

國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

訊號分解對於長短期記憶預測股價準確率之影響

The Impact of Signal Decomposition on LSTM for Stock
Price Prediction

林群歲

Qun-Wei Lin

指導教授: 呂育道 博士

Advisor: Yuh-Dauh Lyuu Ph.D.

中華民國 109 年 7 月

July, 2020

國立臺灣大學碩士學位論文
口試委員會審定書



訊號分解對於長短期記憶預測股價準確率之影響

The Impact of Signal Decomposition on LSTM for Stock
Price Prediction

本論文係林群崑君（學號 R07922164）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 109 年 7 月 28 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

呂育道

（指導教授）

張經瑛

陸裕豪

金國興

莊永裕

系主任





誌謝

本論文能夠如期完成，首先要感謝我的指導教授呂育道老師，兩年間的指導對學生具有非常大的啟發，並提供本論文悉心與嚴謹的校訂，使各個環節得以更臻完善。

才德兼備的呂老師一直是我心目中良師益友的典範，不僅在學術上耐心指導學生，令人如沐春風外；在生活上，嚴謹細緻、精益求精的生活態度同樣深深激勵著我。能夠加入呂老師的研究團隊，我覺得自己很幸運，在此謹向呂老師致以誠摯的謝意和崇高的敬意。

感謝我的家人在我攻讀研究所時期，總是無條件地給予支持，如果沒有家人成為我堅實而溫暖的後盾，這兩年一定無法如此順利地完成學業。此外，必須感謝實驗室的博士班學長：政良、冠倫，與碩士班的同窗們：謙仁、梓安、宇霖、王毅，從論文的選題，一直到最後定稿前的細節，總是不厭其煩地與我討論，給出許多有用的建議，並在我有任何不清楚、不瞭解的學問上，獲得你們的幫忙與意見。同時，也是你們讓我的研究所生活過得如此豐富多彩。

這兩年間獲得的不只是學問上的精進，更多的是學到了如何直面挑戰與磨練自己。一路走來要感謝的人太多，實難一一言謝。僅將這份完成碩士學業的喜悅獻給所有關心我、疼愛我的你們。沒有你們的陪伴，便無法擁有今日的成果，更無法在未知浩瀚的學涯中前行，謝謝你們。





摘要

金融市場的時間序列經常具有高雜訊、非穩態及非線性等特徵，因此提高預測準確率是非常具有挑戰性的任務，而這類型的資料與電機通訊領域中的訊號具有相同特性，因此可以將這些時間序列看成一種訊號。在通訊領域上經常使用小波分析和經驗模態分解等工具進行時頻域分析，將此類不平穩的時間序列拆解為不同頻率的分量以利進一步的分析。本研究嘗試將這些數學工具應用在台灣金融市場所形成的時間序列上，比較不同的分解方式對於長短期記憶類神經網路的股價預測結果有何影響。結果表明，輸入的訊號經過越多的分解次數並不代表能使模型獲得更好的表現。同時，研究中發現深度模型的超參數對於預測的準確率具有決定性之影響，故採用布穀鳥搜尋演算法來調整遞迴類神經網路的超參數，以利進一步的實驗。最終，在本論文選定的幾種分解方式中，對於預測台股大盤隔日漲跌趨勢的深度學習模型來說，其資料預處理以 Coiflet-3 作為基底之小波轉換表現最佳，其精確度可達 57.6%。

關鍵字：布穀鳥搜尋演算法、長短期記憶、遞迴類神經網路、股價預測、小波轉換





Abstract

Most financial time series are inherently noisy, non-stationary, and non-linearity by default. Since these time series have the same characteristics as signals in electrical engineering, we can also regard them as signals. In this study, we try to combine a long short term memory neural network with the frequency-domain analysis techniques. Furthermore, the performance of a model highly depends on the hyper-parameters selection. For this reason, a meta-heuristic algorithm named Cuckoo Search is utilized to identify the suitable hyper-parameters of the model. Our goal is to compare different decomposition methods on the prediction accuracy of the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX). The results show that: (1) The higher decomposition level doesn't improve the model performance. (2) For data preprocessing, Coiflet-3 outperforms all compared wavelet basis functions. It predicts the direction of movement in TAIEX for the next day with 57.6% accuracy.

Keywords: Cuckoo Search algorithm, Long short-term memory, Recurrent neural network, Stock price prediction, Wavelet transform

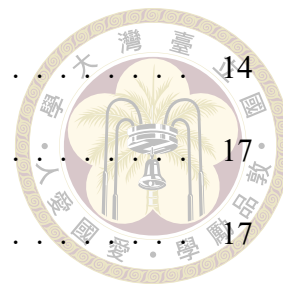




目錄

	Page
口試委員審定書	i
誌謝	iii
摘要	v
Abstract	vii
目錄	ix
圖目錄	xi
表目錄	xiii
第一章 緒論	1
1.1 簡介	1
1.2 論文架構	5
第二章 背景	7
2.1 股價預測相關文獻	7
2.1.1 效率市場假說	7
2.1.2 以深度學習預測股價漲跌	8
2.2 訊號處理相關文獻	9
2.2.1 小波轉換	9
2.2.2 穩態小波轉換	13

2.2.3	經驗模態分解	14
2.3	類神經網路相關文獻	17
2.3.1	遞迴類神經網路	17
2.3.2	長短期記憶類神經網路	18
2.3.3	雙向遞迴類神經網路	19
2.4	布穀鳥搜尋演算法相關文獻	20
2.4.1	布穀鳥搜尋演算法	20
2.4.2	自動調整遞迴類神經網路的超參數	23
第三章	實驗方法	25
3.1	模型概述	25
3.2	資料來源	29
3.3	驗證資料處理	30
第四章	實驗結果	31
4.1	實驗一	31
4.2	實驗二	33
第五章	結論與建議	35
5.1	結論	35
5.2	未來展望	36
	參考文獻	37





圖目錄

2.1	離散小波轉換 (階層分解)	12
2.2	濾波器的上取樣	14
2.3	穩態小波轉換 (階層分解)	14
2.4	經驗模態分解流程圖	16
2.5	簡單 RNN 步驟展開圖	17
2.6	LSTM 結構圖	19
2.7	BRNN 步驟展開圖	20
3.1	結合小波轉換之 LSTM 展開圖	26
3.2	利用窗函數截斷訊號示意圖	26
3.3	整體實驗架構流程圖	28
3.4	滾動視窗驗證	30





表目錄

3.1	布穀鳥演算法之參數	27
3.2	超參數搜尋範圍	29
4.1	超參數	31
4.2	準確率 (不分解)	31
4.3	準確率 (分解 1 次)	32
4.4	準確率 (分解 2 次)	32
4.5	準確率 (分解 3 次)	32
4.6	準確率 (分解 4 次)	32
4.7	優化後之超參數	33
4.8	優化後超參數下之準確率	33





