

東吳大學巨量資料管理學院碩士論文

指導教授：吳政隆 博士

基於生成對抗網路之股票交易行為生成 Stock Transaction Action Generation Based on Generative Adversarial Network

研究生：唐賢融 撰

中華民國一一〇年六月

誌謝

在即將走完碩士生活的現在，能深刻體會到，走在學術研究的這條路上並不是一件容易的事。首先要感謝指導教授吳政隆老師，在研究過程中提供各式各樣的協助與孜孜不倦的提點，在能全心全意專注於研究的同時促使自己有了嚴謹的研究態度與縝密的心思。感謝許晉雄院長，從最早入學時的推薦，在學時的關心，深深感受到院長的照顧。同時也感謝院長以及國立台北商業大學林俊杰助理教授撥空擔任口試委員，讓自己的研究能完整而更具貢獻。

也在此感謝碩士的同班同學，尤其是同研究室的家芸與蕎羽，作為研究室第一批完成論文的他們在論文撰寫過程中提供了許多的幫忙的同時也帶來了歡笑。同時也感謝同研究室的學弟崑亦為研究室的付出。

最後自己要感謝自己的父母與親人，給予自己最大的支持，讓自己可以有自在生活、衣食無缺的環境自由的闖蕩，並在今天得已完成碩士學位。

摘要

隨著深度學習的發展與硬體設備的提升，將深度學習用於金融投資領域也開始蓬勃發展。以往在量化交易的研究普遍關注像是投資標的價格趨勢的監督式學習預測或是使用強化學習建立策略代理人模型。如何向優秀的交易行為學習並生成類似行為的研究相對較少。本研究透過使用生成對抗網路，設計一套從買賣行為學習並模擬類似行為的模型。本研究共包含三個模組，首先建立良好的投資行為生成的模組生成買賣行為，使用線段切割法將歷史價格分段，其中的相對低點與高點作為買點與賣點。接著建立生成對抗式網路，其中的生成模組將多項技術指標的值作為輸入，透過包含兩階層的長短期記憶神經網路模型生成每一日的買賣行為。而相對的，判別模組則是將每日價格與對應的買賣行為作為輸入，透過兩階段的長短期記憶神經網路與接續在後的線性層，嘗試判斷樣本是否來自於真實資料，或是由生成模組所產生。在生成對抗網路學習的過程中訓練出能生成良好的交易策略的生成模型。除了比較最原始的生成對抗網路與監督式學習訓練的結果差異，研究也使用了 WGAN-GP 與 LSGAN，也更進一步地將生成對抗網路與監督式學習結合成一個複合模型。透過實驗結果可以發現基於結合生成對抗網路與監督式學習的模型在二十支股票的平均報酬率比起僅使用監督式學習或 GAN 的模型有著更好的表現。

關鍵字：交易策略、深度學習、生成對抗神經網路

Abstract

Previous studies on quantitative trading have generally focused on supervised learning and forecasting of price trends of investment targets or using reinforcement learning to establish strategic agent models. People have studied how to learn from excellent trading behaviors and to generate similar behaviors. This research uses a generative adversarial networks (GAN) to design a model that learns from historical trading behavior and simulates similar behaviors. This study contains three modules: first, establish a investment behavior mining module to generate buying and selling behaviors on historical trading information, using the piecewise linear representation (PLR) approach to segment historical stock prices, where the relative low and high points are used as buying and selling points. Second, build a GAN model which input the multiple technical indicators to the Generator of the GAN model and the output of Generator is the daily trading behavior . The predicted daily trading behavior is generated through a two-layer long short-term memory (LSTM) model. In contrast, the Discriminator takes the daily adjusted closing price and the corresponding predicted daily trading behavior as the input. There are two-layer LSTM and a linear layer to form neural network of the Discriminator, and the target is whether the trading behavior comes from real data (generated by the PLR) or prediction data (generated by the Generator) The study also uses WGAN-GP and LSGAN models, and combines the GAN and supervised learning models into composite models. The experimental results display that GAN-based models obtain better average return on the 20 stocks which combined generative adversarial networks and supervised learning better performance than models using only supervised learning or GAN models.

Keywords: Trading Strategy, Deep Learning, Generative Adversarial Network

目錄

頁次

誌謝.....	i
摘要.....	ii
Abstract	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	vii
第一章 緒論.....	1
第一節 前言.....	1
第二節 動機與研究方向.....	3
第三節 研究資料範圍與章節內容介紹.....	4
第二章 相關研究.....	5
第一節 投資與技術分析.....	5
第二節 機器學習於金融之應用.....	7
第三節 深度學習於金融之應用.....	9
第四節 生成對抗網路與其在金融之應用.....	11
第三章 研究方法.....	16
第一節 資料前處理.....	17
一、使用技術分析方法計算出特徵.....	18
二、PLR 方法建立買賣點.....	20
三、將特徵與交易行為彙整成樣本資料。.....	22
第二節 生成對抗網路模型建構.....	23

一、生成模型	23
二、判別模型	25
三、損失函數計算	28
第三節 評價方式	30
第四章 實驗結果與分析	33
第一節 實驗數據	33
一、資料來源與處理	33
二、資料分析	33
第二節 實驗設計	36
一、實驗模型	36
二、環境與參數設置	38
第三節 實驗結果	38
一、七個模型的績效表現	38
二、使用不同特徵訓練判別模型的表現	43
三、圖示結果	44
第四節 管理意涵	47
第五章 結論	48
第一節 總結	48
第二節 研究限制	48
第三節 未來展望	48
參考文獻	50

圖目錄

	頁次
圖 1 研究流程.....	17
圖 2 生成模型結構.....	25
圖 3 判別模型結構.....	27
圖 4 在股票 2330、2454、2317、2412、6505、2882、2308、3008、1301、1303 上 最佳模型的預測結果.....	45
圖 5 在股票 2886、2881、1216、1326、2891、3045、2207、2884、3711、2382 上 最佳模型的預測結果.....	46



表目錄

	頁次
表 1 股票產業類別	34
表 2 調整後收盤價敘述性統計資料	35
表 3 股票對應 γ 在 PLR 模型處理後的各項交易行為次數	36
表 4 模型使用參數	38
表 5 測試資料報酬率	39
表 6 報酬最佳模型計數與平均報酬率	40
表 7 模型於 20 支股票中正報酬、負報酬與無報酬數	41
表 8 測試資料夏普比率	42
表 9 夏普比率最佳模型計數與平均夏普比率	42
表 10 平均勝率	43
表 11 不同判別模型輸入的模型平均報酬率	44

第一章 緒論

第一節 前言

自 1988 年，由 James Simons 所成立的公司，透過數理統計、電腦科學的方法從市場上龐大的數據中分析市場與價格的波動模式，讓其公司經營的 Medallion 對沖基金在 1988 年到 2004 年的年均報酬率達到 34%，遠高過索羅斯的年均報酬率 10 個百分點。不同於人為經驗，量化交易具有嚴格的紀律性，也有數理統計上的支持、同時能快速處理大量資訊並即時動作。但另一方面，2017 年 6 月 9 日美科技股大跌，代表美股科技股的那斯達克指數下跌 1.8%，蘋果、Google、亞馬遜、FACEBOOK 公司的跌幅也都超過 3%，法國興業銀行的量化交易分析師 Andrew Lapthorne 推測原因為觸及到多數券商交易員的量化停損點所導致的系統性拋售，這說明了需要注意在同一個市場競爭的交易者，只要資本夠大，就能透過操作市場價格影響對手判斷，若量化交易策略過於簡單呆板或是被對手得知，可能會被短期的價格波動影響決策結果或是反被利用而無法從市場中獲利，所以策略是否具有靈活性也相當重要。

綜觀近年來的科技發展，資料科學可以說是急速發展的一個領域，在 Alphago 以 4:1 擊敗韓國圍棋九段職業棋手李世乭成為了明顯的轉捩點 (Chen, 2016)，不僅越來越多人踏入資料科學的研究中，除了資訊工程以外的科系也開始利用資料科學的技術進行研究。而其中一大重點的深度學習也因為 GPU 分散式運算的成熟而快速崛起，深度學習能直接使用最原始的資料，像是影像的 RGB 數值、工業設備中檢測器讀取到的原始數值、或是網路的文章與回覆，免除大量人工的分析標註或過多繁雜的前處理，透過神經網路的結構強大的擬合能力以及分層的結構自行生成從簡到繁的特徵並且從得到準確的結果，因此深度學習也受到商業界的重視並且被廣泛的應用，AI 概念股與 AI 基金隨著資料科學與深度學習的廣泛應用而出現如雨後春筍般的熱潮，光是國內在 2018 年下半年到 2019 年初就有包括第一金、富邦、元大、國泰及安聯等五

家業者申請發行以投資 AI 相關產業公司股票為主的 AI 基金。而資料科學更是被直接應用在金融投資領域，美國甚至在 2017 年 10 月就成立了以人工智慧建立投資組合的 AI Powered Equity ETF (AIEQ.US)，透過演算法處理大數據的能力建構出主動式的 ETF。

量化交易與資料科學的發展下，接連開發出的機器學習模型也被應用到了金融領域中。而多數的研究集中在使用監督式學習與強化學習 (reinforcement learning, RL) 上。監督式學習一組完整的資料中會包含特徵與這組特徵對應的離散或連續的結果。模型的目標就是在輸入一組特徵下，能輸出正確或近似的預測值，被應用在預測未來價格上 (Lee, Liao, Yeh, & Chang, 2020)。但在制定一個投資策略上，不同於有絕對正確解答的監督式學習問題，策略可能基於一套邏輯，並沒有固定的解答，即便都基於同一套邏輯，可能會因為參數的不同而有不同的行為，策略的好壞評量也有複數個層面需要考量，僅使用監督式學習將模型的學習方向限制在策略的特定參數下的行為，可能會限制了模型探索的空間。RL 不同於監督式學習，模型並沒有絕對的學習對象，影響模型收斂的方向是我們所設定的環境的回饋，嘗試使策略獲得最大的價值。相對於給入更多基於人類經驗的數據，讓模型在沒有任何人為經驗作為參考下自由探索學習，有機會像 AlphaGo Zero (Silver et al., 2017) 一樣，得到更佳策略模型，許多研究也直接將 RL 用做金融策略的生成上 (Xiong et al., 2018)。但即便相對其他遊戲更為複雜，圍棋與現實世界的多變性與不確定性相比，圍棋的環境與規則還是相對於單純。再且，深度學習相當於一個黑盒子，多層的隱藏結構難以解析模型預測結果的重點特徵是深度學習令人卻步的一大致命傷，在沒有限制模型學習的方向下，在一些發生機率過小或不存在於訓練資料當中的情境中，模型可能無法做出保守且合理的行為，很可能因此忽略了金融產業的潛在風險，基於人類經驗的行為模式還是有其必要性存在。

生成對抗網路 (Generative Adversarial Network, GAN) 不同於監督式學習與 RL。相對於 RL 自由的探索環境，生成對抗網路仍是基於一個既定的行為學習，對於模型所學習到的策略與目標策略有一定程度的相似性，而相對於監督式學習絕對的正確解答，生成對抗網路向實際資料的學習並非完全需一模一樣，而是只要具備鑑別器 (discriminator) 無法區分出的相似行為即可。這確保了模型的收斂方向以及一定程度的探索空間。模型不局限於既定的唯一策略，反而能具備一定程度的變化。此外，在預測價格的研究上，不論是單日每分鐘的預測 (Zhou et al., 2018) 或是多天期的隔日預測 (Zhang et al., 2019)，發現使用生成對抗網路有機會能得到比起監督式學習更好的成果。從此可見生成對抗網路在金融領域發展的潛力。所以，本研究將使用生成對抗網路嘗試探索在生成價量資訊以外的金融領域應用。

第二節 動機與研究方向

在基於深度學習的股票交易預測，當前普遍是使用強化學習作為主要的方法，單純的基於最高報酬率與環境互動進行學習。而這樣的學習方式，缺乏金融的理論基礎。我們希望能在股票交易的預測中，能有一個類似顧問的結構，能引導模型判別與收斂的方向。而生成對抗網路的判別模型，正好就類似一個這樣的結構，嘗試讓判別模型學習目標的交易策略生成的行為，或許有機會讓生成模型學習出近似於目標的交易策略的相似行為。但在另一方面，GAN 容易因為判別模型收斂過於快速，導致生成模型常有梯度消失難以訓練的情形，而使用基於 MSE 計算損失函數的 LSGAN (Mao et al., 2017) 或是將梯度偏差加入懲罰項的 WGAN-GP (Gulrajani et al., 2017) 被證實能有更好的收斂效果。另一方面，線段切割法 (Piecewise Linear Regression, PLR) 可以從時間序列資料中找出一系列的相對高點與低點，建立起價格漲跌的趨勢波段 (Wu, & Chang, 2012)。

本研究的目的是希望建構能向既定交易行為學習的模型，此模型能預測出類似目標的交易行為，讓生成的交易行為也具備目標交易行為的部分特性，同時也能獲得正

向報酬。本研究透過生成對抗模型，從歷史的價格趨勢波段中使用線段切割法 (Piecewise Linear Regression, PLR) 產生轉折點，在轉折點上設定合適的買賣點作為買低賣高的目標交易行為，並透過生成對抗網路嘗試讓生成模型作為交易行為產生者，從歷史價量資訊與技術指標產生能仿效 PLR 交易行為的模型，而判別模型作為類似顧問的角色，評斷生成模型的行為是否相似於目標交易策略。過去生成對抗網路在金融上的應用為生成股票價格序列 (Koshiyama et al., 2020)，提供給預測模型更多的訓練樣本已增加預測價格的準確度，或將生成對抗網路直接用於預測未來價格 (Zhou et al., 2018)，在預測未來價格後，能間接的用於交易策略的建立，說明了生成對抗網路在金融領域有一定的效果，而本研究希望透過生成對抗網路在預測股票價格的能力，嘗試探索生成對抗網路是否能直接透過價格資訊，直接產生能獲得正向報酬的交易行為序列，並且嘗試比較基於不同生成對抗網路的模型報酬率的差異。

第三節 研究資料範圍與章節內容介紹

本論文的研究主要對象是台指期的前 20 大成分股，資料使用每日的開盤價、收盤價、最高價、最低價與交易量，使用 2016 年 10 月 20 日至 2018 年 12 月 17 日的股價資料訓練模型，驗證資料為 2018 年 12 月 17 日至 2019 年 7 月 1 日作為參數篩選使用，測試資料為 2019 年 7 月 1 日至 2019 年 12 月 31 日作為主要的預測時間範圍。

本論文主要分成五個章節，各章節架構如下，第一章是緒論，說明前言與各章節內容；第二章是相關研究，在投資與技術分析、機器學習與深度學習在金融的應用、以及生成對抗網路與其在金融的應用這三個領域中分別有哪些相關的文獻；第三章是研究方法，說明模型結構、生成對抗網路的使用、神經網路的訓練方法；第四章為實驗結果；第五章為結論，論述實驗結果的主要發現以及能延伸的研究方向。

第二章 相關研究

第一節 投資與技術分析

在效率市場假設中，市場效率從好到壞分別為強勢效率市場、半強勢效率市場、弱勢效率市場、以及未達弱勢效率市場，而只有在未達弱勢效率的市場中才有可能透過技術分析、使用歷史價格嘗試對未來價格進行預測達到超越大盤的報酬。影響市場效率性包含了信息的傳遞以及市場的流動性等的因素，而在台灣，即便市場股票流動性極高、且市場以高科技產業為主，但在 Wald 弱勢效率市場檢測模型、橫斷面標準偏差 (cross-sectional standard deviation, CSSD) 與橫斷面絕對偏差 (cross-sectional absolute deviation, CSAD) 方法的檢測下，顯示台灣股市也有機會透過歷史價格獲得超越大盤的報酬 (Nguyen et al., 2012)。

