series observations is quite small for this empirical study. (1962:Q1-2016:Q3, 2016:Q4-2019:Q4)

回覆：受限於研究標的經濟成長率僅有季資料，且年份久遠的樣本資訊對於

預測並無法提供有用訊息，本研究即在此先天限制下，尋找合適的機器學習預測模型。

2. It ought to be important to provide some intuitive interpretations regarding the relative performance of different methods in the one-step-ahead forecasts and in the multi-step-ahead forecasts.

回覆：研究結果發現在單步預測下，統計時間序列方法，尤其是 ARIMA 和 bagged 的模型表現很好，但多步預測下機器學習方法則優於時間序列方法。可能的原因為：遞迴預測運用前期預測值進行未來預測，因為機器學習的優勢在透過資訊型態配適出合適的模型及參數，前期預測誤差可藉由學習過程而減少，故機器學習方法有較佳的預測績效。但單步預測運用前期真實值進行未來預測，統計時間序列方法因估計參數較少，自由度高，故在單步預測時有較佳的預測績效，請參考第 51 頁。

3. It would be interesting to see how the CBC's internal models would perform in comparison with the results reported by this report.

回覆：本文亦提供以機器學習模型對 2020Q1-2021Q4 的預測供央行參考，請參見附錄二 (第 118-140 頁)。

4. This project has a clear forecasting target. It would be good if the discussions of the machine learning methods could be more focused on this forecasting target. It is difficult to understand why these methods might be useful for generating better

forecasting performance via the general discussions.

回覆：機器學習的原則係資料驅動，此與建構一個經濟模型下的因果關係不同。選擇變數之前，並沒有什麼特別的事先經濟考量，僅從資料的相關性來判斷，因此不限於經濟理論模型所討論的控制變數。此外，機器學習也透過資訊型態配適出合適的模型及參數，故可有最佳的預測績效，請參見第 82 頁。

三、 其他意見

1. Figure 8: ARIMA has very similar out-of-sample performance to H_2O.autoML. Right?

   回覆：在單步預測下的結果如此。ARIMA 與 H_2O 有相近的預測績效，如前述，單步預測運用前期真實值進行未來預測，ARIMA 因估計參數較少，自由度高，故在簡易單步預測時有較佳的預測績效，請參見第 51 頁。

2. p.5: Out-of-sample period: 2016:Q4-2019:Q2 or -2019:Q4?

   回覆：本研究之樣本切分，以 2017Q1-2019Q4 為測試期，已校對並修正報告，請參考第 3 頁。

3. p.92: Definition of Kappa: 觀察符合率=? 機率符合率=?

   回覆：Kappa coefficient 是一種統計量化指標，平衡類別之間正確性的一種指標，簡單說就是要把大者恆大的影響消除掉的指標。假設表八"高估"的樣本數遠大於低估時，只需把資料都判給"高估"一類，整體的正確率都很高/錯誤

率很低。計算中的觀察符合率係指預測正確的機率，為總樣本中主對角線預測正確率，即百分比混淆矩陣的對角線元素相加；機率符合率則是把每個類別 (實際第 i 類別的總和乘上預測第 i 類別的總和) 做加總，即

$$\frac{(A+C)}{(A+B+C+D)}\times\frac{(B+C)}{(A+B+C+D)}+\frac{(B+D)}{(A+B+C+D)}\times\frac{(A+D)}{(A+B+C+D)}$$

請參見註釋 10 (第 91 頁)。

期末報告 徐士勛老師 彙整意見回覆書

一、 總評

期末報告清楚明確，計畫團隊認真負責，已達成評估以機器學習模型預測經濟成長率之目的。

二、 問題與建議

由預測結果看來，機器學習模型或許不適合運用在經濟成長率預測，困難處在僅有季資料，樣本數不足。建議未來運用機器學習模型的優勢，透過機器學習模型處理月或日等高頻資料產生的預測結果，再拿來做經濟預測。