第一章 緒論

1.1 簡介

1965 年 Fama 提出效率市場理論 (efficient market theory) [10]，是一個目前正在金融界被廣為所知的概念之一，Fama 認為在一個具有效率的市場中，所有消息均已充分的反映在股價上，沒有任何投資者可以利用公開的消息找出能長時間擊敗市場的策略，但有不少學者認為短期的股價仍是可被預測的 [22]。例如利用購入過去一段時間內股價呈現上漲趨勢的標的，並賣出那些表現不佳的標的，即採用所謂的動能策略 (momentum strategy) 進行交易，便有機會在未來 3 到 12 個月內賺得超額報酬 [19]。這種透過報酬本身具有延續性來獲取超額報酬的作法，開始讓不少人對效率市場假說抱持著懷疑的態度，因此陸續有學者提出各種不同的解釋；而當前的股價是否真如 Fama 所說，已經完全反映了市場上的所有歷史資訊，仍有待觀察。因此，本文研究著重於以技術面分析的方式建立模型，將歷史股價相關之資訊用於預測股票價格未來的走向。

在通訊系統、訊號處理等相關技術領域中，為了觀察在一段訊號中較為深層的固有特性，時常需要對原始訊號進行加工處理，因為重要資訊往往被隱藏在訊號裡不同的頻率成分中。人們經常使用傅立葉轉換 (Fourier transform)、小波轉換 (wavelet transform) 等數學工具將訊號進行加工處理。一般來說，訊號是指包含某些資訊在內隨時間或者空間變化的物理量，人們可以用不同的方法從中萃取出


所需要的資訊，金融的時間序列同樣也包含了若干的物理資訊在內，和通訊領域中分析的訊號具有相同的特性，當然也可以視為一種訊號 [13]。但股票市場以其極端的非線性和隨機性而聞名，所以與股票價格有關的訊息存在很大的雜訊和不確定性，仍是時間序列預測任務中較為艱澀困難的任務之一。



不夠平穩的訊號常常影響預測模型的精確度，目前時間序列的相關領域中，長短期記憶 (long short-term memory, LSTM) 和門控循環單元 (gate recurrent unit, GRU) 是遞迴類神經網路 (recurrent neural networks, RNN) 中經常被使用的工具，並且成為了各領先系統中的一部分。經典的 LSTM 和 GRU 雖可以捕捉訊號時域中的長期依賴關係，但它並沒有明確對頻域中的型態 (pattern) 進行建模，也因為後者對於時間序列的追蹤和預測具有重大意義，因此不可輕易忽視。RNN 結構通常只考慮時域上的依賴關係，應用於股市交易之中經常造成中短期投資策略不佳等問題，亦有研究針對此一問題將 LSTM 的結構進行變體 [15]，希望讓模型學習到一系列不同狀態頻率的組合。而本文實驗嘗試將訊號處理常用的數學工具應用在股票價格所形成的時間序列上，將原始訊號分解成不同的頻率分量作為輸入，使深度學習模型更容易捕捉到有利於判斷未來趨勢的特徵，進而獲得更好的預測結果。

標準的傅立葉轉換雖然可以將訊號從時域轉換到由正弦函數與餘弦函數作為正交基底的頻域中作分析，但也失去了訊號在不同時間的資訊，用來分析一個經常隨時間發生不定變化的訊號尚嫌不足，並不能知道該訊號所包含的頻率成份發生在哪一個時間點，為此必須改採同時分析時域和頻域的手法，其中最常被採用的工具如：短時距傅立葉轉換 (short-time Fourier transform, STFT)、小波轉換和雙線性時頻分析等。


利用 STFT 的方式又稱為時頻譜分析 (spectrogram)，即將窗函數 (window



function) 應用於傅立葉轉換之中。然而窗函數在時域上的切割越細，時間解析度 (temporal resolution) 越高，但對於低頻的部分，過細的切割會造成一整段低頻訊號被切成零碎的片段，其所帶來的訊息量並不足以完整理解整段低頻訊號，因此頻率解析度 (frequency resolution) 會隨時間解析度提高而降低；反之，使用越長的時間間隔，頻率解析度越高，時間解析度越低。與之相比，小波轉換以對數 (logarithmic) 而非線性 (linear) 的方式，能夠自適應調節不同的窗口大小，根據不同高低的頻率動態地調整不同的窗口長度；與 STFT 相比，小波轉換不但克服了傅立葉轉換的不足之處，更適用於分析非穩態 (non-stationary) 和非週期性 (non-periodic) 之資料 [7]。

使用一般的離散小波轉換會經過縮減取樣 (downsampling) 步驟，每經過一階段濾波，會將原先金融市場採樣而得的離散訊號其序列長度縮減為一半，使子序列中資料點的時間解析度下降。為了彌補此一缺點，本論文改為使用對濾波器進行上取樣 (upsampling) 的穩態小波轉換 (stationary wavelet transform, SWT) [25]，並與另一種完全由資料驅動 (data-driven) 的轉換方式：經驗模態分解 (empirical mode decomposition, EMD) 相比較，觀察不同的資料預處理方式，對預測結果有何不同的影響。過往的文獻中，有研究將 EMD 結合 LSTM 應用在匯率預測上 [37]，但經本文研究，該作者所提出的模型，其資料處理的方式與一般訓練模型的標準步驟迥異，有造成洩漏未來資料的問題存在，本文將該實驗步驟修正後，重新建立以 EMD 做為資料預處理的模型。此外，亦有研究將 SWT 與 LSTM 結合，用於個人家庭的能源消耗的預測上 [32]。

近年來，深度學習在各種訊號與時間序列的預測上皆有大量的應用，值得探討的是，深度學習模型的超參數往往在模型中起著至關重要的作用 [17]。除了模型中原本就有的大量超參數外，還有各種對於 RNN 架構的延伸，可以在某些標竿測試 (benchmark) 下獲得更好的結果：1997 年，Schuster 和 Paliwal 提出雙向



長短期記憶 [30]，藉由反轉時間序列，同時考慮上文和下文的資訊，在 TIMIT 資料庫上取得了良好的成果；2012 年由 Hinton 等人提出，在訓練神經網路時經常使用的放棄法，是一種神經網路模型平均方法，在當前訓練期間，透過隨機捨棄某些神經元來防止它們之間的共適應 (co-adapting)，可以有效緩解神經網路產生過度擬合 (overfitting) 的問題 [12]。基本上，這些延伸都可以視為 LSTM 架構中廣義的超參數 [28]。