技術分析嘗試透過歷史價格、交易量等歷史資訊做為策略的依據，但單純觀察數據難以發掘出數據間的關聯性，許多研究著重在技術分析工具或指標的開發與使用。在技術面分析以及技術指標的協助下，特徵不再只有價格，而有更多樣化、針對不同層面從價格中萃取而得的各式資訊。最早的技術分析工具為日本大米商所使用的 K 線圖，透過每個時間階段內的開盤價、最高價、最低價、與收盤價呈現出一根根的 K 線，相鄰的 K 線又可以看作是更複雜的 K 線組合，進而衍生出許多不同的 K 線策略。使用 K 線對美國道瓊的成分股建立交易策略，發現不論使用單根的 K 線或 K 線組合都無法達到顯著的獲利能力 (Marshall et al., 2006)，但在台灣 (Lu, 2014) 與泰國市場上 (Thammakesorn & Sornil, 2019)，K 線卻能得到不錯的獲利能力，顯示出 K 線應用在不同的市場時，可能會呈現出不同的情況。移動平均線 (Moving Average, MA) 是計算過去數天收盤價的平均價格，並將不同基準日所計算而得的平均價格連線而得到的技術分析方法，透過計算移動平均的方法，可以得到平滑化的數據、忽略小範圍的雜訊波動，觀察出長期的趨勢，也可透過與移動平均線比較，了解現在價格在短期內是偏高或偏低。在包含了金磚五國與一些新興國家使用移動平均策略時，能從中獲得正

向的收益，但僅有少數能優於買入並持有策略 (Sobreiro et al., 2016)。而在台灣，對於沒有發行選擇權的股票，使用移動平均策略的收益能比買入並持有策略來得更好，但對於有發行選擇權的股票，則沒有顯著性的差異 (Chen, C. H. et al., 2016)。也有透過觀察長天期的移動平均線與短天期的移動平均線相互關係從中獲取訊號的方式做為策略的依據，當短天期移動平均線從長天期移動平均線下方往上突破，被稱為黃金交叉，通常認為現在利多轉強，而相反則稱為死亡交叉，而平滑異同移動平均線指標 (Moving Average Convergence / Divergence, MACD)，則是透過長短天期平均線計算出差值，再將不同天的差值透過移動平均計算而得的指標。另外除了一般的移動平均線，也有使用指數方式計算的指數移動平均線 (Exponential Moving Average, EMA)、雙指數移動平均線 (Double Exponential Moving Average, DEMA)、根據時間點不同給予不同權重的加權移動平均線等等不同的移動平均線的計算方式，現在也有新的移動平均方法被開發，而這些不同的計算方式，通常是為了得到更好的平滑化與更低的滯後 (lag)，以作為更好的指標 (Raudys, & Pabarškaitė, 2018)。相對強度指數 (Relative Strength Index, RSI) 是觀察一定時間範圍內的變化，將所有上漲的幅度與下跌的幅度分別加總並計算上漲幅度佔所有波動變化的百分比。可用於了解當前漲跌的趨勢與強度。Pramudya and Ichsani (2020) 分析 MACD、RSI、與布林通道在股票市場的有效性，相對於 MACD，使用 RSI 或布林通道 (Bollinger Bands) 在股票市場有更好的表現。另外 RSI 方法也被證實在印度的股票市場基準指數 (NIFTY 50) 上對價格有良好的預測能力 (Jain, 2019)。

各式的技術指標皆是透過人類經驗的角度設計而出的，從數據的不同層面得出一個數值化的結果，除了方便觀測現在的市場趨勢，在對於金融研究上也都具有一定程度的效果。但價格當中的隱含資訊有些並未被人所發掘並設計成技術指標，另外在參考多個技術指標時，會發生不同技術指標對於未來漲跌預測不一致的情形，仍需要其他工具整合複數技術指標。

第二節 機器學習於金融之應用

即便有了各式的技術指標，在建立交易策略上多數還是依靠投資者的個人經驗。機器學習相比技術指標不僅能夠自行從價格萃取出重要的隱含資訊，也能搭配既有的技術指標，產生績效更好的交易策略。

機器學習在金融領域上已被應用到許多層面。支援向量機 (Support Vector Machine, SVM) 在多維的樣本空間中建立超平面，達到近似於降維的效果，在分類問題與迴歸上皆有應用。Zhou et al. (2020) 將 SVM 應用於中國股票市場上預測未來漲跌，資料使用歷史股價、技術指標等傳統資料來源、與社群平台留言、新聞、與搜尋引擎關鍵字搜尋量作為非傳統資料來源，在不同的股票皆能得到不錯的結果。同樣是用於降維的機器學習方法。主成分分析 (Principal components analysis, PCA) 使用到線性代數中取特徵值、特徵向量的概念，在機器學習中通常使用在建立模型前對大量特徵進行降維，在避免維度過大所造成的過擬合問題的同時，也能將特徵轉成新的特徵向量。Cao and Wang (2020) 使用 PCA 作為股票的價格預測時，訓練神經網路模型前降維的工具，使用 7 項主成分就可以保留 12 項技術指標特徵近九成的資訊量。而在預測方面，決策樹從資料中選擇出一個最有用的特徵，決定一個閾值並將資料分成兩組，再選擇下一個最有用的特徵不斷重複相同動作，透過建立分枝，每個子節點上的資料都具有相似的特徵，已達到最好的預測效果。決策樹的特點還有資料不需經過正規化，可以被用做建立投資組合策略中被用於篩選投資標的，WANG, K. et al. (2019) 針對中國的科技股，使用基本面分析的指標，建立決策樹模型決定多頭股票、空頭股票與不交易股票，接著用快速小波轉換去除雜訊，並用布林帶建立多頭/空頭交易策略，得到比科技股平均收益率更高的收益率。另外，也有相當多的方法基於決策樹開發 (tree based model)，像是隨機森林、梯度提升決策樹 (Gradient Boost Decision Tree)、極限梯度提升算法 (eXtreme Gradient Boosting, XGBoost) 等，這些方法多數是使用多個相對簡單的決策樹集成，能有效提高預測準確度。Basak et al.

(2019) 分析美國股票市場，使用隨機森林與 XGBoost 嘗試預測數天後的漲跌，特徵使用多項技術指標，能有 63.53% 到 95.44% 的準確率。機率論中的貝氏定理 (Bayes' theorem)，在探討兩個或多個事件同時發生的相互關係，而單純貝氏分類器 (Naive Bayes classifier) 則是延伸自貝式定理、假設了所有特徵獨立前提下的機器學習分類法。Bhandare et al. (2020) 將單純貝式分類器用於對分析師進行評價分類與預測，先根據預測準確率、股票漲幅、以及提出的意見量將分析師進行評分成五個等級，而後用高斯貝氏模型進行分類，並在不同年份的資料上進行預測，對於不同年份的預測至少有 8 成以上的準確率。蒙地卡羅模擬法 (Monte Carlo method) 採用大量取樣的方法，透過取樣後的統計分布，回推目標的參數。此方法可以被用於對選擇權的評價，Jiang (2019) 使用美國股票與指數選擇權，嘗試透過蒙地卡羅模擬法評估選擇權的價格，發現計算出來的估計值近似於經常被用於評估選擇權價格的布萊克-休斯模型 (Black - Scholes model)，並且兩者都與實際價格相近。而 Bakar and Rosbi (2019) 使用蒙地卡羅模擬法評估股票的波動度，不論在小樣本或大樣本中，蒙地卡羅模擬法皆能得到可靠的結果。基因演算法 (Genetic Algorithm, GA) 使用了類似於演化生物學細胞當中的染色體遺傳的規則，同時包含了適者生存概念用於樣本擇優篩選與基因突變概念用於加大搜索的空間。其中精英非支配排序基因演算法 (elitist Non-dominated Sorting Genetic Algorithm, NSGA-II) 縮短了 NSGA 模型的計算時間，很大程度的優化了 NSGA，可應用於多目標的問題上。Faijareon and Sornil (2019) 使用 NSGA-II 優化技術指標的參數，並將技術指標輸入決策樹以產生投資策略，相比買入並持有策略或是其他七個基於技術指標的策略有更好的結果。

在降低資料維度、預測、分群等監督式學習與非監督式學習上的問題，都有其對應可以用的機器學習方法，而這些方法用於金融資料的前整理、預測、建立投資策略等都有一定的成效。但股票價格這類與時間序列相關的預測，涉及短中長期趨勢、資訊的遞移，機器學習仍有所侷限，加上對於股票價格的分析，有時須考慮到如新聞、網路輿論等非結構性資料，傳統機器學習仍難以處理這方面的問題。

第三節 深度學習於金融之應用

深度學習框架普遍基於類神經網路，類似於生物神經元的結構，因其非線性的特性，能透過多個神經元有效擬合出複雜的非線性方程，而多層的神經元網路更能從資料中自行生成有用的特徵，從複雜且大量的資料中得到良好的預測結果。Ravi et al. (2017) 使用多層感知機在三階段混和模型中作為初始預測用於預測未來外匯價格具備相當良好的成效，第一階段使用混沌理論 (Chaos theory) 檢測預測目標是否具備混沌，若有則重構相空間，第二階段使用多層感知機做初步預測，第三階段使用 NSGA-II 改善階段二的預測，在這樣的三階段模型中，效果能優於其他兩階段模型與三階段模型，達到良好的預測水準。

但多層感知機並未考量到特徵間的相關性，對此也有研究根據資料特性發展出了不同的神經網路模型結構。卷積神經網路 (Convolutional Neural Network, CNN) 透過建立特徵圖 (feature map) 在輸入陣列上滑動並計算相應結果而得，特性是從多維資料中的不同位置找出相似的特徵，利用了相鄰資料的相關性。此種模型因其結構在影像處理上能比傳統多層感知機有更好的效果，被應用在圖片分類，物件偵測等等問題，也被使用到自然語言處理 (Shen et al., 2014) 和 AlphaGo 圍棋對弈模型中 (Chen, J. X., 2016)。循環神經網路 (Recurrent neural network, RNN) 是設計用來處理有先後關係或時間序列關係資料的神經網路，RNN 的輸入除了當前輸入資訊，也同時包含了上一個時間點的輸出資訊，每一層神經網路計算完成後，會將輸出保留成記憶向量作為下一個時間點的輸入，透過如此方式使模型能保留過去的重要訊息。Wang, J. H. and Leu (1996) 在 1996 年就已經有使用 RNN 預測股價的研究，透過使用時間序列分析方法的 ARIMA 提取特徵，並使用 RNN 訓練，可在台灣證券交易所加權股票指數 (TSEWSI) 得到有效的預測效果。但 RNN 直接將輸出作為記憶向量的方式，在每個時間點都加以更新，容易造成梯度消失，早期的資訊不容易保留到晚期。而長短期記憶網路 (long short-term memory networks, LSTM) 是 RNN 的一種變體，加入了輸入閥

(Input Gate)、輸出閥 (Output Gate)、與遺忘閥 (Forget Gate)，針對記憶向量的改寫、輸入、輸出作了調整，能讓模型自由調整是否遺忘過去的資訊或記錄更新的資訊，使得較為重要的資訊得以保留，有效解決簡單循環神經網路梯度消失的問題，交通流量預測 (Zhao et al., 2017) 以及自然語言處理 (Ma et al., 2018) 等不同的領域中被廣泛使用。LSTM 亦被用在股價預測，在台灣股票市場中，Lee et al. (2020) 針對台灣 50 指數作為數據來源，使用 20 天的價格資訊與五個不同的技術指標組合輸入到 LSTM 中進行訓練，發現使用 MACD 的模型相對使用其他技術指標的模型表現較佳但預測能力相差不大，不論使用哪一個技術指標皆能有七成以上的正確率。另外，Lai et al. (2019) 使用 ARIMA 結合 LSTM，在台灣個股能得到一定程度的預測結果。而 Kim and Won (2018) 針對韓國綜合股價指數，將歷史資料使用 GARCH，EGARCH 和 EWMA 三個基於 GARCH 的模型估計出參數，並連同當天價格資訊一同傳入神經網路模型，輸入 22 個交易日的資訊以預測第 23 天的價格。比起使用深度前饋神經網路或指使用其中一個或兩個基於 GARCH 的模型參數作為特徵，將三個基於 GARCH 的模型參數一併使用配合 LSTM 能得到最好的預測結果。

但即便能掌握相當程度的未來趨勢，也不代表能建立良好的投資策略。不同於監督式學習，RL 無須有正確標準的答案提供給模型訓練，RL 透過代理人模型與環境互動，並能從中透過反饋的獎勵自行學習出讓獎勵最大化的規則，故使用深度學習建立投資策略的研究多使用 RL。Deep Q-Learning (Mnih et al, 2013) 使用神經網路取代 Q Learning 中的 Q table，在避免維度爆炸的同時，神經網路的特徵提取能力也讓 Deep Q-Learning 的策略能更好。Park et al. (2020) 使用 Deep Q-Learning 建立資產組合的配置策略，比起使用買入並持有、隨機選擇、動量與迴歸等四種傳統策略模型，Deep Q-Learning 模型所訓練出來的策略績效多數優於傳統策略。而 DDPG (Lillicrap et al., 2015) 使用 actor-critic 方法，策略模型中包含了行為模型 actor 負責做出行為的機率分布輸出，而評判模型 critic 負責對 actor 的行為進行評分，actor 基於 critic 的評分進行更新學習，critic 則對於實際環境分數回饋更心學習，比起 Q Learning 只能預測離散

的行為，DDPG 可以做出連續值的行為輸出。DDPG 可被應用於投資組合策略的建立 (Liang et al., 2018)，Xiong et al. (2018) 使用了道瓊指數中包含的 30 支股票作為實驗的資料集訓練 DDPG 模型，並與道瓊工業平均指數和最小平方差投資組合分配策略進行比較。使用了最終投資組合價值、年化回報、年化標準差以及夏普比率做為評估結果的指標。在使用了兩年半的資料進行測試後，DDPG 的模型相較於其他兩種投資策略，不論在比較最終投資組合價值、年化回報及夏普比率，都有較好的表現。說明了 DDPG 具有生成比道瓊工業平均指數與最小方差投資組合分配方法更好的交易策略。

深度學習具備了從特徵中產生複雜特徵的特性，以及模型的靈活性，在金融領域中可被應用於對於金融價格的預測或是交易策略模型的生成。同時，深度學習具備高度的相容性，可以與其他機器學習整合成複合模型，提高模型預測價格的精準度或是策略模型行為的報酬率。但因為比傳統機器學習有著更多的參數，在訓練時需要更多的樣本。

第四節 生成對抗網路與其在金融之應用

比起其他機器學習模型，深度學習能應用的層面相當廣泛，除了能有效的處理音訊、圖片、影像等等的非結構性資料，也能應用在數據生成這類非監督式學習的問題上，Goodfellow et al. (2014) 所提出的 GAN 模型是最早的生成對抗網路模型，在 GAN 模型中有兩組神經網路模型，分別為生成模型與判別模型。生成模型透過輸入得到的資訊，企圖產生接近真實資料分布的生成資料，從而使得判別模型將生成資料判別為真實資料，而判別模型則嘗試從真假資料中辨認出真實資料與生成資料，利用兩個模型的互相抗衡學習，得到良好的生成與判別模型，在生成影像、圖片、音訊上都有很好的表現。而在金融上，Goh and Lai (2019) 將生成對抗網路用於建立投資標的的關聯網路，用生成對抗網路建立美元與其他 15 種貨幣匯率的關聯網路，除了可以生成更為貼近實際資料的數據外，也藉以觀測各個貨幣每年的關聯性變化。另外，