回覆：本研究已依照合約完成計畫書所訂目標。委員所提建議的確高明，可為解決方法之一，未來要如何執行，將由委託單位自行決定。

期末報告「經濟研究處/109.6.22」彙整意見回覆書

一、 文章內容相關意見

(一) 模型預測方法及章節調整

1. 研究團隊在合約提案計畫書說明本案主旨為「以臺灣經濟成長率預測為對象，建構一個基於大規模數據，輔以機器學習方法，找出可供參考的成長預測方式」。並提及「將儘可能透過蒐集大量數據來訓練與學習，以提高分類預測精確性。會視數據量，採用 ICA (Independent Component Analysis)方法，萃取少量因子做為預測變數。」請研究團隊說明，為何本研究最後並未採用蒐集大量數據的研究方法的原因。

   另一方面，提案計畫書第 2 頁第 2 段，「臺灣的財經數據，本研究採用之大數據係來自全球開放數據 (Open Data)，開發自動蒐集與清理資料的程式元件，再導入模型估計，以利未來可以自動更新與執行」。檢視目前期末報告之內容，並無相關說明。

   回覆：

   (1) 有關 ICA 本研究已完成相關研究結果，原考量篇幅未列入期末報告正文。依委員意見，今列入附錄三 (第 141-145 頁)。在計畫書中已經有定義：『第 3 步所提到的「大規模其他數據」，就是其他國內外相關財經數據。』並非坊間所謂的數據工程。另外，本案並非未採用大量數據，6.3 節的實做說明了大數據的維度展開和演算規模，以及相關困難。實務上，蒐集大量數據是一個途徑，我們將之列入網路蒐集數據的程式 (相關說明請見附件一)，技術移轉後，貴行想要蒐集多大規模的資料庫，都可以簡易透過網路傳輸。

   (2) 開放數據部份說明是跟著程式，本研究採用 R 語言，有自動蒐集國外開

放數據的程式元件，相關程式說明在附件一 (第 146-148 頁)。

2. 建議研究團隊增加本研究與國外相關研究 (如 Richardson et al., 2018；Bolhuis and Rayner, 2020；Cepni et al., 2020 等) 之方法與研究期間之比較說明，可凸顯與本研究異同之處；亦可說明本研究結論所說之樣本數不足問題是特例或通案。

回覆： 有關國外相關研究之比較說明請見本研究第二章文獻回顧， Feng and Zhang (2014)、Tiffin (2016)、Richardson and Mulder (2018)、Diebold and Shin (2019)、Bolhuis and Rayner (2020)、Cepni et al. (2020) 等使用機器學習方法如 ANN、Elastic Net regression、RF、SVM、NN、partially-egalitarian 最小絕對收斂和選擇算子(least absolute shrinkage and selection operator，peLASSO)、貝氏變異 LASSO 等方法，進行即時預測 (nowcasting)、樣本外一期、樣本外第二期、或樣本外一年預測經濟成長率，均發現預測誤差相較於計量模型減少許多，研究方法和研究期間請見第 7-12 頁。另針對季資料樣本數不足之處理，Chakraborty and Joseph (2017) 英格蘭銀行研究樣本外只用 8 季，以 60 季開始，用逐次增 1 的方法遞迴更新(recursively update)。本研究亦以遞迴 (Recursive) 和直接 (Direct) 預測，請參考第 3 頁。

3. 在工具箱簡介方面，能否請研究團隊說明如何將相關工具運用於預測經濟成長率，如 LSTM 之舉例，請以經濟成長率說明，例如可於流程圖說明 X 可能是落後 1 期的經濟成長率或其他變數、Y 即為經濟成長率預測值等。另建議對本文 RNN 的層級與節點之數量、如何決定等，再稍加說明。