然而超參數的調整往往得仰賴研究人員的直覺，必須藉由背景知識進行試誤手動選擇，或是利用窮舉法才能決定適合的超參數 [5]。近二十年來，有多種自動選擇機器學習超參數的方法被提出，其中包括眾多研究利用基因演算法 (genetic algorithm, GA) 來求解機器學習模型的超參數 [36]，也有將梯度搜尋演算法 (gradient-based optimization) [3]、粒子群優化演算法 (particle swarm optimization, PSO) [23]、差分進化 (differential evolution, DE) [26] 和模擬退火法 (simulated annealing, SA) [1] 等最佳化方法應用於超參數選擇的演算法。2012 年，Bergstra 和 Bengio 的研究表明，簡單的隨機搜尋方法相較於窮舉法有更好的表現，尤其是在具有低本質維度的高維度問題下會更加明顯 [4]。

經本實驗發現，在手動設定的超參數組合中，不同的組合會使不同的訊號分解方式具有截然不同的預測表現，因此確實需要對超參數優化找出最佳組合，才能比較何種分解方式對於 LSTM 預測股價漲跌較具優勢。本研究採用的隨機搜尋方法為布穀鳥搜尋 (cuckoo search, CS)，為一種自然啟發式演算法 (nature-inspired algorithm) [34]。布穀鳥搜尋演算法為一種參數少、操作簡單且易實作的搜尋演算法；同時，布穀鳥搜尋演算法能夠使用切換參數保持局部和全域隨機漫步之間的平衡，既可以在一定範圍內透過 Lévy flight 來加速探索最佳解，也可以透過棄巢來保持解的族群 (population) 多樣性。布穀鳥搜尋演算法其鮮明的特色讓它在工程最佳化問題上的表現不俗 [33, 35]，並已有相關研究利用改進的布穀鳥搜尋演算

法提出一種自動調整 LSTM 超參數的策略，來有效地求解在超參數選擇上之最佳化問題 [31]。



綜上所述，本文主要研究過程如下：

1. 運用 SWT 和 EMD 將歷史股價資料進行預處理，分解成不同頻率之子序列。
2. 將分解完的資料用以訓練 LSTM，並利用布穀鳥搜尋演算法搜尋其最佳參數組合。
3. 比較各種不同分解方式所建立之模型其預測股票價格漲跌的精確程度。

1.2 論文架構

本論文第一章提及研究動機、建立預測股價漲跌模型時需要克服的問題和相關實驗方法之概述。第二章闡述前人應用於擬合股價的深度學習模型及簡單回顧實驗中使用到的資料前處理、預測模型相關文獻。第三章詳細敘述本論文實驗方法及整體架構，並說明實驗資料的處理方式。第四章探討及呈現各實驗結果。最後一章則總結本論文內容與未來研究展望。





第二章 背景

2.1 股價預測相關文獻

2.1.1 效率市場假說


一般而言，使用統計或各種機器學習的方法來預測股價，需克服 1965 年 Fama 提出的效率市場假說 (efficient-market hypothesis, EMH) [10]，提出者根據主要的三項假設：參與市場的投資人皆理性、市場上的消息即時公開且不須負擔額外成本取得、沒有任何市場參與者的力量足以單獨撼動股票價格的情況下，認為投資人若身處於一個效率市場中，皆無法藉由過去的資訊預測未來的市場進而獲利，其中效率市場又可以詳細分述如下：

1. 弱勢效率 (weak form efficiency)：

股票的歷史價格所包含的訊息，已經被完整的包含在目前股價之中，所有投資者都無法藉由分析股票的歷史價格，作出交易策略來獲得超額報酬。倡導弱勢效率理論的人認為，如果使用基本面分析，可以確定被低估和被高估的股票，投資者可以研究公司的財務報表，以增加獲得超額報酬的機會。

2. 半強勢效率 (semi-strong form efficiency)：

半強勢效率市場的效率程度高於弱勢效率市場，當市場上出現各種公開情



報時，目前股票價格會立即且充分的反映出這些消息，所有投資者都無法利用過去這些免費公開於眾的資訊作出交易策略來獲得超額報酬。所以，無論是基本面或是技術分析皆無法使投資者獲取超額報酬。同時，半強勢效率也被認為是 EMH 最實用的假設，支持該理論的人認為，只有大眾無法輕易獲得的消息才能幫助投資者打敗市場獲得收益。

3. 強勢效率 (strong form efficiency) :

不僅已公開之情報都被反映至當前股價，所有內線消息帶來的價格波動亦被同時反映。在強式效率市場上，所有投資者都掌握了所有的消息，沒有任何情報可以為投資獲利帶來好處，此種情況亦是 EMH 中最嚴格的假設。

本論文使用的時間序列資料皆由台灣證交所每日揭露的實際數字，包括每日股價及成交量，即一般投資人也能輕鬆取得全部的資料，進行技術面分析。由於台灣證券交易所將所有操作資訊透明化並提供公平的訊息接收管道，因此 EMH 中的弱勢效率與本論文研究使用的深度學習預測方法為對立之論述。在一個弱勢效率市場中，所有的技術面操作都沒有辦法帶來任何好處，市場中每個投機者取得的超額利潤皆為純粹隨機的現象，但仍有不少研究認為短期內過去的資料對於股價預測具有一定的意義。舉例而言，Lo 和 MacKinlay 認為股價並不遵從隨機漫步 [22]，而在 Jadadeesh 和 Titman 的研究中，發現股票之報酬率在隨後的 3 至 12 個月會持續其表現，代表過去的資訊對於未來股市走向有一定的價值，可以作為選股之依據，形成可獲利的策略 [19]。

2.1.2 以深度學習預測股價漲跌

對於預測金融時間序列，訓練集 (training set) 的大小選擇非常重要，因為其高度非穩態且高雜訊的特性，容易在訓練集的規模過大或過小時，學習到過多與

驗證集無關或訓練集中隨機出現之特徵，因此如何選擇適當的訓練資料規模也是極其重要的問題之一 [13]。



另外，觀察以往深度學習模型預測股票價格的文獻中，對於預測漲跌方向性的分類問題著墨較少，通常是以迴歸問題（regression analysis）為大宗。2011年，Kara 利用各類技術指標做為學習資料，比較了支撐向量機（support vector machine, SVM）和類神經網路（artificial neural network, ANN）對漲跌預測的準確率 [20]。2015年，Ding 等人則從新聞文章中擷取特徵，並利用事件向量（event embedding）作為深度學習模型的輸入，對 S&P 500 指數進行漲跌預測 [8]，據作者所述該模型擊敗了當時所有最先進的預測方法。