也可以使用生成對抗網路生成投資標的的共變異矩陣，Marti (2020) 以 S&P500 中的 500 支股票隨機抽取部分股票的一年資料所計算出來的共變異矩陣作為樣本，嘗試生成相同分布的新共變異矩陣。實驗得出，生成的資料與實際資料具有相近的分布。生成對抗網路也被用預測價格，對於一日內的價格預測，Zhou et al. (2018) 將當天的每分鐘歷史資訊與歷史價格作為生成對抗網路的輸入，嘗試生成下一分鐘的價格預測，LSTM 用作生成模型，CNN 用作鑑別模型，構成生成對抗網路，同時將方向正確性與誤差大小加入損失函數中，使用 42 檔中國市場股票 2016 年一整年的每分鐘資料，與 GARCH、ANN 和 SVM 三種模型比較，並使用均方根相對誤差(Root mean square relative errors, RMSRE)和方向預測精度(Direction Prediction Accuracy, DPA)做為評估指標，基於生成對抗網路的調整模型在最多種情境下獲得最好的成果，可以被使用在當沖交易中。而對於多天的價格預測，Zhang et al. (2019) 使用開、高、低、收、交易量、周轉率與五日的移動平均做為特徵，並使用了七個指數與股票，並使用 LSTM 作為生成器以及 MLP 作為鑑別器，使用生成器透過現在狀態生成未來狀態，在多項績效指標上，生成對抗網路的模型表現皆優於其他三項對照方法。除了 LSTM，也可以使用最初用於解決語音生成問題、結合循環神經網路與卷積神經網路的時間卷積網絡(Temporal Convolution Network ,TCN) (Bai et al., 2018) 作為生成模型，Wiese et al. (2020) 將 TCN 結合 GAN 開發出 Quant GAN 模型，使用 S&P500 指數價格作為樣本資料生成新的資料，結果證實 Quant GAN 在多數評估指標優於另外兩個對照模型。生成對抗網路也被用於生成更多的金融數據用做其他模型的訓練，Yoon et al. (2019) 基於 GAN 開發出 TimeGAN，當中的生成器與鑑別器皆是由 LSTM 構成，用於生成近似於真實數據的資料，資料使用 GOOGLE 2004 年到 2019 年的每日收盤資訊，包含交易量、開盤價、最高價、最低價、收盤價與調整後收盤價，並與其他六種模型比較，在 t-SNE 的表現上 TimeGAN 相比其他方法，生成的資料與實際資料重疊度更高，同時對未來價格的預測表現更好，說明 TimeGAN 優於其他方法。TimeGAN 更被延伸用於判斷投資標的的過去資訊對未來價格的預測能力。Grilli and Santoro (2020) 使

用了 TimeGAN 能生成時間序列樣本資料的能力，將 TimeGAN 做為分析投資標的預測能力的工具，研究對於能 24 小時交易的加密貨幣與受制於市場交易時間的股票在預測能力上是否具有差異。使用 google、amazon 兩支股票與以太幣 Ethereum (ETH)、Bitcoin Cash (BCH) 等加密貨幣，並測試兩種情境，分別是僅使用每日開盤價與收盤價兩項資訊或是使用每日開盤價、收盤價、最高價、最低價與成交量等五項資訊。並透過三個方法分析投資標的的歷史價格資訊對未來價格的預測能力。分別是透過 t-SNE 分析並視覺化、透過判別模型分析真實資料與預測模型誤差、以及透過使用生成資料訓練預測模型的預測能力。發現加密貨幣生成資料與原始資料的重疊度比起股票來得更高，生成資料的預測能力也比股票來的好，顯示加密貨幣的價格資訊含有更多能預測未來價格的資訊量。

基於 GAN 開發的 Conditional GAN (Mirza & Osindero, 2014)，在生成模型與辨別模型中的輸入端加入條件特徵，引導模型的學習的方向，最早的研究是使用 MNIST 數字圖像生成，在模型的輸入端加入數字的類別，另外還有使用在圖片標籤自動生成，有一定程度的效果。為了避免數據分析時所碰到的資料隱私的問題，Efimov et al. (2020) 使用 Conditional GAN 解決，將三個來自美國通運的金融產業中的數據集作為樣本資料訓練 Conditional GAN 生成新的資料，並分別使用真實資料與生成資料訓練監督式學習，結果顯示生成資料所訓練出來的模型表現僅略差於真實資料。使用 t-Distributed Stochastic Neighbor Embedding (t-SNE) 算法呈現真實資料與生成資料在二維平面上的散布情形。從圖可以看出真實資料與生成資料具相似的分布情形。顯示透過生成對抗網路能生成的資料能有效代表真實資料，同時能解決真實資料的侵犯隱私的疑慮。Conditional GAN 也能被用於生成金融數據資料。Koshiyama et al. (2020) 使用 Conditional GAN 生成大量資料，並用生成資料來優化和集成多個策略模型。使用了 579 種資產價格作為樣本資料，包含了英美股票、貨幣匯兌、債券等等。將 Conditional GAN 與 Stationary Bootstrap 做比較，實驗結果顯示使用 Conditional GAN 生成的資料所訓練出來的模型結果比起 Stationary Bootstrap 能有更好的收益表現。

而 GAN 模型有梯度消失導致難以收斂的問題。Arjovsky et al. (2017) 提出了使用 JS Divergence 來衡量生成資料分布與實際資料分布的偏差情況。並證明若判別器函數輸出能符合利普希茨連續 (Lipschitz continuity)，則能有效地解決梯度消失的問題。而最早實作此理論的方法，是使用使用權重修剪 (weight clipping) 方法。設定值域，在倒傳遞更新權重後，若權重超出值域則將權重重新調整到值域的範圍內。這樣的方法被稱為 Wasserstein GAN (WGAN)。WGAN 比 GAN 更容易收斂，且提升了圖像生成品質。但權重修剪方法是將超過值域的權重重新設定到值域上下界，使得權重過度集中在值域的最大值與最小值。模型仍無法被有效訓練。這樣的實作方法仍不完美。Gulrajani et al. (2017) 使用 gradient penalty 方法實作 WGAN 的模型，又被稱作 WGAN-GP。透過使用隨機變數作為權重產生出生成樣本與實際樣本間的過渡樣本。計算過渡樣本在模型上倒傳遞後的梯度，得到此梯度與 1 的偏差。將此偏差作為損失函數的調整項。研究結果說明此方法比起使用權重修剪方法更易收斂且能得到更好的圖像生成結果。計算過渡樣本在模型上 backpropagation 後的梯度，得到此梯度與 1 的偏差。將此偏差作為損失函數的調整項。研究結果說明此方法比起使用 weight clipping 方法更易收斂且能得到更好的圖像生成結果。WGAN-GP 也被用作對未來價格序列模擬生成。Mariani et al. (2019) 將所有股票價格與先驗分布做為特徵，生未來每支股票的收益，並透過重複的生成得到未來收益的機率分布。接著使用多目標遺傳算法 NSGA-II 對現有的資產進行配置，以達到最大化收益、最小化風險的投資組合。比起 Markowitz 投資組合優化，此方法能有效提高收益並減少風險。而為了避免模型的過擬合的問題，Lezmi et al. (2020) 使用 WGAN-GP 生成 S&P500 與 VIX 等金融指標每日的變化。此方法能夠生成具備真實資料統計特性的樣本資料。WGAN-GP 結合 Conditional GAN 而成的 Stock-GAN (Li et al., 2020)，則用於生成股票市場逐筆訂單資料，以模擬實際股票逐筆訂單的情形。使用 google 的真實訂單流量資料。以前 20 筆訂單資訊做為特徵生成下一筆訂單資料。Stock-GAN 生成資料與原始資料間的差異

比起另外兩個對照模型來得更小。表示相比起另外兩個方法，Stock-GAN 生成的訂單資料更貼近真實資料。

LSGAN (Mao et al., 2017) 則是將損失函數從交叉熵換成了均方誤差 (mean-square error, MSE)。在生成圖片的問題上模型的收斂能比 WGAN 穩定且更加快速。LSGAN 所生成的圖片相對於原始的 GAN 也更加良好。LSGAN 也被應用在分散式阻斷服務攻擊 (DDoS) 的研究中。Flash Crowd 不同於 DDoS，是代表因某些事件導致大量合法使用的客戶突然湧現的情形。Sun et al. (2017) 使用 LSGAN 模擬 Flash Crowd 的行為並生成更多的訓練樣本。結果表明使用 LSGAN 生成的行為比 GAN 表現得更好。以隨機森林建構的防禦系統不能辨別生成模型產生的行為與 flash crowd 的差異。生成對抗網路也可以設計成多層的結構。Zhang et al. (2017) 透過使用兩層的生成對抗式網路。第一個生成模型生成畫素較低的圖片，第二個生成模型根據第一個生成模型的結果生成畫素較高的圖片。相比直接生成高畫素圖片，此方法能有更好的結果。

生成對抗網路目前在金融方面的使用，多數是透過既有的價量資訊試圖生成更多的價量資訊，並將產生的價量資訊使用在其他模型進行訓練，可用於彌補深度學習模型需要大量資料訓練的問題。不像監督式學習有明確的收斂方向，生成對抗網路對模型收斂方向的限制僅有是否來自原始的資料，這讓模型探索得更廣。也不同于 RL 只看中最後的績效，而失去了對學習方向的控制。另一方面，即便深度學習具備了從原始價量資訊中萃取有效特徵的能力，但在所探討的金融文獻中，多數都還是有使用到技術指標做為特徵。也有論文探討使用不同技術指標訓練模型的績效比較。說明使用技術指標在金融相關領域研究的訓練上有其作用。

第三章 研究方法

整個研究過程呈現在圖 1，主要可以分成資料處理、模型訓練與模型評估，資料前處理共有三個步驟：

步驟一：使用每日開盤價、收盤價、最高價、最低價、交易量與各式技術分析方法得到特徵。

步驟二：使用調整後收盤價與 PLR 方法計算得到每日的交易行為，交易行為包含買入、平倉或不交易。

步驟三：將特徵與交易行為配對成樣本資料，作為生成對抗網路訓練的資料來源。

而模型訓練則將處理過的樣本資料作為生成對抗網路訓練的資料來源，進行下列步驟：

步驟一：生成對抗網路的生成模型透過特徵，預測交易行為。

步驟二：交易行為與調整後收盤價透過生成對抗網路的判別模型，辨識交易行為是否為目標交易行為或預測交易行為。

步驟三：判別模型與生成模型的效果評估與訓練。

模型評估則會根據模型在測試資料上的報酬率、夏普比率以及勝率做比較。

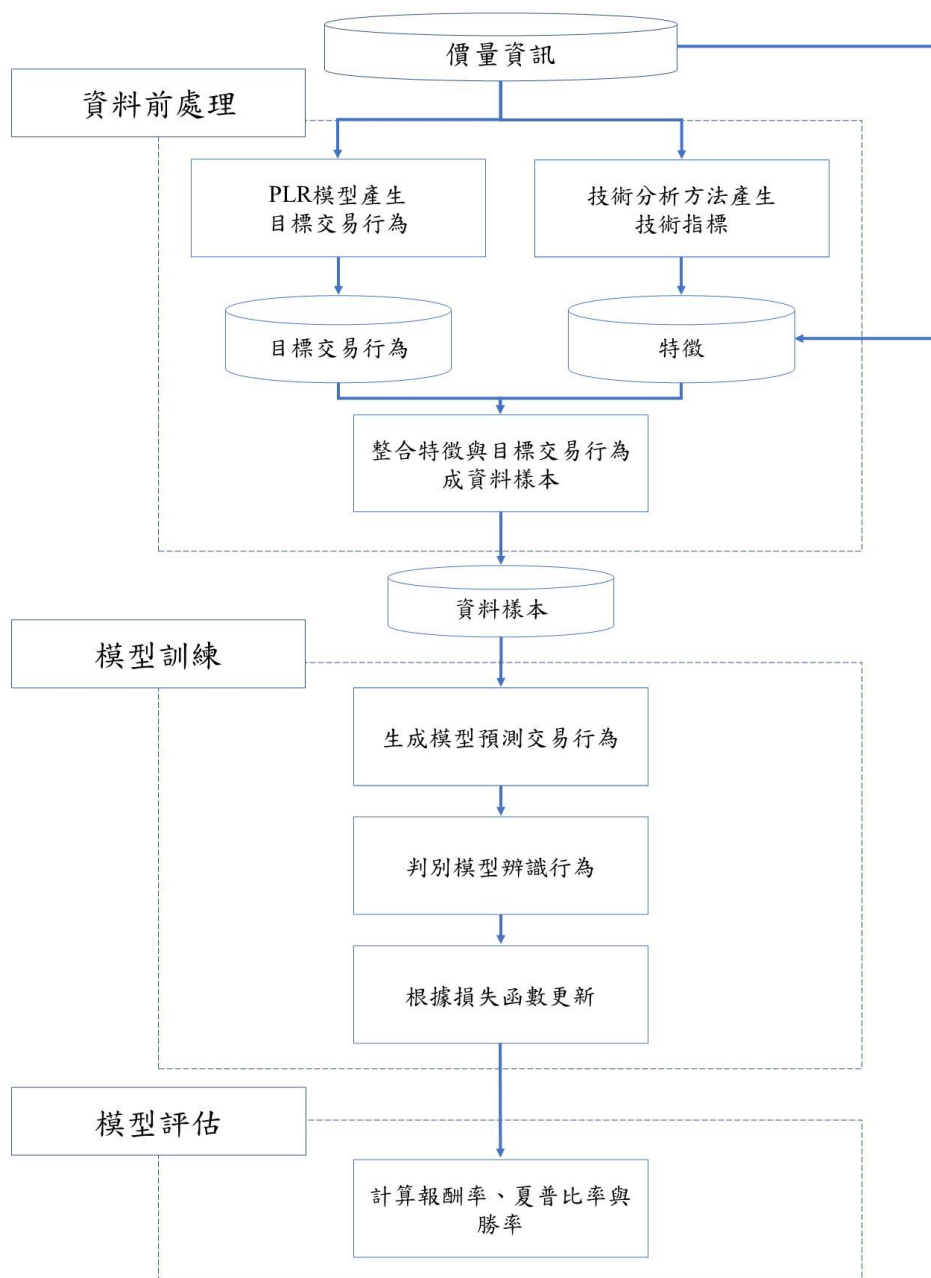


圖 1 研究流程

第一節 資料前處理

本節介紹資料前處理的流程。首先使用每日開盤價 (O)、收盤價 (C)、最高價 (H)、最低價 (L)、交易量 (V) 計算出各式技術指標作為每日的特徵。另外一方面，

使用調整後收盤價 (ADJC) 找出漲跌趨勢的轉折點作為目標交易行為。在完成前面兩個步驟後取一段連續時間的特徵與相對應的目標交易行為作為一組樣本資料。

一、使用技術分析方法計算出特徵

本研究使用技術分析方法計算出技術指標作為特徵，共使用了 97 項技術分析方法，包含了：

- (1) K 線相關的技術分析方法如裁縫線指標 (Heikin-Ashi)。
- (2) 使用多種價量資訊或多天期價格進行計算平均而得的技術分析方法如雙指數移動平均線 (Double Exponential Moving Average, DEMA)、指數平均線 (Exponential Moving Average, EMA)、Fibonacci's Weighted Moving Average (FWMA)、High-Low Average、High-Low-Close Average、Hull Exponential Moving Average(HMA)、Ichimoku Kinkō Hyō、Kaufman's Adaptive Moving Average、Linear Regression、Midpoint、Midprice、Open-High-Low-Close Average、Pascal's Weighted Moving Average、Wilder's Moving Average、Sine Weighted Moving Average、單純移動平均線 (Simple Moving Average, SMA)、Supertrend、Symmetric Weighted Moving Average、T3 Moving Average、Triple Exponential Moving Average、Triangular Moving Average、Volume Weighted Average Price、Volume Weighted Moving Average、Weighted Closing Price、Weighted Moving Average、Zero Lag Moving Average。
- (3) 與績效相關的技術分析方法如 Log Return、Percent Return。
- (4) 與統計相關的技術分析方法 Entropy、Kurtosis、Mean Absolute Deviation、Median、Quantile、Skew、Standard Deviation、Variance、Z Score。