回覆： X 是解釋變數的通稱。作為落後期或其他解釋變數，全看所採用的模式；如第六章的三例之中，Case 1 是經濟成長率的落後期，Case 2 是其他解

釋變數的落後期，Case 3 則兩者混合落後皆有。另有關機器學習模型與經濟成長率預測之結合，將補充說明：如遞迴類神經網路 (RNN) 本身網路架構特性使其本身具有記憶性，故已成功應用在時序數據的處理上，應用在經濟成長率預測上時，投入 X_t (向前一期經濟成長率及其他變數) 透過類神經網絡可預測出 Y_t (經濟成長率預測)；當變數受到長期記憶 (向前多期經濟成長率及其他變數)的影響時，則為 LSTM，請參考第 27-30 頁。

另有關層級與節點之數量，本研究係在模型的合理值範圍內做 grid search，由 RMSE 最小值來判斷最適的層級與節點之數量，且由程式內建，請參考第 31 頁補充說明。

4. SVM 是機器學習中常用的分類器，建議明確說明本文如何將分類問題應用於數值之預測 (如其預測方程式如何產生)。

回覆：對變數型態是 numeric 時，SVM 延伸為 Support Vector Regression (SVR)，即可處理連續的預測問題。應用在經濟成長率預測時，通過經濟成長率或其他變數等特徵值，訓練出這些特徵值和經濟成長率“增加”和“減少”的關係，即通過特徵值劃分指定經濟成長率“增加”和“減少”的邊界，並進而進行預測。我們已於修正稿中註釋 5 說明。(第 25 頁)

5. P.30 第一段描述「遞迴多步預測步驟」，據描述內容是帶入解釋變數真實值的落後期而產生預測值，是否應為「直接多步預測步驟」，而第二段描述的「多步預測步驟」才是「遞迴多步預測步驟」。

回覆： 初稿 P.30 第一段描述為「遞迴多步預測步驟」，第二段描述的「多步預測步驟」是以前期的預測值再進行未來期數值的預測，為「動態多步預測」。

遞迴多步是一步一步遞迴而成多步，請參考第 37-40 頁。

6. 建議調整部分章節排序，例如 1.2 經濟成長率章節 (P.4) 較適合整併放入第 4 章的 4.1 經濟成長季資料 (P.27)。

回覆：感謝行內意見，已調整該章節順序，請參考第 33-34 頁。

7. 本文顯示 H_2O.autoML 的多步預測表現較佳，但 H_2O 演算平台採用多個機器學習模式 (見 P.22 上半部說明)，希望能指出 H_2O 預測績效較佳的是哪些機器學習模式？或者 H_2O 平台不同模型的結果如何彙總成一個「預測數值」？其中各模型的表現如何？相關說明稍嫌不足。

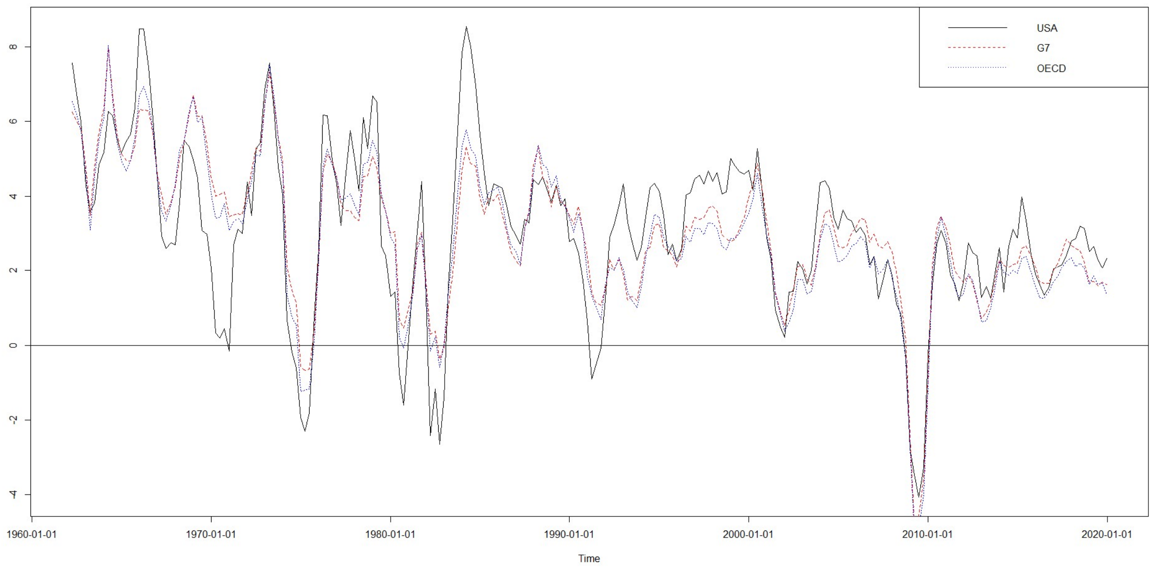
回覆：H_2O.autoML 包含多種普遍使用的機器學習模型，如：監督深度學習 (類神經網絡)、隨機森林 (random forest)、一般化線性模型、梯度提高機器 (gradient boosting machine)、素樸貝式 (naïve Bayes) 等等。以上模式在訓練和驗證兩個子樣本之間經大規模演算後取出各模型的最佳結果，最後以疊積強化 (stacked ensembles) 合併出最適的預測，在機器學習，此方式也稱為委員會方法 (committee approach)，類似統計的預測平均法 (forecasting average)。也就是說：autoML 求得的預測數列並非基於單一種機器學習模型，而是多模型組合模式下的綜合結果，請參考第 25-26 頁。