2.2 訊號處理相關文獻

2.2.1 小波轉換

頻譜分析法是訊號處理中經常被使用到的技術，為了使計算更加有效率，快速傅利葉轉換（fast Fourier transform, FFT）在 1965 年一躍成為訊號處理最主要的分析方法之一，但利用傳統傅立葉轉換所得到的頻譜，會完全喪失時域上的相關訊息，亦即頻譜只能看出包含哪些頻率的弦波，而頻譜本身的分佈不會隨著時間而改變，無從得知每個頻率所發生的時間，致使發展其他的訊號處理分析工具有其必要。

小波分析的起源大約在 1930 年代左右，但實際上相似的理論在不同的領域一直都有學者提出，直到 1984 年才由 Morlet 與 Grossmann 將其正式命名為小波（wavelet），現代常見的落地應用包括邊緣偵測、不連續點偵測、濾波與壓縮等，其主要精神在於透過一個能量快速衰減或集中於一小頻帶之母小波（mother

wavelet)，使用其振盪波形所產生的基底函數 (basis function) 來重構一個訊號，在不同的時域解析度上作描述與分析。將母小波經過縮放 (dilation) 和平移 (translation) 可以決定不同的基底函數，這些基底函數可以根據需求製造出具有持續時間短、高頻率，或者持續時間長、低頻率的特性，以使用於重新表達一個函數。此種方法不但保留了傅氏理論的優點，還為訊號分析提供了一個時頻解析度可調並富有彈性的機制。

小波分析又可以粗略地分為連續小波轉換 (continuous wavelet transform, CWT) 和離散小波轉換 (discrete wavelet transform, DWT)。兩者間最大的不同，在於 CWT 利用母小波了所有可能的縮放和平移組合進行操作，而 DWT 則為了降低 CWT 計算上的複雜度，只採用了特定組合的子集 [6]。在金融時間序列的預測任務中，通常將小波轉換當成資料的前處理，用以擷取特徵和濾除雜訊，之後再配合統計或機器學習的方法建立預測模型，如 [27] 便嘗試以 Haar、Daubechies-3、Coiflet-3 和 Symlet-3 等不同的母小波分析股價，再透過 LSTM 搭配注意力機制 (attention) 建立股價預測模型。

考量勒貝格空間 (L^p spaces) 之子空間 $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ 中的一函數 $\psi(t)$ ，此函數具有以下限制：

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (2.1)$$

$$\|\psi(t)\|^2 = \int_{-\infty}^{\infty} \psi(t) \psi^*(t) dt = 1 \quad (2.2)$$

稱此函數為母小波，根據擴張和平移的狀態，可以將其餘函數表示為：

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \quad a \in \mathbb{R}, b \in \mathbb{R}^+ \quad (2.3)$$

式 (2.3) 又稱為子小波 (child wavelet)，其中 a 代表尺度參數， b 為平移參數。 a

值的大小可使母波產生擴張或壓縮的效果，母波之擴張程度與 a 的大小成正比；而 b 的作用則為調節母波沿時間軸移動時之步距，當母波較寬時，可適度選取較大的 b 值，以避免位移重疊。反之，若母波較窄時，應選擇較小的 b 值使位移縮小。將這些子小波與函數 $f(t)$ 的內積所得到的係數，則稱為小波係數：

$$w_{a,b} = \langle \psi_{a,b}, f(t) \rangle = \int_{-\infty}^{\infty} \psi_{a,b}(t) f(t) dt \quad (2.4)$$

上式 (2.4) 即為 CWT。

CWT 與 DWT 相比，最大的差別在於如何將尺度參數進行離散化 (discretization)，CWT 通常採用底數小於 2 的指數尺度，而 DWT 則一律使用 2 的冪作為尺度參數。實作上為了簡化運算時間，通常將 DWT 視為多解析度的分解，採用快速小波轉換 (fast wavelet transform) 的方式實作，可將一維的 DWT 使用階層式架構來表示。

利用小波函數為正交基底逼近一個離散函數 $f(t)$ ：

$$f(t) = \frac{1}{\sqrt{M}} \sum_n w_\phi(m_0, n) \phi_{m_0, n}(t) + \frac{1}{\sqrt{M}} \sum_{m=m_0}^{\infty} \sum_n w_\psi(m, n) \psi_{m, n}(t) \quad (2.5)$$

其中， $f(t)$ 、 $\phi_{m_0, n}(t)$ 和 $\psi_{m, n}(t)$ 皆為在 $[0, M - 1]$ 中有 M 個點的離散函數，且 $\{\phi_{m_0, n}(t)\}_{n \in \mathbb{Z}}$ 和 $\{\psi_{m, n}(t)\}_{(m, n) \in \mathbb{Z}^2, m > m_0}$ 彼此正交，可以很容易地得出小波係數：

$$w_\phi(m_0, n) = \frac{1}{\sqrt{M}} \sum_n f(t) \phi_{m_0, n}(t) \quad (2.6)$$

$$w_\psi(m, n) = \frac{1}{\sqrt{M}} \sum_n f(t) \psi_{m, n}(t), \quad m \geq m_0 \quad (2.7)$$

通常稱式 (2.6) 為近似係數 (approximation coefficients)，而式 (2.7) 為細節係數

(detailed coefficients)。其中， ϕ 為尺度函數 (scaling function)：

$$\phi_{m,n}(t) = \sqrt{2^m} \phi(2^m t - n) \quad (2.8)$$



而下一階層之近似係數與細節係數可由遞迴關係來表示：

$$w_\phi(m, n) = \sum_k h(k - 2n) w_\phi(m + 1, n) \quad (2.9)$$

$$w_\psi(m, n) = \sum_k g(k - 2n) w_\psi(m + 1, n)$$

式 (2.9) 中 h 代表低通濾波器 (low-pass filter) 之係數，用以將原始訊號分離出低頻部份， g 為高通濾波器 (high-pass filter) 之係數，用以分離出原始訊號之高頻部分，並在每一次分解後根據尼奎斯特取樣定理 (Nyquist-Shannon sampling theorem) 將資料長度減半，因此其每一階層輸出的時域解析度都會降低一半 [21]。

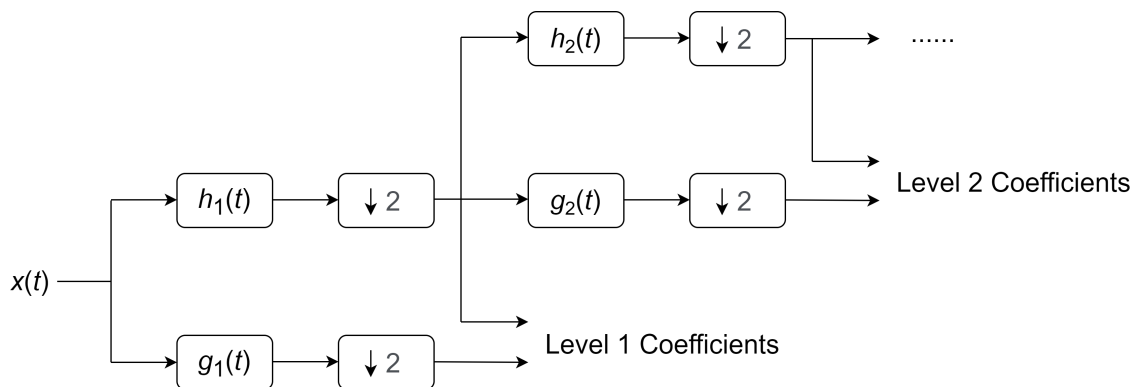


Figure 2.1: 離散小波轉換 (階層分解)