- (5) 與動量相關的指標如動量震盪指標 (Awesome Oscillator)、絕對價格震盪指標 (Absolute Price Oscillator)、乖離率 (Bias)、Balance of Power、情緒指標 (BRAR)、Commodity Channel Index、地心引力線 (Center of Gravity)、錢德動量震盪指標 (Chande Momentum Oscillator)、估波指標 (Coppock Curve)、費雪變換 (Fisher Transform)、KDJ 指標、MACD、動量指標 (Momentum)、Percentage Price Oscillator、Psychological Line、Rate of Change、RSI、Stochastic Oscillator、Stochastic RSI、Trix、True Strength Index、Ultimate Oscillator、Williams %R。
- (6) 與趨勢相關的技術分析方法如 Average Directional Movement Index、Archer Moving Averages Trends、Aroon & Aroon Oscillator、Choppiness Index、Chande Kroll Stop、Decay、Decreasing、Detrended Price Oscillator、Increasing、Parabolic Stop and Reverse、Q Stick、Vortex。
- (7) 與波動度相關的技術分析方法如 Aberration、Acceleration Bands、Average True Range、布林通道、Donchian Channel、Keltner Channel、Mass Index、Normalized Average True Range、Price Distance、Relative Volatility Index、True Range
- (8) 與交易量相關的技術分析方法如 Accumulation/Distribution Index、Accumulation/Distribution Oscillator、Archer On-Balance Volume、Chaikin Money Flow、Elder's Force Index、Ease of Movement、Money Flow Index、Negative Volume Index、On-Balance Volume、Positive Volume Index、Price-Volume、Price Volume Trend。

計算完成後的技術指標加上原有的價量資訊共用 N 項特徵，特徵共包含了 $S + T^{train}$ 日，其中 S 表示在模型中基於 S 日的交易資訊做隔日的交易行為預測， T^{train} 則

是訓練資料長度。故訓練資料特徵 $F = [f_{-S+2}, \dots, f_t, \dots, f_{T^{train}}]$ ，而其中 $f_t = [f_t^1, \dots, f_t^N]$ 表示第 t 日的 N 個特徵。

二、PLR 方法建立買賣點

研究使用以 PLR 方法建立目標交易行為，使用 PLR 方法可以從歷史價格找出趨勢波段，並產生在趨勢轉折點交易的買賣行為序列。步驟如下：

(一)、使用線性迴歸找到所有的可能交易點。

PLR 方法使用調整後收盤價 $ADJC = [adjc_1, adjc_2, adjc_3 \dots]$ 計算，其中 $P = [p_1, p_2 \dots, p_t, \dots, p_{T^{train}}] = [adjc_2, adjc_3 \dots, adjc_{t+1}, \dots, adjc_{T^{train}+1}]$ 為訓練資料中所有時間點 $[1, 2, \dots, T^{train}]$ 的隔日調整後收盤價。將所有調整後收盤價的起始日第 1 日與結尾日第 T^{train} 日做為最起始的兩個轉折點 $D = [1, T^{train}]$ ，其中 $D = [d_1, \dots, d_i, d_{i+1}, \dots]$ 為轉折點數列，將兩相鄰轉折點 d_i 與 d_{i+1} 建立一個通過這兩點的線性方程式，即：

$$\tilde{p}_j^{d_i, d_{i+1}} = \frac{d_{i+1} - d_i}{p_{d_{i+1}} - p_{d_i}}(j - d_i) + p_{d_i} \quad (1)$$

其中 j 代表介於 d_i 和 d_{i+1} 的每一個交易日：

$$j \in \{d_i + 1, d_i + 2, \dots, d_{i+1} - 1\} \quad (2)$$

並計算出每日對應之值：

$$\tilde{P}^{d_i, d_{i+1}} = [\tilde{p}_{d_{i+1}}^{d_i, d_{i+1}}, \tilde{p}_{d_{i+2}}^{d_i, d_{i+1}} \dots, \tilde{p}_{d_{i+1}-1}^{d_i, d_{i+1}}] \quad (3)$$

接著比較 P 與 $\tilde{P}^{d_i, d_{i+1}}$ 的差距得到差距最大的日期 $M^{d_i, d_{i+1}}$ ，即：

$$M^{d_i, d_{i+1}} = \underset{j}{\operatorname{argmax}} p_j - \tilde{p}_j^{d_i, d_{i+1}} \quad (4)$$

而閾值 β 的計算，使用所有資料的收盤價，計算出標準差 s ，並設定一參數 γ 相乘得到閾值 β ，即：

$$\beta = \gamma s \quad (5)$$

，且若時間點 $M^{d_i, d_{i+1}}$ 的價格 p 與線性方程式值 \tilde{p} 的差距 $|p_{M^{d_i, d_{i+1}}} - \tilde{p}_{M^{d_i, d_{i+1}}}^{d_i, d_{i+1}}|$ 大於閾值 β ，則將這個日期 $M^{d_i, d_{i+1}}$ 新增為新的轉折點，即：

$$M^{d_i, d_{i+1}} \in D_{new} \text{ if } |p_{M^{d_i, d_{i+1}}} - \tilde{p}_{M^{d_i, d_{i+1}}}^{d_i, d_{i+1}}| > \beta \quad (6)$$

D_{new} 即是考慮完 D 中所有相鄰轉折點後形成的新的轉折點序列，包含了符合條件的新的轉折點以及既有的轉折點從小到大排序而成，並重新按以上方法嘗試從新的轉折點集合 D_{new} 嘗試找出新的轉折點，直到所有時間點價格與線性方程式的最大差距

$|p_{M^{d_i, d_{i+1}}} - \tilde{p}_{M^{d_i, d_{i+1}}}^{d_i, d_{i+1}}|$ 皆小於 β ，代表已經找到所有的轉折點，則此轉折點序列為 D_{final} 。

(二)、根據轉折點與價格決定每個時間點的交易行為

找出所有轉折點 $D_{final} = [d_1, \dots, d_i, d_{i+1}, \dots, d_l]$ 後，根據順序從最早的時間點判斷每個時間點 $[1, 2, \dots, T^{train}]$ 的交易行為 $A = [a_1, \dots, a_t, \dots, a_{T^{train}}]$ ，其中 a_t 為時間點 t 的隔日交易行為。只考慮全買與平倉而不考慮放空的情況下，假設一個長度為 T^{train} 的向量 $State = [state_1, \dots, state_t, \dots, state_{T^{train}}]$ 。對於當前時間點 t 的持有狀態，當 $state_t = 0$ ，表示現在為未持有股票的狀態，而若 $state_t = 1$ ，則表示當前為購入股票、持有股票的狀態。在最先開始 $state_1 = 0$ ，若時間點 $t = d_i \in D_{final}$ 是轉折點且當前 $state_t = 0$ ，且下一個轉折點價格 $p_{d_{i+1}}$ 比此轉折點價格 p_{d_i} 還高，則 $a_t = 1$ ，代表在這個時間點執行買入，並且下一天的持有狀態 $state_{t+1} = 1$ ；若時間點 $t = d_i \in D_{final}$ 是轉折點且現在為持有狀態 $state = 1$ ，且下一個轉折點價格 $p_{d_{i+1}}$ 比此轉折點價格 p_{d_i} 低或是當前為最後一個轉折點 $d_i = d_l$ ，則 $a_t = 2$ ，表示平倉賣出，並且轉變為未持有狀態 $state_{t+1} = 0$ ，其他轉折點則與未被標記為轉折點的日期一同標註為不交易 $a_t = 0$ ，即：

$$a_t = \begin{cases} 1, & \text{if } state_t = 0 \text{ and } p_{d_i} < p_{d_{i+1}} \\ 2, & \text{if } state_t = 1 \text{ and } (p_{d_i} > p_{d_{i+1}} \text{ or } p_{d_i} = p_{d_i}) \\ 0, & \text{otherwise} \end{cases}, t \in [1, \dots, T^{train}] \quad (7)$$

且

$$state_{t+1} = \begin{cases} 1, & \text{if } state_t = 0 \text{ and } p_{d_i} < p_{d_{i+1}} \\ 0, & \text{if } state_t = 1 \text{ and } (p_{d_i} > p_{d_{i+1}} \text{ or } p_{d_i} = p_{d_i}) \\ state_t, & \text{otherwise} \end{cases}, t \in [1, \dots, T^{train}] \quad (8)$$

三、將特徵與交易行為彙整成樣本資料。

模型訓練需要固定模型的交易期間長度 TP ，並決定每個交易日所參考的歷史資料天數 S 。所以對於一筆完整樣本資料 \mathbb{X} 是由長度為 TP 的 X 構成，即：

$$\mathbb{X} = [X_1, \dots, X_k, \dots, X_{TP}] \quad (9)$$

而每一個 X 則具備 S 個時間長度的特徵資料，取自資料特徵 F ，即：

$$X_k = [x_k^1, \dots, x_k^S] = [x_{k-S+1}, \dots, x_k] = [f_{m-S+k}, \dots, f_{m+k-1}] \quad (10)$$

其中 m 是一筆樣本資料 \mathbb{X} 的起始交易日，每一個樣本資料 \mathbb{X} 不會有重疊的交易日，相互間隔 TP 个交易日，故 $m \in M = [1, 1+TP, 1+2TP, \dots]$ 。訓練模型時，模型會從訓練資料的第 m 天歷經長達 TP 个交易日。而在第 k 个交易日，模型會根據前 S 个交易日的特徵 $X_k = [f_{m-S+k}, \dots, f_{m+k-1}]$ 預測隔一个交易日的交易行為。而相對應的目標交易行為 Y 為：

$$Y = [y_1, \dots, y_k, \dots, y_{TP}] = [\delta(a_m), \dots, \delta(a_{m+TP-1})] \quad (11)$$

，其中 δ 為獨熱編碼函數(One-Hot Encoding function)且：

$$y_k = [y_k^0, y_k^1, y_k^2] \quad (12)$$

y_k 是 a_{m+k} 經過獨熱編碼函數後的結果，故 $y_k^{a_{m+k}} = 1$ ，另外兩個等於0，即：

$$y_k^u = \begin{cases} 1, & \text{if } u = a_{m+k}, u \in [0,1,2] \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

\bar{P} 是一筆樣本資料期間的每一日的調整後收盤價。即：

$$\bar{P} = [\bar{p}_1, \dots, \bar{p}_{TP}] = [p_m, \dots, p_{m+TP-1}] \quad (14)$$

另外，對於驗證資料以及測試資料的部分，一樣會依步驟(一)進行特徵計算，但不會使用調整後收盤價去計算目標交易行為，同時驗證資料與測試資料不會將資料處理成 TP 長度，而是以整段時間長度預測交易行為。

第二節 生成對抗網路模型建構

生成對抗網路模型共可拆分為生成網路與判別網路。生成模型透過樣本資料中的特徵 \mathbb{X} 預測交易行為 \tilde{Y} ，而判別模型則負責分辨交易行為是預測出的交易行為 \tilde{Y} 或是目標交易行為 Y 。本研究在生成對抗網路上分別使用了 GAN、LSGAN、WGAN 進行比較。

一、生成模型

生成模型根據特徵 \mathbb{X} ，預測交易行為 \tilde{Y} 。共有兩層的 LSTM，對於一個交易日，第一層 LSTM 透過過去多天期的特徵資訊得到新的隱藏層輸出，第二層 LSTM 則透過第一層的輸出預測交易日的交易行為，整個過程如圖 3 所示。GAN、LSGAN、WGAN 的生成模型結構相同，步驟如下。

(一)、 \mathbb{X} 中的 X 分別傳入第一層 LSTM。

對於一個樣本資料 \mathbb{X} 。首先， G^1 為生成器的第一層 LSTM，會依序使用 \mathbb{X} 當中的每一個 X 計算得到第一層的隱藏層輸出 h^1 ，即：

$$h_{s,k}^1 = G^1(x_k^s), s \in [1, \dots, S], k \in [1, \dots, TP] \quad (15)$$

其中 $h_{s,k}^1$ 表示 X_k 中所包含的第 s 個時間點特徵 x_k^s 傳入 G^1 後的隱藏層輸出。

(二)、第一層 LSTM 的最後一次輸出，傳入第二層的

LSTM，其輸出再進入兩層線性層。

將 X_k 最後一個隱藏層向量 $h_{S,k}^1$ 作為經 G^1 萃取後的狀態向量，故 \mathbb{X} 共有 TP 個狀態向量 $[h_{S,1}^1, \dots, h_{S,k}^1 \dots, h_{S,TP}^1]$ ，傳入第二層 LSTM 隱藏層 G^2 ，即：

$$h_k^2 = G^2(h_{S,k}^1), k \in [1, \dots, TP] \quad (16)$$

另外，本研究中的激活函數 φ 使用 Leaky ReLU，而第二層 LSTM 的輸出 h_k^2 會再經由 G^3 ， G^3 包含了兩層的線性層與歸一化指數函數 (softmax function) 得到三個輸出，且進入每一層線性層前，皆會先經過激活函數，即：

$$\tilde{y}_k = G^3(h_k^2) \quad (17)$$

其中 \tilde{y}_k 包含 $\tilde{y}_k^0, \tilde{y}_k^1, \tilde{y}_k^2$ ，分別代表不交易、買入、售出的機率，且：

$$\tilde{Y} = [\tilde{y}_1, \dots, \tilde{y}_k \dots, \tilde{y}_{TP}] \quad (18)$$

\tilde{Y} 代表在整個交易期間內所有交易日的預測結果。

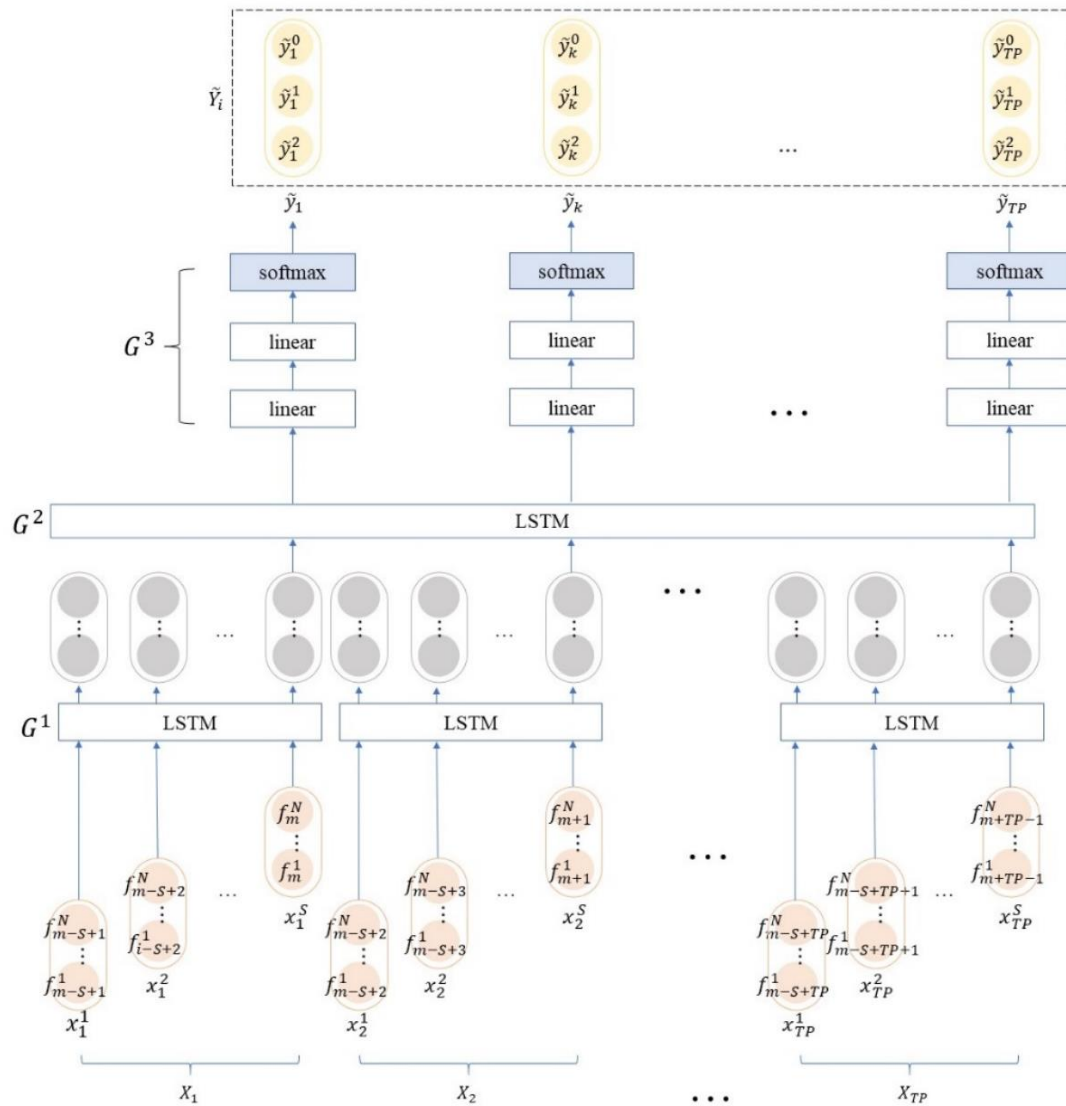


圖 2 生成模型結構

二、判別模型

判別模型用於判斷交易行為是否為源自 PLR 模型所產生的目標交易行為。判斷方式是將調整後收盤價 \bar{P} 與交易行為序列 \hat{Y} 使用的資料來自於樣本資料，而交易行為可能來自目標交易行為 Y 或是經生成模型的預測交易行為 \tilde{Y} 。在判別模型中有三層不同的 LSTM，行為與調整後收盤價會分別進入不同的 LSTM 層，將對應日期的隱藏層輸出合併後進入第三個 LSTM，整個過程如圖 4 所示，步驟如下。