(二) 變數選擇

1. 第 6 章的外國變數有兩個不同組合，P.51 為美國、OECD 和 G7 經濟成長率，P.84 則為 OECD 經濟成長率、OECD 領先指標成長率和美國領先指標成長率。再者，OECD 與 G7 均已包括美國。請補充說明在同一章節內採用不同組合之

理由。此外，建議另考慮加入我國主要貿易夥伴 (中國大陸) 的經濟成長率，作為其他變數。

回覆：第 6.2、6.3 節的其他變數不同；因為第 6.2 節的結果並不理想，所以第 6.3 節我們重新擷取國外數據以找出最適合預測臺灣經濟成長率的變數組合。其次，第 6.2 節的 OECD 與 G7 雖有包括美國，但是很多時段，走勢並不相同 (如下圖，該圖請參見本研究報告附錄一之經濟成長率：USA，G7，OECD (第 102 頁))，請參考註釋 8 (第 81 頁)。



最後有關加入我國主要貿易夥伴(中國大陸)經濟成長率為其他變數，因大陸季資料偏少，就整體而言，臺灣起自 1962Q1，大陸則更短 (最長季始自 1992Q1，再早多半是年，且早年資料品質堪慮)。在擷取解釋變數時，都必須盡量不影響觀察值，所以類似臺灣的景氣指標 (始 1982M1) 等數據，都必須割愛。

2. (P.51-52) 請研究團隊補充 8 個臺灣總經變數在研究分析中是採年增率或水準

值進行討論。一般而言，貿易條件變動損益、實質國外淨需求與淨國外要素所得似乎較少會以年增率進行分析。上述三個變數以外的國內變數為實質 GDP 的組成項目，與 GDP 多為同步關係，不知選取這些變數的考量因素是甚麼? 如果改採其他影響 GDP 變化的總體變數或金融變數，可否改進預測結果?

回覆： 預測變數為年增率，故本研究解釋變數中的臺灣總經變數亦採年增率。此案以大數據機器學習為主，也就是任何可能預測經濟成長率的其他數據，都可以放進來演算。機器學習的原則係資料驅動，此與建構一個經濟模型下的因果分析不同。本案的國內解釋變數，業已依照央行建議置換，期末報告後，6.2 節已全面用新資料重做，請參考第 76-80 頁。