圖2.1為 DWT， $x(t)$ 為原始訊號， $h_i(t)$ 表示第 i 階層之低通濾波器的係數，用以濾除輸入訊號之高頻部份。 $g_i(t)$ 代表第 i 階層之高通濾波器的係數，功能與低通濾波器相反。而 $\downarrow 2$ 代表縮減取樣之處理，若以 $x(t)$ 作為輸入，則輸出 $y(n) = x(2n)$ 。



2.2.2 穩態小波轉換

穩態小波轉換 (stationary wavelet transform, SWT) 是 DWT 的一種變體，可以彌補 DWT 因縮減取樣而失去的平移不變性 (translation-invariant) [25]。其分解出之子訊號資料長度不變，因此在金融時間序列上能夠保留住每個資料點原本對應的時間點。SWT 操作上不同於 DWT 的部分，在於對濾波器進行上取樣 (upsampling)，取代 DWT 在經過高通濾波器和低通濾波器之後的縮減取樣。SWT 在很多領域中皆有不同的研究者提出相似的概念，但萬變不離其宗，只是因為在不同的時間被提出，而有很多相異的名字，如：à trous 演算法、冗餘小波轉換 (redundant wavelet transform) 與平移不變量小波轉換 (translation invariant wavelet transform) 等，其實並無二致。為了與下一小節所介紹之 EMD 互相比較，需要保留每個資料點對應之時間訊息，並讓兩者 LSTM 所經過的時間步數相等，本實驗採用 SWT 而非 DWT 做為訊號分解方法。

其中最常用的實作方法即 à trous 演算法，又稱為快速二元小波轉換 (fast dyadic wavelet transform)，該實作除了移除對訊號的縮減取樣以外，其餘部分皆與快速小波轉換非常相似，主要差別是在每個分解階段中，傳統的 DWT 會在式 (2.9) 中進行縮減取樣，意即每個階段的訊號分量皆減少為上一階段一半的長度，直到函數長度變為 1 為止；而 SWT 改由在每個尺度上執行濾波器捲積之前，先利用零填充 (zero padding) 將濾波器係數進行上取樣 [9]。

圖2.3為 SWT 階層分解之示意圖， $x(t)$ 為原始訊號， $g_j(t)$ 表示第 j 階層之低通濾波器的係數， $h_j(t)$ 代表第 j 階層之高通濾波器的係數。且在圖中每次經過一階段分解時， j 到 $j+1$ 的濾波器會進行上取樣一次，圖2.2為示意圖， $\uparrow 2$ 代表上取樣之處理，濾波器係數以零填充增加為兩倍的長度。

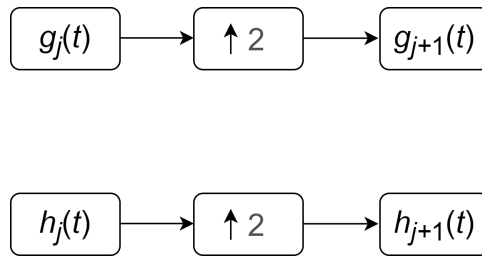


Figure 2.2: 濾波器的上取樣

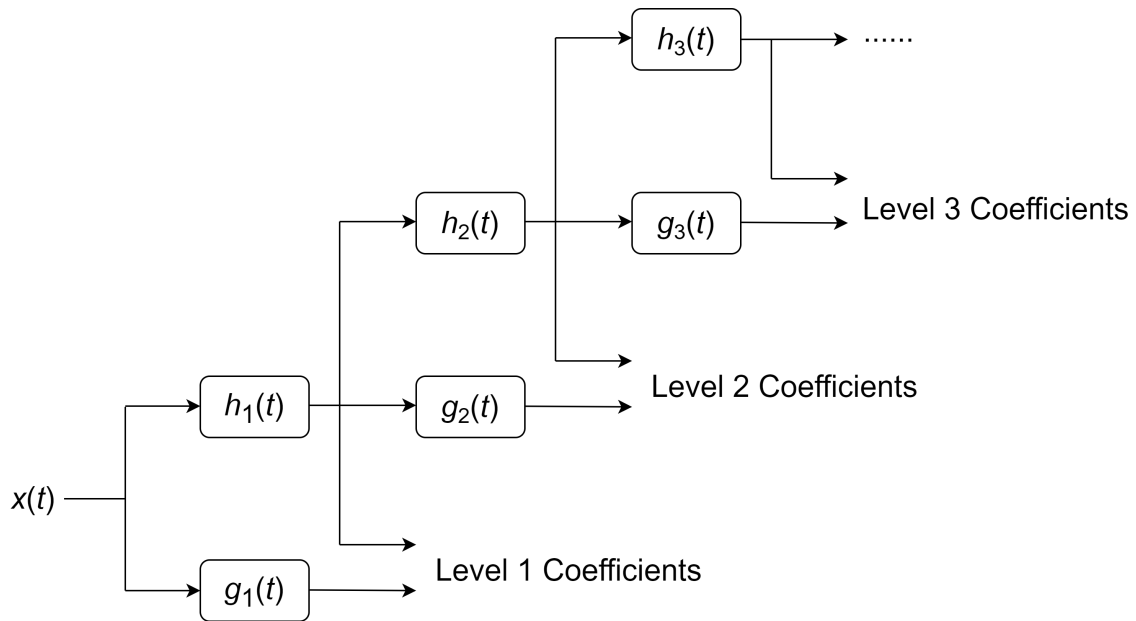


Figure 2.3: 穩態小波轉換 (階層分解)

2.2.3 經驗模態分解

上一節介紹的小波轉換仍然是線性分解，其時間解析度和頻率解析度之乘積為一常數，還是需要在兩者間作取捨，因而受到測不準原理 (uncertainty principle) 所制約；而黃鐸等人於 1988 年提出希爾伯特-黃轉換 (Hilbert-Huang transform, HHT) 則解決了這個問題，讓時間和頻率同時擁有最佳解析度變得可能 [16]。