(一)、將 \bar{p} 依日期先後順序傳入 LSTM。

一筆樣本資料的調整後收盤價 $\bar{P} = [\bar{p}_1, \dots, \bar{p}_{TP}]$ 共包含了 TP 個時間點。將調整後收盤價 \bar{P} 其依序經過一個名為 D^{price} 的 LSTM 隱藏層，即：

$$h_k^f = D^{price}(\bar{p}_k), k \in [1, \dots, TP] \quad (19)$$

其中 h_k^f 為第 k 個時間點下的隱藏層特徵。

(二)、對於預測交易行為 \tilde{Y}_i 或是目標行為 Y_i ，依序傳入

LSTM。

相對的，交易行為資料 $\hat{Y} = [\hat{y}_1, \dots, \hat{y}_{TP}]$ 按日期先後順序進入另外一個名為 D^a 的 LSTM 隱藏層，即：

$$h_k^a = D^a(\hat{y}_k), k \in [1, \dots, TP] \quad (20)$$

其中 D^a 是隱藏層，其輸入為交易行為 \hat{Y} ，且 \hat{Y} 包含了兩種情況，當使用的是目標交易行為時， \hat{Y} 為 Y ，反之當使用的是預測交易行為， \hat{Y} 為 \tilde{Y} 。

(三)、將兩個 LSTM 輸出合併依序進入第二層 LSTM，其

最後一次輸出再進入兩層線性層。

對於 D^{price} 以及 D^a 的所有輸出，依時間進行合併，即：

$$h_{tp}^1 = [h_k^f, h_k^a], k \in [1, \dots, TP] \quad (21)$$

並依序進入第二層名為 D^2 的 LSTM。即：

$$h_k^2 = D^2(h_k^1), k \in [1, \dots, TP] \quad (22)$$

第二層 LSTM 最後一次的輸出 h_{TP}^2 再經過 D^3 ， D^3 包含了兩層線性層以及在每一層線性層前的激活函數。輸出結果為 Z ，是單一個數字。即：

$$Z = D^3(h_k^2) \quad (23)$$

其中 Z 為一個值， Z 越大，表示判別模型將此交易行為序列 \hat{Y} 認定為目標交易行為的可能性就越大，以上為 LSGAN 的判別模型，而 GAN 與 WGAN 的判別模型會於 D^3 後，再計算 S 函數 *Sigmoid*，即：

$$Z = \text{Sigmoid}(D^3(h_k^2)) \quad (24)$$

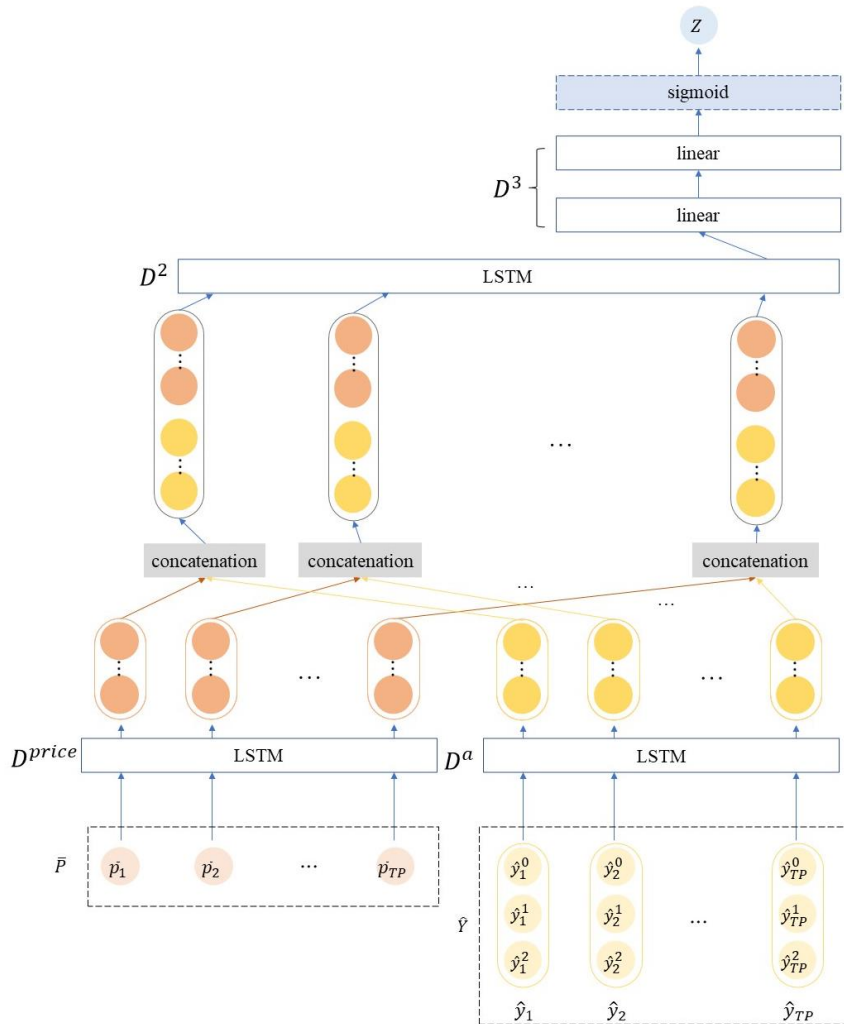


圖 3 判別模型結構

三、損失函數計算

(一)、生成對抗網路損失函數計算

1. GAN

GAN 模型的生成模型損失函數計算的第一步，是計算將預測交易行為輸入判別模型後得到的輸出 $Z_{fake} = D_{GAN}(\mathbb{X}, \tilde{Y})$ 。 Z_{fake} 會因 s 函數 *sigmoid* 輸出介於 0 和 1 之間，並將 1 作為目標計算 binary cross entropy 得到生成模型的損失函數。即：

$$L_G = \log(1 - Z_{fake}) \quad (25)$$

而判別模型的訓練將真實資料 Y 輸入判別模型得到輸出 $Z_{real} = D_{GAN}(\mathbb{X}, Y)$ ， Z_{real} 也會因為 *sigmoid* 會介於 0 和 1 之間。接著將 1 做為目標進行 binary cross entropy 計算得到損失值。也將預測交易行為 \tilde{Y} 輸入判別模型得到輸出 $Z_{fake} = D_{GAN}(\mathbb{X}, \tilde{Y})$ ，接著將 0 做為目標進行 binary cross entropy 計算得到損失值。整合兩個損失值的計算方式得到判別模型的損失函數 L_D 。

$$L_D = \log(Z_{fake}) + \log(1 - Z_{real}) \quad (26)$$

當真實資料經過判別函數後的輸出 Z_{real} 越靠近 1，得到的損失函數值越小，而當預測交易行為經過判別函數後的輸出 Z_{fake} 越靠近 0，得到的損失函數值也會越小。

2. LSGAN

LSGAN 模型的生成器損失函數計算的第一步，是使用預測的交易動作輸入判別器後得到的輸出 $Z_{fake} = D_{LSGAN}(\mathbb{X}, \tilde{Y})$ 。LSGAN 的判別模型並沒有 *sigmoid*，將 1 作為目標進行 MSE 計算得到生成模型的損失函數。即：

$$L_G = (Z_{fake} - 1)^2 \quad (27)$$

而判別模型的訓練，是使用真實資料 Y 進入判別模型的輸出 $Z_{real} = D_{LSGAN}(\mathbb{X}, Y)$ ，並將 1 做為目標進行 MSE 計算得到的結果以及預測交易行為 \tilde{Y} 進入判別模型的輸出 $Z_{fake} = D_{LSGAN}(\mathbb{X}, \tilde{Y})$ ，並將 0 做為目標進行 MSE 計算得到的結果，整合兩個結果得到判別模型的損失函數 L_D ：

$$L_D = \frac{1}{2}(Z_{fake} - 0)^2 + \frac{1}{2}(Z_{real} - 1)^2 \quad (28)$$

3. WGAN

在本研究中的 WGAN 模型僅部分使用了 Gulrajani et al. (2017) 所開發的 WGAN-GP，判別模型部分與 Gulrajani et al. (2017) 所開發的 WGAN-GP 相同的損失函數計算方法。損失函數不會計算 MSE 或是 binary cross entropy，而是將 Z_{real} 乘上 -1 並加上 Z_{fake} 作為損失函數。即：

$$L_D = -Z_{real} + Z_{fake} \quad (29)$$

另外會產生 0 到 1 的隨機數 ϵ ，並透過此隨機數產生目標交易行為 Y 與預測的交易行為 \tilde{Y} 混和的交易行為 Y_{mix} 。即：

$$Y_{mix} = \epsilon(Y) + (1 - \epsilon)(\tilde{Y}), 0 \leq \epsilon \leq 1 \quad (30)$$

並且嘗試計算混和交易行為 Y_{mix} 經過判別函數所產生的梯度 L2 範數與 1 的平方差。即：

$$(\|\nabla_{Y_{mix}} D(\mathbb{X}, Y_{mix})\|_2 - 1)^2 \quad (31)$$

將此懲罰項加入損失函數的一部份，故完整的損失函數為：

$$L_D = -Z_{real} + Z_{fake} + \lambda(\|\nabla_{Y_{mix}} D(\mathbb{X}, Y_{mix})\|_2 - 1)^2 \quad (32)$$

λ 為懲罰項的權重參數，本研究 λ 設為 0.1，而為了讓本研究的三個損失函數盡量一致，WGAN 生成模型的損失函數使用與 GAN 的生成模型損失函數相同。

4. GAN-S、LSGAN-S、與 WGAN-S

另外，本研究也嘗試同時使用生成對抗網路與監督式學習更新生成模型，期望能讓模型快速的收斂。故生成器的損失函數是透過生成對抗網路的損失函數以及監督式學習的損失函數組合而成。而判別模型的部分與單純使用生成對抗網路的模型相同。

GAN-S、LSGAN-S 與 WGAN-S 分別是本研究 GAN 模型、LSGAN 模型與 WGAN 模型與監督式學習複合的模型，在生成模型的部分除了原來基於判別模型計算其損失函數外，會再將目標交易行為作為監督式學習的正確解答，直接將目標行為 Y_i 與預測交易行為 \tilde{Y}_i 進行比對，並透過 crossentropy 計算損失函數：

$$L_{\text{Supervised}} = \sum_{k=1}^{TP} \sum_{a=0}^2 -y_k^a \log(\tilde{y}_k^a) \quad (33)$$

所以 GAN-S、WGAN-S 完整的生成模型損失函數為：

$$L_G = 0.5 \log(1 - Z_{\text{fake}}) + 0.5 L_{\text{Supervised}} \quad (34)$$

而 LSGAN-S 完整的生成模型損失函數為：

$$L_G = 0.5(Z_{\text{fake}} - 1)^2 + 0.5 L_{\text{Supervised}} \quad (35)$$

而另一方面，GAN-S 模型、LSGAN-S 模型與 WGAN-S 模型的判別模型損失函數則分別與本研究的 GAN 模型、LSGAN 模型與 WGAN 模型的判別模型損失函數相同。

第三節 評價方式

研究假設在買入時將所有現金轉成股票資產，在平倉售出時則將所有資產轉成現金，故不考慮加碼、減碼，也不考慮股票資產不得分割的問題。對於連續預測買入或連續預測賣出會以最初預測買入或平倉賣出的時間點為主。對於預測交易行為資料 $\tilde{Y} = [\tilde{y}_1, \dots, \tilde{y}_k, \dots, \tilde{y}_{TP}]$ ，先找出每一個 \tilde{y} 中的預測機率最大的交易行為，即：

$$\tilde{q}_k = \underset{u}{\operatorname{argmax}} y_k^u, u \in [0,1,2] \quad (36)$$

其中 \tilde{q}_k 為第 k 日預測機率最大的交易行為，且 $\tilde{Q} = [\tilde{q}_1, \dots, \tilde{q}_k, \dots, \tilde{q}_{TP}]$ 為交易期間所有 \tilde{q} 的序列。接著定義兩個 TP 長度的向量 \tilde{A} 與 E 。 $\tilde{A} = [\tilde{a}_1, \dots, \tilde{a}_k, \dots, \tilde{a}_{TP}]$ 為每日的實際交易行為， $E = [e_1, \dots, e_k, \dots, e_{TP}]$ 為每日交易前的持有狀態向量， $e_k = 0$ 為未持有股票的狀態， $e_k = 1$ 為持有股票的狀態。初始 $e_1 = 0$ 表示未持有狀態。而後續每一個 \tilde{a}_k 與 e_{k+1} 會根據 e_k 與 \tilde{q}_k 決定。若 $e_k = 0$ 為未持有股票狀態且 $\tilde{q}_k = 1$ 表示執行買入行為時 $\tilde{a}_k = 1$ ，且 $e_{k+1} = 1$ 表示隔日狀態為持有股票狀態；而若 $e_k = 1$ 為持有股票狀態且 $\tilde{q}_k = 2$ 表示執行平倉行為時 $\tilde{a}_k = 2$ ，且 $e_{k+1} = 0$ 表示隔日狀態為未持有股票狀態。除了以上兩種情況以外， $e_{k+1} = e_k$ 表示與前一日狀態相同， $\tilde{a}_k = 0$ 表示當日沒有執行任何交易行為，即：

$$e_{k+1} = \begin{cases} 1, & \text{if } e_k = 0 \text{ and } \tilde{q}_k = 1 \\ 0, & \text{if } e_k = 1 \text{ and } \tilde{q}_k = 2 \\ e_k, & \text{otherwise} \end{cases} \quad (37)$$

$$, k \in [1, \dots, TP - 1]$$

且：

$$a_k = \begin{cases} 1, & \text{if } e_k = 0 \text{ and } \tilde{q}_k = 1 \\ 2, & \text{if } e_k = 1 \text{ and } \tilde{q}_k = 2 \\ 0, & \text{otherwise} \end{cases} \quad (38)$$

$$, k \in [1, \dots, TP]$$

接著，從 \tilde{A} 得出所有的買入時間點序列 T_{buy} 與平倉時間點序列 T_{sell} ，即：

$$k \in T_{buy} \text{ if } \tilde{a}_k = 1, k \in [1, \dots, TP] \quad (39)$$

且：

$$k \in T_{sell} \text{ if } \tilde{a}_k = 2, k \in [1, \dots, TP] \quad (40)$$

而 T_{buy} 與 T_{sell} 為按時間排序的序列，而若 T_{buy} 比 T_{sell} 的長度多 1 表示最後一次買入的交易未平倉。實際的收益應將此未平倉交易以期末股票價格現值列入收益計算，但這樣的交易行為並不是模型所決定的買賣點，對模型的績效評比會有一定程度的落差，故在本研究中不考慮未平倉交易所帶來的損益結果。將 T_{buy} 的最後一個時間點剝