3. 以機器學習法逐一檢視變數組合預測能力，雖也可以達到降維目的，但耗時。是否有更系統化的方法降維或篩選解釋變數，如先以傳統時間序列模型進行預測，從中挑選幾組預測表現優良的模型，再以機器學習再重新預測，減少計算時間?研究結果呈現方面，亦能事先篩選出最佳的兩、三個模型 (例如 SVM:LY13、ARIMA(2,1,2)(2,0,1) 等)，並以此作為後續比較及延伸測試之基礎，其他被淘汰的模型則可放在附錄。

回覆： 有關系統化的方法降維或篩選解釋變數的方法，本研究以獨立成分分析和主成分分析萃取的結果均不理想，說明如附錄三 (請見第 141-145 頁)。因此，我們才進行逐一檢視。逐一檢視是一個非常繁重的工作，由此應可說明本團隊對所面對的工作認真負責。另在研究結果的呈現上，因單變量 (第 5 章) 與混合落後結構和添加其他變數 (第 6 章) 模型不盡相同，故在多變量模型下仍重估所有結果，並發現 H_2O、LSTM、SVM 表現較佳，故後續僅摘錄 H_2O、LSTM、SVM 與傳統時間序列模型的預測結果。

對於刪減模型的建議，本案的建議是採用預測平均法 (forecasting averaging) 予以包容採納。貴行的篩選建議或更多可能性，未來可由貴行自行執行評估成效。

4. 第 6 章針對自我相關落後模型之落後期數選擇，請補充如何解釋非連續落後期數的選取 (如(1,4)、(2,3) 等)。

回覆： 非連續落後在時間序列 ARMA 架構下並不常見，然而機器學習實務上旨在萃取可預測訊號，並不需要遵從 AR。6.1 節運用非連續落後期數下的自我相關落後模型，在經濟意義解釋上，因經濟衝擊對經濟成長率的影響有時間落後，故出現非連續落後期數含有可預測資訊的狀況。事實上，類似的問題在 ARIMA 也有，即處理長期記憶時間序列的 ARFIMA (F 表示 fractionally)，此情形下 ARIMA(p,d,q)差分項 d 不一定是整數。請參考第 62-63 頁。

(三) 預測結果

1. 不同模型的直接多步預測為幾步，應更明顯指出。例如預測開始站在哪個時點，以及確切要預測時間。

回覆： 這個問題在內文就解釋很清楚了。一旦確立落後資料結構，就會將資料分成兩段：訓練和測試。每個模型的多步皆相同，就是樣本外的測試集：2017Q1-2019Q4；站在時點就是訓練集：1962Q1-2016Q4。

2. P.51 第 6 章起將單變量模式延伸為混和落後結構和添加其他變數，在後續的分析中多是比較 LSTM、SVM 與 H_2O.autoML 三種機器學習演算法的預測表

現，但卻未與傳統線性迴歸模型 (如在 ARIMA 模型中添加其他變數) 進行比較，建議文中可增加與傳統迴歸模型預測表現進行比較。

回覆： 因本文重點不在於和傳統時間序列比較，也沒有利用傳統時間序列建模做預測。雖增添很多額外工作，基於對此問題的善意回應，第 6 章第 2 節製作 ARIMAX 的多個設定做為參考，且在圖形上以橘色標注，以利比較，請參見第 76-80 頁。結果也已全面改寫。

3. 可否請研究團隊運用您認為較佳的時間序列模型與機器學習模型 (尤其是機器學習模型)，以單步、多步 (遞迴、直接) 預測方式，分別估計今年經濟成長率 (yoy) 預測值，並顯示結果，提供委託單位參考。

回覆：本研究運用央行提供的資料，執行 4 個 SVM 模型和 autoML 遞迴多步估計未來兩年 8 季經濟成長率預測，請參考附錄二 (第 118-140 頁)。

4. P.54 第二段，關於 H_2O.autoML 預測結果，目視 C 圖，可發現與 A 及 B 圖不同，樣本內似乎是完全配適，為何在樣本外則存在明顯落後，研究團隊能否提出相關說明。另外，橫跨機器學習模型比較，各模型的樣本內配適度和樣本外的預測績效是否有顯著的正相關性?建議研究團隊可以進一步探討。