HHT 包含兩個部分，第一部分運用 EMD 將欲分析之訊號分解為多個頻率高低各不同的本質模態函數 (intrinsic mode functions, IMF)，接著第二部分透過希爾伯特轉換就可以得出每個 IMF 的瞬時頻率。此方法與其他訊號分析方法如：傅立

葉轉換、小波轉換等數學工具不同，它並不是一種理論工具，而是一套應用在實際資料上的演算法。



上述的 EMD 為一套能將非穩態、非線性、高噪音之訊號有效拆解成不同頻率分量的分解流程。跟小波轉換不一樣，EMD 不需要選擇基底函數。該演算法透過找出序列中的局部極值，再利用三次樣條插值（cubic spline interpolation）定義上包絡線 (envelope) 與下包絡線，並不斷減去上下包絡線之中心值，由高至低不斷過濾出不同頻率之分量，直至最終輸出的子訊號呈現穩態，而經拆解之原始訊號可由下式來表示：

$$f(x) = \sum_{j=1}^n \text{IMF}_j + r_n(t) \quad (2.10)$$

其中， r_n 為呈現穩態之剩餘訊號（residual signal），而最低頻率之分量 IMF_n 可視為原始訊號 $f(x)$ 之微緩變化趨勢。

使用 EMD 方式之濾波效果良好，但有可能會發生混模問題（mode mixing problem）和邊界效應（boundary effect），前者會使同一個 IMF 中同時包含差異極大的時間尺度，或者同一個時間尺度的訊號被分散在多個 IMF 中。原因是在分解過程中，極值在短時間內多次變動而導致的；後者則是 EMD 在訊號邊界時會因為周圍極值難以精準判斷，包絡線無法準確定義，使後續分解出之 IMF 其邊界發生震盪或扭曲的現象。

圖2.4顯示了 EMD 將訊號 $f(x)$ 分解為多個 IMF 的流程，利用遞迴演算法不斷拆解訊號。其中，IMF 必須滿足以下情況：

- 局部極值的數量必須等於零交叉點的數量或相差不超過 1，即兩極值間須具有一個零交叉點。
- 在任何一點，上包絡線與下包絡線取平均要接近為 0。

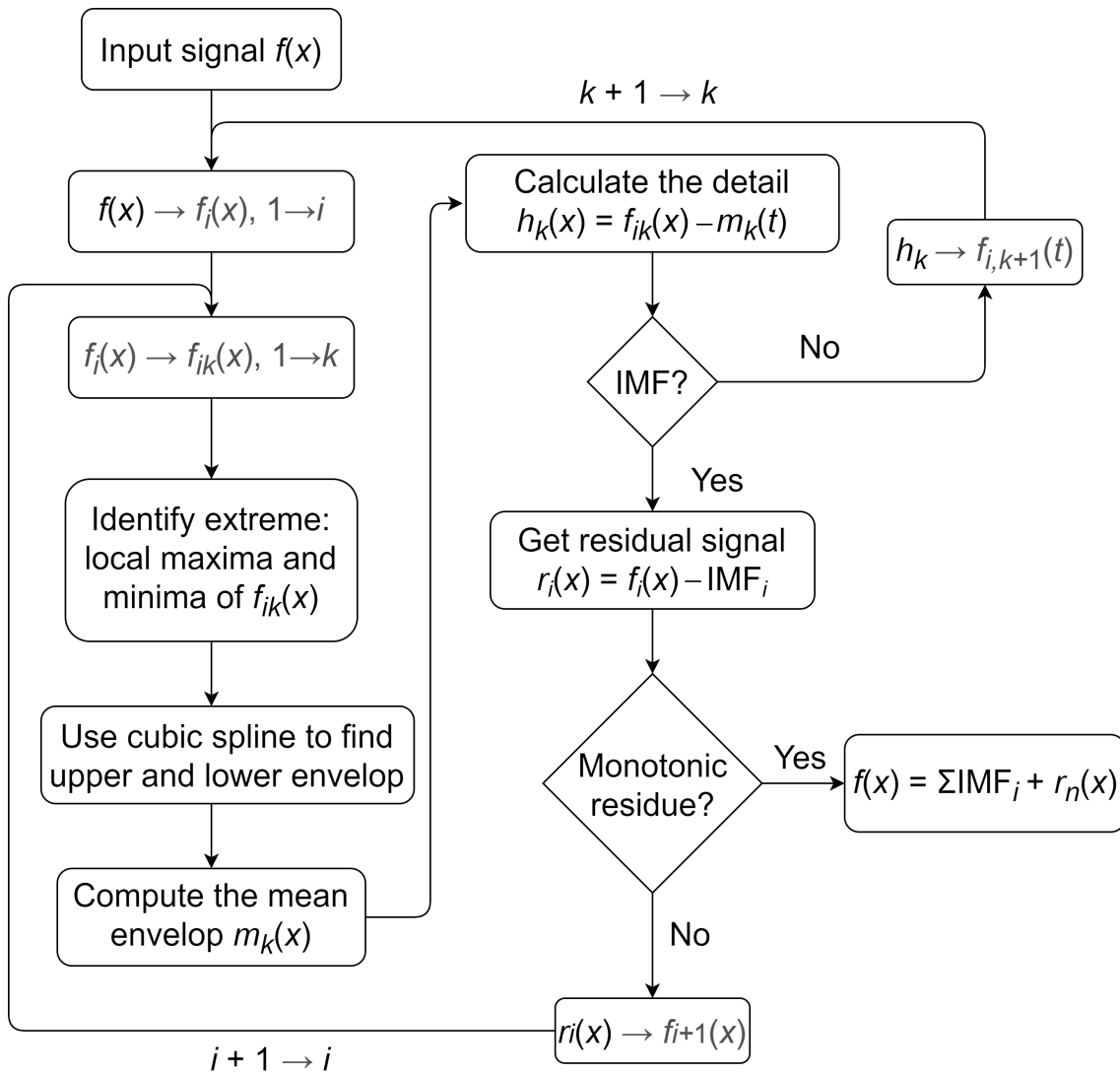


Figure 2.4: 經驗模態分解流程圖



2.3 類神經網路相關文獻

2.3.1 遞迴類神經網路

前饋式類神經網路最大的缺點，在於無法理解輸入本身的時序性，而 RNN 便為此特別開發，用以處理這類序列資料。RNN 和前者最大的差異，在於 RNN 進行計算時會重複使用相同的神經元處理過去及現有的輸入，並在傳遞訊息時會將先前歷史資訊透過狀態 (state) 傳遞至下一個時間點，達到跨不同時間步長共享參數的效果，因此可以處理更廣泛的序列變化。比如一段文章中，某個詞的意思會根據其前後文產生不同的含義，使用 RNN 類型的類神經網路結構就可以好好地解決這類問題。然而單純的 RNN 有一致命的缺點，其權重會隨著遞迴產生梯度爆炸或是發生梯度消失等問題，讓資料中跨較長時間步距的關係不容易被 RNN 記住。意即簡單 RNN 進行訓練時，隨著記憶時間的間隔加大，梯度傳遞能力會以指數數量級上升或下降，訓練時間因此大幅增長，此一缺陷稱之為長期依賴問題 (long-term dependency problem)，後人因此開發 LSTM 來解決這類問題，將於下一節介紹。