除，則在 TP 期間內發生的所有 N 個交易對的買入時間點為 $T_{buy} =$

$[t_{buy}^1, \dots, t_{buy}^n, \dots, t_{buy}^N]$ ，而賣出時間點則為 $T_{sell} = [t_{sell}^1, \dots, t_{sell}^n, \dots, t_{sell}^N]$ ，一買一賣構成一個交易對，並使用調整後收盤價計算報酬率。則第 n 個交易對的個別報酬率 r_n 則為：

$$r_n = \frac{\bar{p}_{t_{sell}^n} - 1}{\bar{p}_{t_{buy}^n}} \quad (41)$$

而累積的報酬率 r 即：

$$r = \left(\prod_{n=1}^N (1 + r_n) \right) - 1 \quad (42)$$

另外也評估模型持有股票期間的夏普比率 (Sharpe ratio)，公式為：

$$\text{夏普比率} = \frac{\text{持有股票區間之日報酬率平均} - \text{日化定存利率}}{\text{日超額報酬標準差}} \quad (43)$$

其中持有股票區間之日報酬率平均 r_e 為：

$$r_e = \frac{\sum_{n=1}^N \sum_{i=t_{buy}^n}^{t_{sell}^n-1} \frac{\bar{p}_{i+1} - \bar{p}_i}{\bar{p}_i}}{\text{持有股票總天數}} \quad (44)$$

日化定存利率 r_f 為：

$$r_f = (1 + \text{半年定存利率})^{1/125} - 1 \quad (45)$$

日超額報酬標準差 σ_{r_e} 為：

$$\sigma_{r_e} = \frac{\sqrt{\sum_{n=1}^N \sum_{i=t_{buy}^n}^{t_{sell}^n-1} \left(\frac{\bar{p}_{i+1} - \bar{p}_i}{\bar{p}_i} - r_f \right)^2}}{\text{持有股票總天數}} \quad (46)$$

也會比較每一個模型在不同股票上的勝率，公式為：

$$WPTC = \frac{\text{報酬大於零的交易對數}}{N} \quad (47)$$

其中 N 為模型在當支股票的交易次數。

第四章 實驗結果與分析

第一節 實驗數據

一、資料來源與處理

研究分析對象為台指期 (TX) 比重最高的前 20 大成分股，數據蒐集使用 Yahoo Finance 的 API。蒐集 2016 年至 2019 年 12 月的每日價量資訊。交易資訊包含了日期、當日開盤價、當日收盤價、當日最高價、當日最低價、當日交易量以及調整後收盤價。接著使用每日價量資訊計算 97 種指標，得到包含當日開盤價、當日收盤價、當日最高價、當日最低價、當日交易量共計 157 項特徵。特徵中的空值使用 0 補上。為了平衡不同特徵的尺度差異，將所有的特徵使用最小值最大值正規化 (Min-max normalization)，將所有訓練資料正規化到 0 與 1 之間，而驗證與測試資料依據訓練資料的縮放尺度同步調整。計算完成的訓練資料範圍為 2016 年 10 月 20 日至 2018 年 12 月 17 日共 500 個交易日資料，驗證資料為 2018 年 12 月 17 日至 2019 年 7 月 1 日共 125 個交易日資料，測試資料為 2019 年 7 月 1 日至 2019 年 12 月 31 日共 125 個交易日資料。若股票的部分交易日缺失，則資料選取範圍向前遞移取到對應的資料筆數。

而目標交易動作序列則是將訓練資料 2016 年 10 月 21 日至 2018 年 12 月 17 日的調整後收盤價透過 PLR 模型產生。

二、資料分析

研究所使用的 20 支股票，根據 yahoo 股市中基本面公司資料分類進行產業類別分類，結果如表 1，其中半導體業、金融保險業、塑膠業、通信網路產業這四個產業在 20 支股票中包含了兩家以上的公司，半導體產業有台積電、聯發科與日月光控股，金融保險產業有國泰金、兆豐金、富邦金、中信金與玉山金，塑膠產業有台塑、南亞與台化，而通信網路產業則有中華電與台灣大。

表 1 股票產業類別

代號	公司名稱	產業類別	代號	公司名稱	產業類別
2330	台積電	半導體	2886	兆豐金	金融保險
2454	聯發科	半導體	2881	富邦金	金融保險
2317	鴻海	其他電子	1216	統一	食品
2412	中華電	通信網路	1326	台化	塑膠
6505	台塑化	油電燃氣	2891	中信金	金融保險
2882	國泰金	金融保險	3045	台灣大	通信網路
2308	台達電	電子零組件	2207	和泰車	汽車
3008	大立光	光電	2884	玉山金	金融保險
1301	台塑	塑膠	3711	日月光投控	半導體
1303	南亞	塑膠	2382	廣達	電腦及週邊設備

在表 2 是每一支股票在訓練資料、驗證資料、與測試資料期間調整後收盤價的敘述性統計資料，調整後收盤價計算基於 2020 年 8 月 12 日，比較三個時間跨度的平均數可以看出在半導體產業的台積電、聯發科與通信網路產業的中華電與台灣大，從訓練資料、驗證資料到測試資料的價格是持續上漲的，半導體產業的日月光在驗證資料的平均數低於訓練資料，但測試資料平均數相比訓練資料與驗證資料皆來得高，而塑膠產業的三支股票驗證資料平均數均高於訓練資料，但測試資料時又下跌到低於驗證資料的平均數，金融產業的五支股票在三個時間跨度則有不同的漲跌趨勢。

表 2 調整後收盤價敘述性統計資料

股票	訓練資料				驗證資料				測試資料			
	平均數	標準差	最小值	最大值	平均數	標準差	最小值	最大值	平均數	標準差	最小值	最大值
2330	199.0	25.0	155.7	248.3	223.1	13.5	194.9	252.0	274.9	30.0	234.3	336.5
2454	245.0	40.0	182.7	333.4	262.5	27.8	207.1	301.6	368.8	45.3	296.6	452.2
2317	99.5	16.2	61.2	134.1	68.2	5.7	61.1	83.0	76.4	7.3	68.1	88.3
2412	93.7	4.7	84.5	102.2	101.2	2.3	98.1	105.5	106.5	2.2	102.6	109.9
6505	99.3	10.7	83.0	137.9	104.9	3.3	95.7	110.1	96.3	4.0	91.9	106.5
2882	44.0	3.5	32.8	49.5	40.2	1.7	36.6	43.6	39.3	1.0	37.3	40.9
2308	127.8	17.2	93.0	153.2	141.5	9.2	118.5	156.1	139.6	6.5	123.3	150.9
3008	4179.1	735.9	2842.1	5684.3	3900.4	509.5	2797.7	4670.8	4221.0	346.8	3656.3	4960.1
1301	83.5	9.7	68.4	106.0	95.8	4.4	85.9	103.7	92.6	3.4	85.8	103.7
1303	65.2	6.9	51.4	77.5	69.4	1.5	65.6	71.9	68.5	1.7	63.8	71.3
2886	19.73	1.66	16.26	23.33	23.89	1.63	21.38	27.05	27.74	0.91	26.16	29.28
2881	43.25	2.56	35.85	47.93	41.06	1.16	38.26	43.60	42.56	1.20	39.75	45.00
1216	57.05	8.78	43.77	74.60	69.93	3.78	63.10	77.31	72.87	2.35	69.07	77.59
1326	86.60	11.78	70.66	115.17	96.25	2.28	89.36	101.28	85.28	3.89	80.83	96.88
2891	17.21	1.70	13.81	20.89	18.75	0.40	17.93	19.84	20.10	0.73	19.00	21.42
3045	93.91	3.54	84.12	101.08	102.76	4.12	97.42	112.06	108.29	2.05	103.98	112.06
2207	294.42	39.70	192.91	355.12	364.21	89.30	235.03	502.61	505.22	79.93	403.31	693.77
2884	15.55	2.10	12.73	20.13	20.78	1.98	17.70	24.25	25.50	1.13	22.92	27.17
3711	66.92	5.49	52.80	82.48	59.18	5.02	51.49	70.68	71.61	6.18	60.01	82.96
2382	51.05	5.33	42.73	64.36	51.19	2.38	45.43	55.15	55.79	3.00	51.08	61.68

而表 3 顯示在 γ 分別為 0.01、0.05、0.1 與 0.5 下，20 支股票在 500 日的訓練資料中經過 PLR 模型所產生的三種交易行為次數。在不同的 γ 下，PLR 模型會產生不同數量的交易對，當 γ 越小，所產生的交易次數就越多，同時每筆交易持倉的時間就越短，在 $\gamma = 0.01$ 時，多數股票的交易次數皆大於 100 次，但當 $\gamma = 0.5$ 時交易次數皆不超過 20 次。

表 3 股票對應 γ 在 PLR 模型處理後的各項交易行為次數

γ	0.01			0.05			0.1			0.5		
股票	不行為	買	平倉	不行為	買	平倉	不行為	買	平倉	不行為	買	平倉
2330	270	115	115	312	94	94	382	59	59	482	9	9
2454	248	126	126	294	103	103	356	72	72	478	11	11
2317	290	105	105	354	73	73	418	41	41	494	3	3
2412	310	95	95	310	95	95	366	67	67	482	9	9
6505	248	126	126	268	116	116	312	94	94	482	9	9
2882	242	129	129	254	123	123	310	95	95	470	15	15
2308	264	118	118	302	99	99	380	60	60	482	9	9
3008	226	137	137	274	113	113	330	85	85	472	14	14
1301	266	117	117	312	94	94	380	60	60	486	7	7
1303	278	111	111	330	85	85	388	56	56	486	7	7
2886	260	120	120	304	98	98	376	62	62	486	7	7
2881	258	121	121	278	111	111	316	92	92	464	18	18
1216	240	130	130	320	90	90	386	57	57	490	5	5
1326	256	122	122	316	92	92	392	54	54	488	6	6
2891	268	116	116	304	98	98	374	63	63	484	8	8
3045	280	110	110	280	110	110	288	106	106	466	17	17
2207	220	140	140	292	104	104	370	65	65	490	5	5
2884	288	106	106	380	60	60	444	28	28	494	3	3
3711	282	109	109	306	97	97	342	79	79	466	17	17
2382	264	118	118	306	97	97	366	67	67	478	11	11

第二節 實驗設計

一、實驗模型

本研究整合監督式學習與各式生成對抗網路，比較了七個模型：

- (1) Hierarchical LSTM: 使用監督式學習方式訓練模型。
- (2) GAN：使用基於 Goodfellow et al. (2014) 所開發之 GAN 模型訓練生成模型與判別模型。Optimizer adam G softmax bcross D sigmoid bcross

- (3) GAN-S：使用 GAN 模型的訓練方式訓練判別模型，並且同時使用 GAN 模型與監督式學習模型訓練生成模型。Optimizer adam G softmax bcross crossentropy D sigmoid bcross
- (4) LSGAN：使用基於 Mao et al. (2017) 所開發之 LSGAN 模型訓練生成模型與判別模型。
- (5) LSGAN-S：使用 LSGAN 模型的訓練方式訓練判別模型，並且同時使用 LSGAN 模型與監督式學習模型訓練生成模型。Optimizer adam G softmax bcross crossentropy D sigmoid bcross
- (6) WGAN：使用基於 Gulrajani et al. (2017) 所開發之 WGAN-GP 模型訓練判別模型，並使用二元交叉熵 (binary crossentropy) 方法計算的損失函數訓練生成模型。
- (7) WGAN-S：使用 WGAN-GP 模型的訓練方式訓練判別模型，並使用二元交叉熵 (binary crossentropy) 方法計算的損失函數與監督式學習模型方法訓練生成模型。

所有模型的優化器使用 Adam。在模型訓練一輪後，會將所有訓練資料與所有驗證資料分別使用生成器預測所有時間點的動作，並計算整段長度的收益率。當訓練資料與 validation 資料的報酬率均大於零，且驗證資料的報酬率比暫存的最佳模型的報酬率來得大，則將現在的模型記為暫存的最佳模型。接著開始下一輪的訓練。在訓練結束後，分別記錄最佳模型在驗證資料與測試資料的收益率。對於相同模型不同參數，以驗證資料的報酬率做參數的篩選，在比較模型時，則以測試資料的報酬率做比較。

二、環境與參數設置

本研究所使用的程式是基於 python 語言，神經網路模型的搭建則是基於 pytorch 套件。參數設定如表 4。而生成模型與判別模型的隱藏層則使用相同的 Hidden size 大小。

表 4 模型使用參數

參數	參數值	參數意義
T^{train}	500	訓練資料天數
T^{val}	125	驗證資料天數
T^{test}	125	測試資料天數
S	2、10、20	每日的交易行為是基於多少天的交易資訊決定。
TP	50	一筆訓練資料需要預測多少天的交易動作。
γ	0.01、0.05、0.1、0.5	PLR 方法所需設定的閾值，數字越大，每一段的漲跌波動趨勢越長且交易次數越少。
隱藏層大小	200、400	生成模型與判別模型隱藏層的神經元數量。
學習率	10^{-3} 、 10^{-4} 、 10^{-5}	模型訓練的更新率。
epoch	4000	模型使用所有資料訓練的次數。

第三節 實驗結果

一、七個模型的績效表現

對於每一個模型在不同參數下的訓練結果，以驗證資料的報酬率作為依據，選擇出最好參數下訓練出的結果，並比較不同模型在測試資料時的報酬率，使用四捨五入取小數點後 4 位，另外根據台灣銀行在 2019 年 7 月 1 日的新臺幣存(放)款牌告利率，滿六個月定存的年利率為 0.795%，故定存半年、四捨五入到小數點後四位的實際報酬率為 0.0040，將其作為比較基準。結果如表 5 所示，可以看出 GAN 在中華電、台塑與和泰車是報酬率最佳的模型；GAN-S 在聯發科、台塑化、國泰金、大立光與日月光控股是報酬率最佳的模型；LSGAN 在鴻海、兆豐金與富邦金是報酬率最佳的模型；LSGAN-S 在統一與廣達是報酬率最佳的模型；WGAN 在台積電與廣達是報酬率最佳的模型；WGAN-S 在南亞、台化、中信金、玉山金是報酬率最佳的模型，而

Hierarchical LSTM 僅在台灣大獲得了最佳的報酬表現。另外，台達電、統一與台灣大在這七個模型中皆沒有正報酬。

表 5 測試資料報酬率

股票\模型	GAN	GAN-S	Hierarchical LSTM	LSGAN	LSGAN-S	WGAN	WGAN-S
2330	0.0150	0.0547	0.0385	0.0221	0.0858	0.2249	0.0700
2454	0.2038	0.3082	-0.0112	0.1143	0.2004	0.1049	0.0291
2317	0.1702	0.0029	0.0000	0.1935	0.0194	0.1468	0.0855
2412	0.0375	0.0047	0.0000	-0.0226	-0.0044	0.0233	0.0371
6505	0.0000	0.0676	-0.0851	0.0000	-0.0633	0.0000	0.0186
2882	-0.0070	0.0477	-0.0352	0.0192	-0.0140	0.0034	0.0230
2308	-0.0291	-0.1192	-0.0188	-0.0456	-0.0136	0.0000	-0.1143
3008	0.0000	0.1995	0.1319	0.1261	0.1128	0.1859	0.1914
1301	0.0567	-0.0737	-0.0241	-0.0195	-0.0840	-0.0468	0.0212
1303	0.0274	0.0027	0.0000	0.0204	0.0145	0.0278	0.0525
2886	-0.0100	0.0000	0.0936	0.1214	0.0353	0.1016	0.0853
2881	-0.0163	0.0000	-0.0844	0.0625	-0.0021	0.0518	-0.0471
1216	-0.0843	-0.0398	-0.0488	-0.0355	-0.0153	-0.0535	-0.0703
1326	-0.0199	-0.0873	-0.0127	-0.1065	-0.0057	0.0000	0.0711
2891	0.0493	0.0973	0.0719	0.0486	0.0240	0.0582	0.1082
3045	-0.0131	-0.0511	-0.0046	-0.0126	-0.0753	-0.0075	-0.0293
2207	0.3231	0.2455	0.0000	0.0000	0.0385	0.0000	0.2035
2884	0.1158	0.0502	0.1276	0.0940	0.0000	0.1320	0.1466
3711	0.1717	0.1995	-0.0010	0.0264	0.0000	0.0000	0.1685
2382	0.0931	0.0898	0.0656	0.1628	0.3107	0.0000	-0.0038