回覆： 不只是機器學習，統計時間序列也不敢說樣本內配適度和樣本外的預測績效之間，有一定相關，尤其是未來出現重大 shocks 時。然而，機器學習模型的確可能發生此情形，樣本內配適結果良好，但樣本外預測績效太差，遠大於樣本內的預測績效，此即過度配適 (overfitting)。本研究樣本內外的預測績效差異在容許範圍內，無過度配適問題，請見註釋 7 (第 61 頁)。

5. 圖二十八 (P.89) H_2O.autoML 預測平均 MAPE=0.0899、RMSE=0.314 較表五 (P.87) H_2O.autoML 多步預測 MAPE(0.071)、RMSE(0.29)最小值為高，顯示模型平均方法未改善?

回覆： 模型平均方法係指將多個模式的預測值予以平均，得到樣本外預測的平均值，而圖三十係以此樣本外預測的平均值來計算預測績效。而表五呈現的則是在各種其他變數組合下，篩選出預測績效最佳的變數組合，故表五的 MAPE 及 RMSE 必定小於圖三十。請見註釋 9 (第 85 頁)。

二、 其他意見

(一) 圖表文字與數值呈現

1. 各預測結果的 MAE、MAPE、RMSE、Theil. U、ACF1 圖形，擺放順序宜跟表格 (表二與表三) 一致。

回覆： 感謝行內意見，已依行內意見調整。

2. 圖形標題有的字體太小，不容易看清楚 (如圖 20-22，即 P.73-75)。樣本預測走勢圖形可以只放預測最好那一組，方便閱讀。

回覆： 因須藉由各種預測走勢圖的呈現，歸納出不同機器學習模型下的預測結果特性，故建議保留現有圖形。為便於閱讀，期末報告已將放大圖形。

3. 實證結果中，部分表所列各模型的預測績效與圖所列的結果不一致，例如 P.35 表二的模型名稱 (如 ARIMA(0,1,0)(1,0,1)) 與後續圖三不同，以內文說明來看，

應該是表二有誤、表二與圖三相同的模型下，預測誤差的數值卻不同。表二中的 MAE 有負值並不合理，請再確認。

表二中的單步預測績效，simple random walk 與 seasonal random walk 的 MAE、MAPE、RMSE、Theil. U、ACF1，其與 P.44 表三的多步動態遞迴預測績效數據一模一樣，是否有誤?請進一步檢查全文數字跟圖形是否有誤植情況。

回覆：

(1) 經重估，各模型的 MAE 值均為正值，已修正於期末報告。請參考第 43 頁。

(2) 由於 simple random walk 與 seasonal random walk 的解釋變數僅有截距項及季節虛擬變數，因此不論單步或多步預測時的投入變數均相同，故預測績效相同。

4. P.94 表九之註解，預測「績效」低於時間序列模式應為預測「誤差」低於時間序列模式。

回覆： 感謝行內意見，此為誤植，已修正至報告，請參考第 93 頁。

(二) 參考文獻

1. 內文與參考文獻不一致：P.1 提及 Zhao et. al. 2017 ，惟參考文獻有 Zhao et. al. (2017a) 與 Zhao et. al. (2017b)，建議修正。P.7 第 2 行出現 Huang (1999) 似應為 Huang (2000)。似遺漏部分參考文獻，如第 P.87 中的 Hansen and Racine (2012)，Liao and Tsay (2020)，Bishop (2009) 並未列在後面的參考文獻，請再確認。

回覆：

(1) 已依行內意見校對並修正，將第 1 頁之論文引用校對為 Zhao et al., 2017a。

(2) 第 6 頁引用的黃朝熙老師的論文是“Phases and Characteristics of Taiwan's Business Cycles: A Markov Switching Analysis”，該論文於經濟論文叢刊及黃老師網頁均顯示發表時間為 1999。(3) Hansen and Racine (2012)，Liao and Tsay (2020)，Bishop (2009) 等三篇論文已列在參考文獻。