圖2.5為一個簡單的 RNN 示意圖，在時間 t 模型接受輸入 x_t 和先前的狀態值 h_{t-1} 並計算當前隱藏狀態 h_t ，因為每個狀態隨時間傳遞下去，最終輸出值 y_t 可能會受到先前時間的輸入值 x_{t-1} 所影響。

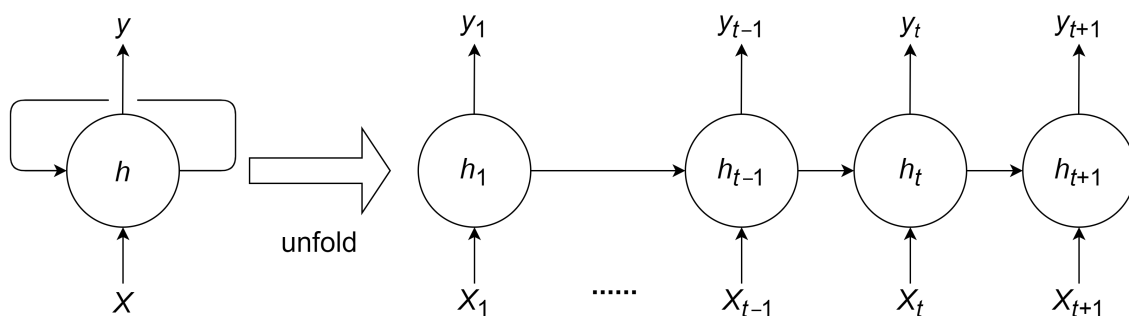


Figure 2.5: 簡單 RNN 步驟展開圖



2.3.2 長短期記憶類神經網路

遞迴類神經網路傳統的學習方式為透過時序反向傳播法 (back-propagation through time, BPTT) 或即時學習演算法 (real-time recurrent learning, RTRL) 來優化參數，但此種優化方法會讓簡單遞迴類神經網路的權重每一輪的更新量成倍取決於誤差大小，導致梯度爆炸 (gradient blow up) 或消失 (gradient vanish)，前者會讓學習過程中的權重產生大幅度的震盪，後者則會使得時間點距離現在較遠的權重難以被更新，因此無法處理較長的序列。

LSTM 是一種遞迴類神經網路的分支，由 Hochreiter 和 Schmidhuber 於 1997 年發表，透過特殊單元的作用，LSTM 可以學習到如何橋接超過 1000 個步長的時間間隔，打造梯度傳播的「高速公路」，使 LSTM 減緩了簡單 RNN 上會有梯度消失的問題，其數學式如下 [14]：

$$\begin{aligned}f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\ \tilde{c}_t &= \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \\c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\h_t &= o_t \odot \sigma_h(c_t)\end{aligned}\tag{2.11}$$

初始值 $c_0 = 0$ 、 $h_0 = 0$ ， \odot 為阿達瑪乘積 (Hadamard product)，表示逐元素積 (element-wise product)。其中：

- $x_t \in \mathbb{R}^d$ ：時間 t 之輸入向量
- $f_t \in \mathbb{R}^h$ ：遺忘閘 (forget gate)
- $i_t \in \mathbb{R}^h$ ：輸入閘 (input gate)



- $o_t \in \mathbb{R}^h$: 輸出閘 (output gate)
- $h_t \in \mathbb{R}^h$: 隱藏狀態向量 (hidden state vector)，同時也是 LSTM 的輸出向量。
- $c_t \in \mathbb{R}^h$: 記憶單元狀態向量
- $W \in \mathbb{R}^{h \times d}$ 、 $U \in \mathbb{R}^{h \times h}$ 和 $b \in \mathbb{R}^h$: LSTM 訓練期間需要經由演算法學習的權重矩陣和偏差參數，其中上標 d 和 h 分別表示輸入特徵的維度和隱藏單元的維度。

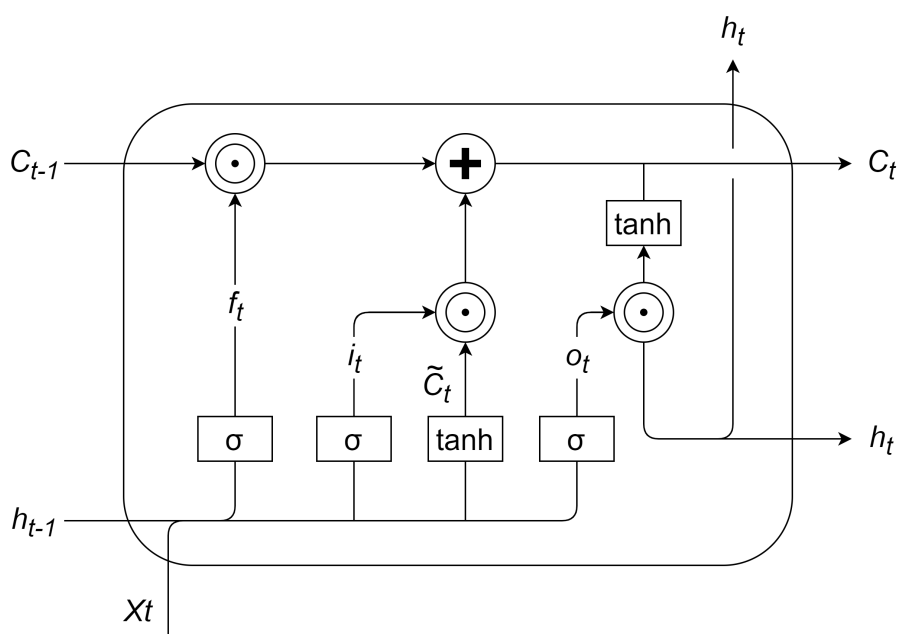


Figure 2.6: LSTM 結構圖

本實驗目標為透過歷史股價資訊預測下一時間點之股價漲跌，為一分類問題，而 LSTM 可以在時間尺度上進行選擇性的記憶，有助於捕捉歷史序列資料中之特徵，因此在實驗中採用 LSTM 進行分類，會有很大的優勢。