從表 5 中針對每一支股票從七個模型中找出測試資料表現最佳的模型，並統計出每個模型在多少支股票的報酬表現為最佳模型，結果呈現於表 6，可以看出在全數 20 支股票中，GAN-S 在五支股票中是報酬率最佳的模型，是在七個模型中計數最多的。而若對 20 支股票使用等額的方式分別訓練模型做投資組合，則投資組合的報酬率即為 20 支股票的平均報酬率，比較七個模型在 20 支股票使用等額方式做投資組合的報酬率，結果同樣呈現在表 6，其中 GAN 為七個模型中投資組合報酬率最高者，平均報酬率為 0.0542，而排名第二為 WGAN-S，平均報酬率為 0.0523，最低的是 Hierarchical LSTM 為 0.0102，七個模型皆優於定存報酬率。另外針對同時使用了監督式學習與生成對抗網路的模型進行分析，最佳報酬的計數的表現上，GAN-S 與

WGAN-S 在分別比 GAN 與 WGAN 來得更佳，但在就平均報酬率而言，僅有 WGAN-S 優於 WGAN。

表 6 報酬最佳模型計數與平均報酬率

統計項目\模型	GAN	GAN-S	Hierarchical LSTM	LSGAN	LSGAN-S	WGAN	WGAN-S
報酬最佳模型計數	3	5	1	3	2	2	4
平均報酬率	0.0542	0.0500	0.0102	0.0384	0.0282	0.0476	0.0523

從表 5 統計出每個模型在這 20 支股票的測試資料當中的獲得正報酬、負報酬、或無交易的次數，並計算正報酬數與負報酬數的比值，結果於表 7，WGAN-S 在 20 支股票中有 15 支股票的表現為正報酬，為所有模型中最多的，而 WGAN 有最大的正報酬數/(正報酬數+負報酬數)的比值。另外針對同時使用了監督式學習與生成對抗網路的模型進行分析，GAN-S 的正報酬數比 GAN 的正報酬數多了 2，正報酬數/(正報酬數+負報酬數)比值高了 0.09，WGAN-S 的正報酬數比 WGAN 的正報酬數高了 4，但正報酬數/(正報酬數+負報酬數)比值低了 0.03，但 LSGAN 的正報酬數比 LSGAN-S 的正報酬數高了 3，且負報酬數是 LSGAN 較低，故正報酬數/(正報酬數+負報酬數)比值是 LSGAN 較高。基於生成對抗網路的模型正報酬數皆比負報酬數多，相對於 Hierarchical LSTM 皆有更好的結果。另外，可以觀察到 WGAN 是唯一一個著不交易數大於負報酬數的模型。

表 7 模型於 20 支股票中正報酬、負報酬與無報酬數

統計項目\模型	GAN	GAN-S	Hierarchical LSTM	LSGAN	LSGAN-S	WGAN	WGAN-S
正報酬數	11	13	6	12	9	11	15
無交易數	2	2	4	2	2	6	0
負報酬數	7	5	10	6	9	3	5
正報酬數/(正報酬數+負報酬數)	0.61	0.72	0.38	0.67	0.50	0.78	0.75

表 8 為使用半年定存利率，年化利率為 0.795% 作為基礎，計算最好的參數訓練結果下，模型在測試資料持有投資標的區間的夏普比率 (Sharpe ratio)，表格中「-」表示整段訓練資料交易期間未有交易，故未具有持有投資標的區間，可以看出 GAN 在台塑與和泰車有著最高的夏普比率，GAN-S 在中華電、台塑化、國泰金與玉山金共四支股票中是夏普比率表現最佳的模型，LSGAN 在鴻海、兆豐金與日月光控股共三支股票中是夏普比率表現最佳的模型，LSGAN-S 在台達電、統一與廣達共三支股票中是夏普比率表現最佳的模型，WGAN 在聯發科、富邦金與台灣大共三支股票中是夏普比率表現最佳的模型，WGAN-S 則在台積電、大立光、南亞、台化與中信金五支股票中是夏普比率表現最佳的模型。

表 8 測試資料夏普比率

股票\模型	GAN	GAN-S	Hierarchical LSTM	LSGAN	LSGAN-S	WGAN	WGAN-S
2330	0.0409	0.0684	0.1028	0.1910	0.0969	0.2772	0.4263
2454	0.1432	0.1947	-0.0302	0.1442	0.1918	0.3292	0.0788
2317	0.1361	0.0128	-	0.2225	0.1755	0.1744	0.1301
2412	0.4468	0.4830	-	-0.1727	-1.0144	0.0896	0.0648
6505	-	0.0822	-0.0961	-	-0.0576	-	0.0268
2882	-0.3749	0.1851	-0.3790	0.0206	-0.0244	0.0036	0.0316
2308	-0.2659	-0.1155	-0.1945	-0.0343	-0.0106	-	-0.1742
3008	-	0.1975	0.0764	0.0598	0.0820	0.0763	0.2030
1301	0.0743	-0.1943	-0.1631	-0.0146	-0.0881	-0.1488	0.0583
1303	0.0642	0.0051	-	0.0282	0.0455	0.0646	0.0740
2886	-0.0269	-	0.1024	0.2688	0.1311	0.1320	0.1387
2881	-0.0615	-	-0.1883	0.1167	-0.5565	0.1366	-0.1574
1216	-0.0952	-0.0469	-0.0617	-0.3059	-0.0190	-0.0587	-0.0771
1326	-1.0785	-0.4177	-0.0296	-0.2103	-0.0086	-	0.1110
2891	0.0917	0.1436	0.1166	0.0965	0.0558	0.0776	0.3708
3045	-0.0420	-0.0931	-0.0226	-0.0271	-0.1587	-0.0127	-0.0431
2207	0.4728	0.1214	-	-	0.0378	-	0.1614
2884	0.2049	0.6938	0.4491	0.2751	-	0.1957	0.1464
3711	0.1479	0.2171	-0.0038	0.2755	-	-	0.1230
2382	0.1241	0.1090	0.1619	0.2597	0.3018	-	-0.0032

從表 8 中針對每一支股票從七個模型中找出測試資料夏普比率表現最佳的模型，並統計出每個模型在多少支股票的夏普比率表現為最佳模型，結果呈現於表 9，可以看出在全數 20 支股票中，WGAN-S 則在台積電、大立光、南亞、台化與中信金共五支股票中是夏普比率最佳的模型，是在七個模型中計數最多的。而將同模型在 20 支股票的夏普比率進行平均，WGAN 為 0.0955 為所有模型最高的。另外，Hierarchical LSTM 與 LSGAN-S 則是平均夏普比率為負值的模型。另外針對同時使用了監督式學習與生成對抗網路的模型進行分析，GAN-S 與 WGAN-S 的夏普比率最佳模型計數皆分別高於 GAN 與 WGAN，但平均夏普比率僅只有 GAN-S 大於 GAN。

表 9 夏普比率最佳模型計數與平均夏普比率

統計項目\模型	GAN	GAN-S	Hierarchical LSTM	LSGAN	LSGAN-S	WGAN	WGAN-S
夏普比率最佳	2	4	0	3	3	3	5

模型計數							
平均夏普比率	0.0001	0.0915	-0.0100	0.0663	-0.0455	0.0955	0.0845

另外，將每一個模型在不同股票上的勝率平均後，得到表 10。在表 10 考慮了兩種平均方式，第一種是若當模型在當支股票沒有交易行為則不列入計算，第二種則是若當模型在當支股票沒有交易行為則直接是為勝率為 0 計算。考慮第一中平均計算方式，平均勝率最高的為 WGAN，勝率有 0.7009，相對的最低的為 GAN，平均勝率為 0.4701，同時考慮 WGAN 的不交易數是所有模型中最高的，可以看出 WGAN 有最高勝率，並在難以獲利的股票上直接放棄嘗試交易的特性。而第二種計算方式則是 WGAN-S 模型以 0.5300 的平均勝率在這七個模型的表現中為最佳的模型，其次是 WGAN 模型，最差的則是 Hierarchical LSTM。

表 10 平均勝率

統計項目\模型	GAN	GAN-S	Hierarchical LSTM	LSGAN	LSGAN-S	WGAN	WGAN-S
平均勝率(不計算不交易)	0.4701	0.5325	0.5176	0.5691	0.4963	0.7485	0.5300
平均勝率(不交易視為勝率 0)	0.4231	0.4793	0.4141	0.5122	0.4467	0.5239	0.5300

透過報酬率、夏普比率與平均勝率的結果，可以看出不同模型在不同的評價方法下有不同的表現結果，但在多數評價方式下，基於生成對抗網路的模型表現皆優於監督式學習的模型，其中又以 GAN、GAN-S 與 WGAN-S 有良好的表現。

二、使用不同特徵訓練判別模型的表現

本研究中，也嘗試調整後收盤價 \bar{P} 改成特徵 X 輸入進判別模型，嘗試比較提供給判別模型更多資訊下，能否有效提升生成模型的訓練水準。分別訓練使用調整後收盤價作為判別器輸入的六個生成對抗網路相關的模型以及使用特徵作為判別模型輸入的六個生成對抗網路相關模型。並將每個模型在不同股票的報酬率進行平均。結果如表 11。不論是使用調整後收盤價 \bar{P} 或是使用特徵 X 訓練判別模型，所有的模型結果皆比監督式學習模型的平均報酬率 0.0102 來得更佳。而使用特徵 X 作為判別模型輸入的模型，僅在 GAN-S 與 LSGAN-S 有更好的結果，但在其他四個模型皆是使用調整後收

盤價作為判別模型輸入的模型有更好的結果，且在比較六個模型的平均報酬率時，是使用調整後收盤價作為判別模型輸入的模型更佳。

表 11 不同判別模型輸入的模型平均報酬率

判別模型輸入\模型	GAN	GAN-S	LSGAN	LSGAN-S	WGAN	WGAN-S	平均
\bar{P}	0.0542	0.0500	0.0469	0.0282	0.0495	0.0523	0.0454
X	0.0382	0.0630	0.0439	0.0424	0.0136	0.0152	0.0361

三、交易行為預測視覺化

為了瞭解實際的交易情形，將預測後的行為透過圖進行觀察。圖 4 與圖 5 是表示每一支股票表現最佳的模型的交易預測結果。每張圖包含了每一支股票透過價格繪製出的折線圖，圖所包含的時間範圍是預測資料所包含的時間範圍，可以看出台塑化、統一與台化是下跌的趨勢，台達電、台塑與南亞是下跌後回升，其他多數的股票則是逐步上升的趨勢。另外，因為模型可能會預測連續多天的買點與賣點，圖上僅表示出預測並實際發生交易的買點與賣點。從圖可以看出在每一支股票上有不同的最佳模型，且模型的表現也不盡相同，交易的頻率也有差異。在部份股票的表現上，模型有在趨勢的低點買入並在趨勢相對高點平倉。且在部分不是在上漲趨勢的股票如台塑化與台化也有正向報酬。而在穩定上漲的股票中，也可以看出有許多股票上的交易特性是長期持有的。以上結果可以說明模型有機會能從價格中找到隱含資訊並獲利。