2. 參考文獻寫法與格式請統一。

回覆： 感謝行內意見，已修正至報告，請參考第 97-101 頁。

3. 內文中若需一次引述多篇文獻之時，建議可使用註解簡化文字以便閱讀。例如 P.4 第 2 段「雖然 Chakraborty and Joseph（2017），Kong et al.（2017）…等研究認可」，建議改以「雖然部分研究 1 認可」，並將欲引述之文獻放入註解 1。

回覆： 已依照行內意見修正。

4. P.92 最後一段，「臨床醫生和研究者們根據這一指標」，請研究團隊增列相關文獻，供讀者參考。

回覆： 混淆矩陣中若有類別間樣本不平均，使得預測時發生大者恆大，也就

是只要把資料都判給數量最多的一類，整體的正確率都很高的情況，而 Kappa 指標就是在判別是否有此情況。臨床應用上，如 Landis and Koch (1977) 以混淆矩陣應用在神經學的試驗上，來分析不同地區病患的病徵差異，同樣使用 Kappa 指標判斷不同地區間不平均的情況。近年研究機器學習模型廣泛應用在睡眠階段的診斷，也有運用 Kappa 指標 (Peker, 2016；Kuo and Chen, 2020)。

(三) 補充說明與勘誤

1. 文章編排請左右對齊。
2. 緒論未增加第 6、7 章之內容概述。
3. P.2 第 1 行得以處「裡」→處「理」。
4. P.3-4 以 2016 年為「基期」的實質 GDP→以 2016 年為「參考年」(因實質 GDP 統計已由定基法改為連鎖法)。「生產面」→「支出面」。P.3-4 說明樣本期間分成兩個子樣本期間，1962Q1-2016Q3 以及 2016Q4-2019Q4；惟 P.27 則是說明子樣本期間分成 1962Q1-2016Q4 以及 2017Q1-2019Q4，前後說明並不一致。

   回覆：本研究之樣本切分，以 1962Q1-2016Q4 為訓練期，2017Q1-2019Q4 為測試期，已校對並修正報告，請參考第 3 及第 33 頁。

5. P.5 說明選擇的樣本外期間必須包含轉折，惟檢視圖 1 的走勢圖，樣本外預測期間卻並未發現明顯的轉折點，宜調整第 5 頁倒數第 6 行之說明。

   回覆：原先使用是依照「生產面」計算的經濟成長率，在 2016Q4-2019Q2 為樣本外預測期間本有一個往下的轉折。後經央行指示改採用「依支出面」計算的經濟成長率，爾後全部重新做過。依照支出面計算的經濟成長率，在後

段都沒有太明顯的轉折，我們也就不強調轉折。

6. P.8 倒數第 2 行提及灰預測 GM(1,1)，但在本文其他地方均未出現，似可加註扼要說明。

回覆：灰預測 GM(1,1)主要功能是應用於系統模型訊息不完全、影響變數紛雜、行為模式不確定的狀況下，進行系統的關聯分析、模型建構及預測。主要優點為所需的數據不用太多且數學基礎簡單，但缺點是模型參數採最小平方法推估，在系統受干擾時會產生偏誤估計。我們已於修正稿中註釋 1 中說明。(第 8 頁)

7. P.28「230 季的觀察值」應為「232 季的觀察值」。
8. P.29 最後一行「一稱」→「亦稱」。
9. P.31「檢」而言之→「簡」。
10. P.52 倒數第 2 行淨國「際」要素所得→淨國「外」要素所得。
11. P.87 從「表五」可知道→從「表六」可知道。
12. P.92 第 3 點 Kappa 的公式中，「觀察符合率」與「機率符合率」 各如何計算?

回覆：此與陳宜廷老師所提「三、其他意見」第 3 題重複，意見回覆同上。

13. P.93 第一段 Youden’s J index 似應等於 Sensitivity+Specificity-1。

回覆：其餘未單獨回覆各點，我們已於修正稿中補充修正。