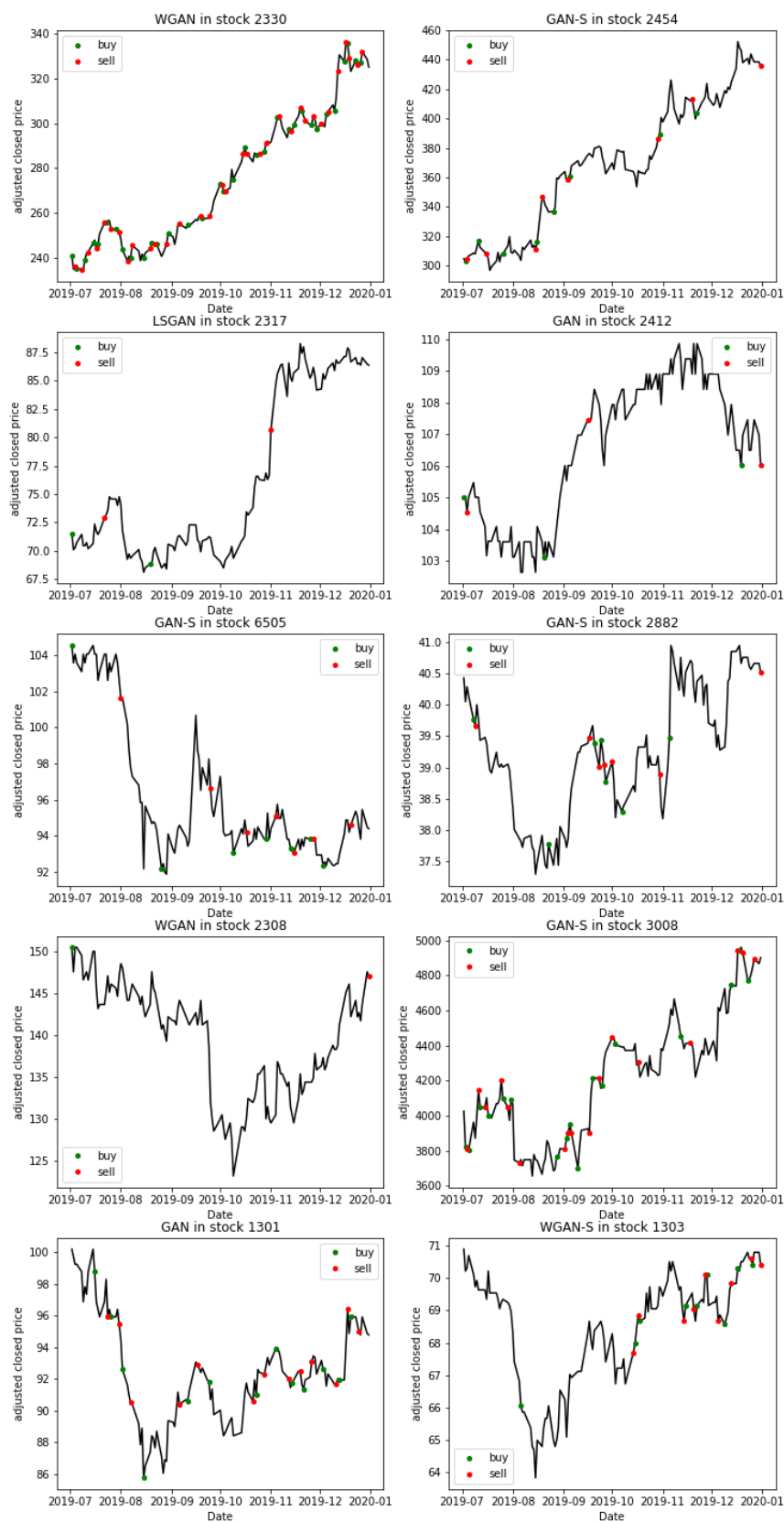


圖 4 在股票 2330、2454、2317、2412、6505、2882、2308、3008、1301、1303 上最佳模型的預測結果

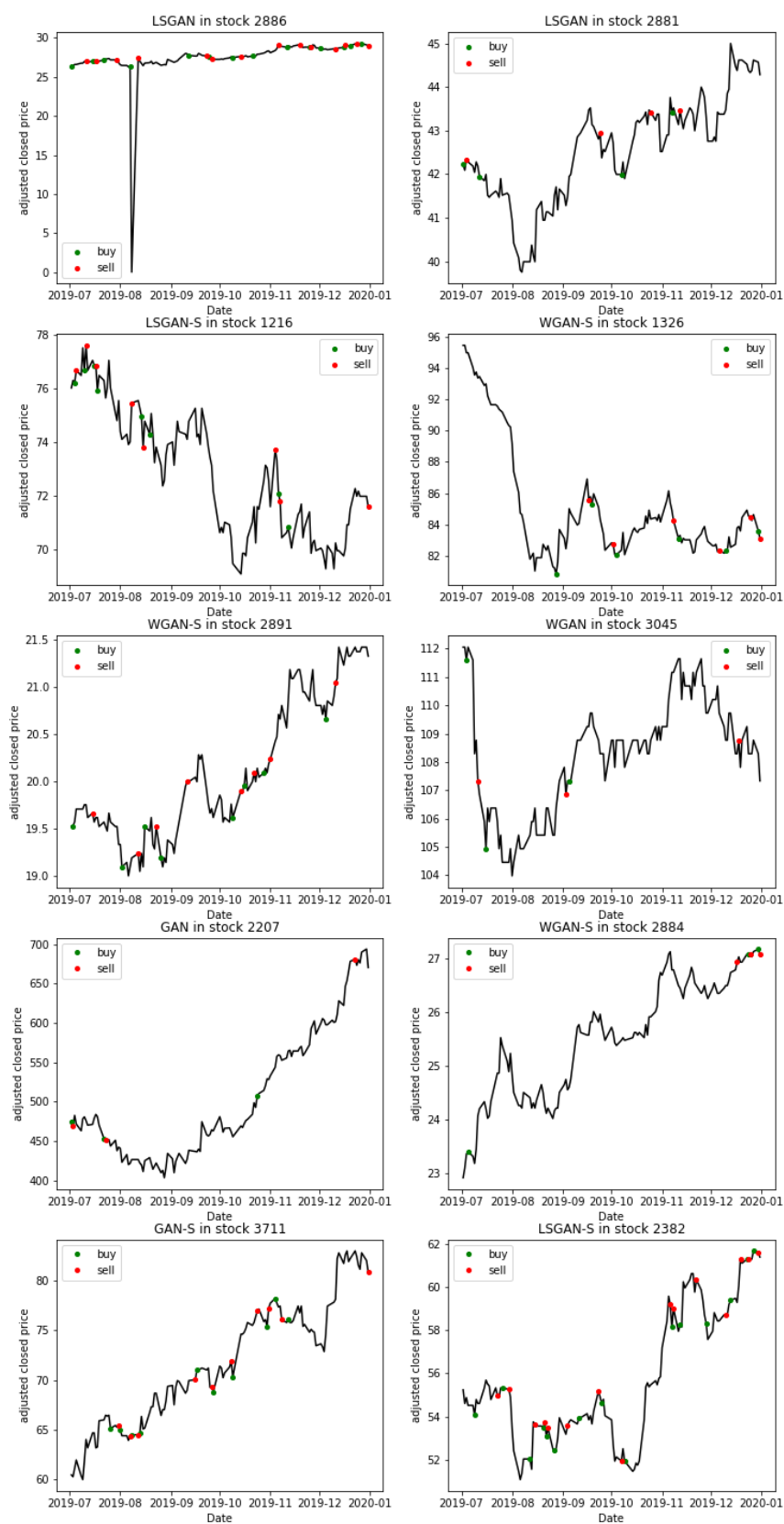


圖 5 在股票 2886、2881、1216、1326、2891、3045、2207、2884、3711、2382 上最佳模型的預測結果

第四節 管理意涵

本研究的目的是希望模型學習股票價格的趨勢，嘗試從價格中找出趨勢轉折點並做出正確的交易，從而預測出高報酬的交易行為，研究假設歷史價格包含了對於預測未來的隱含價格資訊。從實驗結果可以發現，基於生成對抗網路所生成的交易行為比起監督式學習模型有更好的報酬率。投資者可藉由此策略嘗試獲得超額報酬，進而達到市場的有效性。基金公司可基於此模型建立相應的產品，讓市場上的產品更加多元。觀察在不同的股票的收益率，可以看出在不同的產業中有不同的預測結果，可以用於判斷模型應適合用於哪一產業，或者推論那些產業基於價量指標的預測能力較佳，可以提供給投資人作為投資標的選擇。以本研究結果顯示，在金融及半導體產業的預測效果較佳。透過將 20 支股票上進行個別模型的訓練並形成投資組合，也能夠具有的正向收益率。

在生成對抗網路的研究，LSGAN 與 WGAN-GP 能有比 GAN 具有更好的收斂性，在圖片生成問題上這兩個模型都能有比 GAN 產生出更好的圖片結果。但在本研究中，基於 LSGAN 的模型，包含了 LSGAN 以及 LSGAN-S，雖然在比較勝率、夏普比率時有機會比其他模型表現更好，但這兩個模型都沒有較好的收益率，僅優於單純使用監督式學習的模型。並且平均收益率最好的反而依序是 GAN、WGAN-S 與 GAN-S。WGAN 模型雖然有著最佳的勝率，但在平均報酬率仍是六個模型的第四名。或許能說明不同的 GAN 模型有不同的適合使用情境。現階段多數的研究將生成對抗網路用於生成訓練資料的樣本，部分直接用於預測未來漲跌趨勢，而本研究嘗試使用生成對抗網路用於交易行為的產生。本研究是基於台灣股票進行模型的設計，藉由生成對抗網路，生成更靈活、高報酬率的交易行為。除了股票以外，也可以被應用到期貨、外匯、數位貨幣與其他投資商品。

第五章 結論

第一節 總結

使用生成對抗網路於交易行為預測，用生成模型預測交易行為，並用判別模型辨別行為是來自目標交易行為或是生成模型預測的交易行為，主要是希望在已知歷史交易行為下，透過模型解析歷史交易行為，在未來生成出近似的交易行為。而實驗結果也證實，基於生成對抗網路相關的模型在報酬率與正報酬率的表現上，有達到甚至超過監督式學習模型的水準。使用情境為試圖從歷史價量資料所切分出的趨勢波段與區域最高價最低價萃取出特徵，並更進一步建立出新的交易策略模型，如同本研究使用 PLR 所切分出的最高最低點作為目標交易行為。

第二節 研究限制

本研究的研究限制如以下幾點：

- (1) 本研究假設每次購買將所有資金轉換成股票，並以此方式計算報酬率，但在實務上股票的交易有固定的單位，最小的單位為一張 1000 股，一張的價值從數千到數十萬不等，在使用整股交易時難以將所有的資金轉換成股票，若嘗試使用零股交易，則還須面臨零股流動性較差、手續費較貴的問題。
- (2) 本研究是基於每日收盤後的價量資訊去預測隔日的交易行為，而其中價量資料是一個時間段的累積結果，無法將本研究套用至需即時反應的當沖交易。

第三節 未來展望

本研究除了試圖解決研究限制所面臨的問題外，對於未來展望主要有以下幾點：

- (1) 現階段 PLR 模型產生的目標交易行為，僅有在區域最高點與區域最低點定義為交易行為，其他時間點使用「不交易」作為第三個行為，但交易行

為僅需要買入價比賣出價低，就能獲利，現階段多數的不交易或無報酬的訓練結果可能就是因為多數目標交易行為是不交易而產生，未來可以優化 PLR 模型，改變交易行為的定義方式，或許有機會提升模型的收斂程度以及最後的報酬率。

- (2) 另外，本研究所使用的模型是基於是否與目標交易行為相似，而並非針對模型是否有正向報酬，若預測行為與目標行為相似卻不能有正向報酬也不是我們所期望的，可以加入 RL 的概念，在追求相似行為的同時也可以具備正向報酬率。
- (3) 本研究在訓練模型時，是將報酬率最佳的模型作為訓練是否好壞的依據，而未考量到模型的收斂情形，導致在不同參數下的表現結果有著極大的差異，這可能是在大量參數搜索下模型過度配適於驗證資料的原因之一。未來可能得將生成模型與訓練模型的收斂情形進行觀察，設計出從訓練過程中篩選最佳模型的方式。
- (4) 除了總結所提的使用情境外，可以嘗試以既有的交易策略做為學習對象生成近似而更具有彈性的交易策略，甚至基於多個交易策略學習以整合複數個交易策略。

參考文獻

1. Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214-223). PMLR.
2. Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, *abs/1803.01271*, <http://arxiv.org/abs/1803.01271>.
3. Bakar, N. A., & Rosbi, S. (2019). Monte Carlo Simulation for Data Volatility Analysis of Stock Prices in Islamic Finance for Malaysia Composite Index. *International Journal of Advanced Engineering Research and Science*, 6(3).
4. Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, 47, 552-567.
5. Bhandare, Y., Bharsawade, S., Nayyar, D., Phadtare, O., & Gore, D. (2020, June). SMART: Stock Market Analyst Rating Technique Using Naive Bayes Classifier. In *2020 International Conference for Emerging Technology (INCET)* (pp. 1-4). IEEE.
6. Cao, J., & Wang, J. (2020). Exploration of stock index change prediction model based on the combination of principal component analysis and artificial neural network. *Soft Computing*, 24(11), 7851-7860.
7. Chen, C. H., Su, X. Q., & Lin, J. B. (2016). The role of information uncertainty in moving-average technical analysis: A study of individual stock-option issuance in Taiwan. *Finance Research Letters*, 18, 263-272.
8. Chen, J. X. (2016). The evolution of computing: AlphaGo. *Computing in Science & Engineering*, 18(4), 4-7.
9. Efimov, D., Xu, D., Kong, L., Nefedov, A., & Anandakrishnan, A. (2020). Using generative adversarial networks to synthesize artificial financial datasets. *arXiv preprint arXiv:2002.02271*.
10. Faijareon, C., & Sornil, O. (2019, April). Evolving and combining technical indicators to generate trading strategies. In *Journal of Physics: Conference Series* (Vol. 1195, No. 1, p. 012010). IOP Publishing.

11. Goh, Y. K., & Lai, A. C. (2019, August). Financial network construction of a set of coupled stochastics differential equations using generative adversarial network. In *AIP Conference Proceedings* (Vol. 2138, No. 1, p. 030016). AIP Publishing LLC.
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).
13. Grilli, L., & Santoro, D. (2020). Generative adversarial network for market hourly discrimination.
14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in neural information processing systems* (pp. 5767-5777).
15. Jain, E. (2019). Technical analysis and National Stock Exchange of India: testing the RSI rule using CNX Nifty index. *Afro-Asian Journal of Finance and Accounting*, 9(4), 406-419.
16. Jiang, Q. (2019, December). Comparison of Black–Scholes Model and Monte-Carlo Simulation on Stock Price Modeling. In *2019 International Conference on Economic Management and Cultural Industry (ICEMCI 2019)* (pp. 135-137). Atlantis Press.
17. Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, 103, 25-37..
18. Koshiyama, A., Firoozye, N., & Treleaven, P. (2020). Generative adversarial networks for financial trading strategies fine-tuning and combination. *Quantitative Finance*, 1-17.
19. Lai, C. Y., Chen, R. C., & Caraka, R. E. (2019, July). Prediction Stock Price Based on Different Index Factors Using LSTM. In *2019 International Conference on Machine Learning and Cybernetics (ICMLC)* (pp. 1-6). IEEE.
20. Lee, M. C., Liao, J. S., Yeh, S. C., & Chang, J. W. (2020, January). Forecasting the Short-term Price Trend of Taiwan Stocks with Deep Neural Network. In *Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning* (pp. 296-299).
21. Lezmi, E., Roche, J., Roncalli, T., & Xu, J. (2020). Improving the Robustness of Trading Strategy Backtesting with Boltzmann Machines and Generative Adversarial Networks. *Available at SSRN 3645473*.

22. Liang, Z., Chen, H., Zhu, J., Jiang, K., & Li, Y. (2018). Adversarial deep reinforcement learning in portfolio management. *arXiv preprint arXiv:1808.09940*.
23. Li, J., Wang, X., Lin, Y., Sinha, A., & Wellman, M. (2020, April). Generating Realistic Stock Market Order Streams. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 01, pp. 727-734).
24. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2016). Continuous control with deep reinforcement learning. International Conference on Learning Representations (Poster).
25. Lu, T. H. (2014). The profitability of candlestick charting in the Taiwan stock market. *Pacific-Basin Finance Journal*, 26, 65-78.
26. Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2794-2802).
27. Mariani, G., Zhu, Y., Li, J., Scheidegger, F., Istrate, R., Bekas, C., & Malossi, A. C. I. (2019). PAGAN: Portfolio Analysis with Generative Adversarial Networks. *arXiv preprint arXiv:1909.10578*.
28. Marshall, B. R., Young, M. R., & Rose, L. C. (2006). Candlestick technical trading strategies: Can they create value for investors?. *Journal of Banking & Finance*, 30(8), 2303-2323.
29. Marti, G. (2020, May). CorrGAN: Sampling Realistic Financial Correlation Matrices Using Generative Adversarial Networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8459-8463). IEEE.
30. Ma, Y., Peng, H., & Cambria, E. (2018, February). Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM. In *Aaai* (pp. 5876-5883).
31. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
32. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013) Playing atari with deep reinforcement learning. In *NeurIPS Deep Learning Workshop*.

33. Nguyen, C. V., Chia-Han, C., & Nguyen, T. D. (2012). Testing the Weak-Form Efficient Market Hypothesis: Using Panel Data from the Emerging Taiwan Stock Market. *International Journal of Business and Social Science*, 3(18).
34. Park, H., Sim, M. K., & Choi, D. G. (2020). An intelligent financial portfolio trading strategy using deep Q-learning. *Expert Systems with Applications*, 113573.
35. Pramudya, R., & Ichsani, S. (2020). Efficiency of Technical Analysis for the Stock Trading. *International Journal of Finance & Banking Studies*, 9(1), 58-67
36. Raudys, A., & Pabarškaitė, Ž. (2018). Optimising the smoothness and accuracy of moving average for stock price data. *Technological and Economic Development of Economy*, 24(3), 984-1003.
37. Ravi, V., Pradeepkumar, D., & Deb, K. (2017). Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms. *Swarm and Evolutionary Computation*, 36, 136-149.
38. Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014, April). Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd international conference on world wide web* (pp. 373-374).
39. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Chen, Y. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676), 354-359.
40. Sobreiro, V. A., da Costa, T. R. C. C., Nazário, R. T. F., e Silva, J. L., Moreira, E. A., Lima Filho, M. C., ... & Zambrano, J. C. A. (2016). The profitability of moving average trading rules in BRICS and emerging stock markets. *The North American Journal of Economics and Finance*, 38, 86-101.
41. Sun, D., Yang, K., Shi, Z., & Chen, C. (2017, November). A New Mimicking Attack by LSGAN. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 441-447). IEEE.
42. Thammakesorn, S., & Sornil, O. (2019, April). Generating trading strategies based on candlestick chart pattern characteristics. In *Journal of Physics: Conference Series* (Vol. 1195, No. 1, p. 012008). IOP Publishing.
43. Wang, J. H., & Leu, J. Y. (1996, June). Stock market trend prediction using ARIMA-based neural networks. In *Proceedings of International Conference on Neural Networks (ICNN'96)* (Vol. 4, pp. 2160-2165). IEEE.

44. WANG, K., LI, Z. Y., SONG, T. Y., & ZHANG, Z. Y. (2019). Quantitative Strategy Based on Decision Tree and Bollinger Band. *DEStech Transactions on Computer Science and Engineering*, (ammso).
45. Wiese, M., Knobloch, R., Korn, R., & Kretschmer, P. (2020). Quant gans: Deep generation of financial time series. *Quantitative Finance*, 1-22.
46. Wu, J. L., & Chang, P. C. (2012). A trend-based segmentation method and the support vector regression for financial time series forecasting. *Mathematical Problems in Engineering*, 2012.
47. Xiong, Z., Liu, X. Y., Zhong, S., Yang, H., & Walid, A. (2018). Practical deep reinforcement learning approach for stock trading. *Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy*. Thirty-second Conference on Neural Information Processing Systems.
48. Yoon, J., Jarrett, D., & van der Schaar, M. (2019). Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems* (pp. 5508-5518).
49. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 5907-5915).
50. Zhang, K., Zhong, G., Dong, J., Wang, S., & Wang, Y. (2019). Stock market prediction based on generative adversarial network. *Procedia computer science*, 147, 400-406.
51. Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68-75.
52. Zhou, X., Pan, Z., Hu, G., Tang, S., & Zhao, C. (2018). Stock market prediction on high-frequency data using generative adversarial nets. *Mathematical Problems in Engineering*, 2018.
53. Zhou, Z., Gao, M., Liu, Q., & Xiao, H. (2020). Forecasting stock price movements with multiple data sources: Evidence from stock market in China. *Physica A: Statistical Mechanics and its Applications*, 542, 123389.