2.3.3 雙向遞迴類神經網路

雙向遞迴類神經網路 (bi-directional recurrent neural network, BRNN) 的概念由 Schuster 和 Paliwal 於 1997 年提出，在訓練遞迴類神經網路中同時利用時間序列中過去與未來的資訊作為特徵 [30]。該網路連接兩層 RNN 結構，利用互為反向的序

列作訓練，並且都連接著同一個輸出層，其連接方式可以因目標而異。BRNN 的結構可以把前後文每一個時間點的完整資訊都提供給輸出層使用，以利下一層神經元之分析。圖2.7簡單示意一個沿著時間展開的 BRNN，該方法同樣可以應用到 LSTM 和 GRU 上。Althelaya 在研究中利用堆疊 LSTM (stacked LSTM, SLSTM)、雙向 LSTM (bi-directional LSTM, BLSTM) 和傳統的 LSTM 在 S&P 500 的股價預測上作比較，其結果表明，訓練時 SLSTM 和 BLSTM 與傳統 LSTM 相比收斂的更快，且 BLSTM 不論在長期或短期的預測上都具有較佳的結果 [2]。

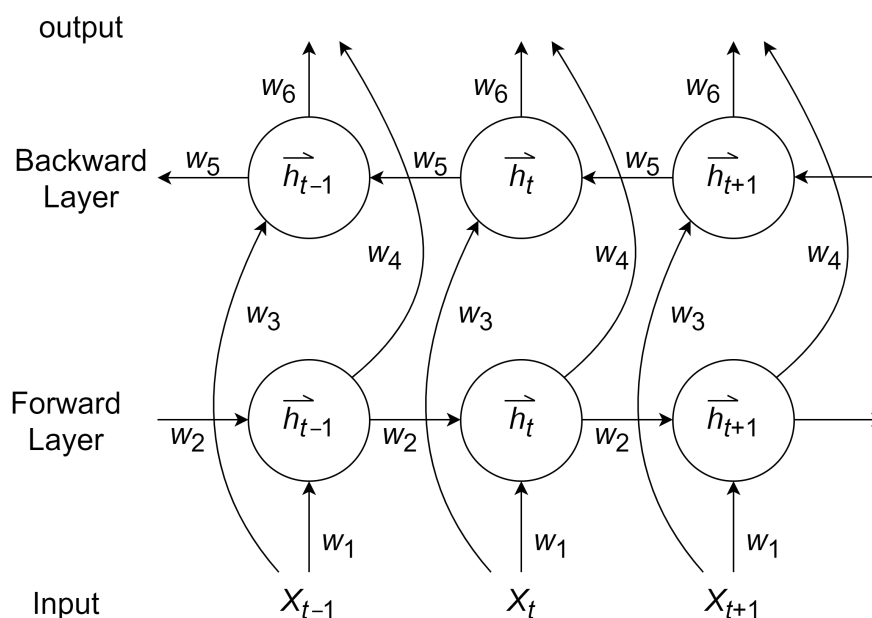


Figure 2.7: BRNN 步驟展開圖

2.4 布穀鳥搜尋演算法相關文獻

2.4.1 布穀鳥搜尋演算法

布穀鳥搜尋演算法是由 Yang 和 Deb 於 2009 年提出的一種新興啟發式演算法 [34]。此最佳化演算法透過 Lévy flight 來加強搜尋機制，而不是簡單的隨機漫步 (random walk)。Lévy flight 和一般的隨機漫步分佈極為相似，但偶爾會發生一些巨幅波動。不少動物的覓食方式就是以 Lévy flight 的方式呈現，牠們會利用一

系列直線飛行和九十度轉彎來探索周圍景觀，在某個小區域內產生出類似 Lévy flight 的行為模式，隨機尋覓食物。但當經過一段時間後，若發現該區域中無法找到食物，就會移動到下一個新的地點，重新展開新一次的搜尋。跟完全採用隨機搜尋的方式比較起來，這種以 Lévy flight 為基礎的覓食方法會比較有效 [29]。

布穀鳥搜尋演算法透過模擬特定種類的布穀鳥其托卵寄生的行為 (brood parasitism) 來有效求解最佳化問題。這類型的鳥會把牠們自己的蛋下在宿主鳥 (host birds) 的巢，並蓄意地去除其他蛋來增加自己的蛋孵化的機率。而若宿主鳥發現這些蛋不是自己的，則會將這些外來卵予以清除，或是直接放棄此巢重新另闢他巢。

簡而言之，布穀鳥搜尋演算法可以透過一開關參數來呈現不同的表現方式，既可以在一定範圍內透過 Lévy flight 進行隨機漫步，另一方面可以透過隨機產生與目前最佳解足夠遠的新解來避免陷入局部最佳解 (local optimum)，因此同時具有局部搜尋能力和全域搜尋能力，即演算法兼顧探勘 (exploration) 與開發 (exploitation) 的特性。目前布穀鳥演算法及其衍生演算法的應用都得到了快速的發展，在實際的工程應用問題中已經表現出其高且優的效率，並在多個標竿測試中擊敗 PSO 和 GA 等其他最佳化演算法 [11]。

作者在布穀鳥搜尋演算法中假設了以下三種理想狀態：

1. 每隻布穀鳥每次只產一顆蛋，並且隨機選擇一個鳥巢存放。
2. 在求解的過程中，所有鳥巢中最優質的蛋會被保留至下一代。
3. 可用的鳥巢總數固定不變，且鳥巢中外來卵被宿主鳥發現的機率為 $p_a \in [0, 1]$ 。若外來卵一經發現，則宿主鳥將另起爐灶，重新築巢。

其中，第三點可由每次疊代時從 n 個巢中隨機替換 $n \cdot p_a$ 個巢來模擬。



根據以上三點規則，該演算法可總結為如演算法1的虛擬碼。該演算法由 p_a 切換全域搜尋和局部搜尋，可寫為：

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t) \quad (2.12)$$

上式中的 H 為 Heaviside 函數（單位階梯函數）， ϵ 為均勻分布中抽取的隨機數， x_j 和 x_k 則為演算法中隨機選擇的兩個不同位置的解，而當產生新解 $x^{(t+1)}$ 時，布穀鳥的位置更新公式如下：

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \otimes \text{Lévy}(s, \lambda) \quad (2.13)$$

$x_i^{(t)}$ 為第 t 代第 i 隻布穀鳥的位置， $\alpha > 0$ 為根據問題規模而設定的步長縮放控制量，通常設定為 0.01 到 1 中間的值可以解決大部分的問題，能有效產生飛行路徑並避免飛得太遠。

式 (2.13) 中的 Lévy 為隨機搜尋路徑，其步長服從萊維分配 (Lévy distribution)：

$$\text{Lévy}(s, \lambda) \sim s^{-\lambda}, \quad (1 < \lambda \leq 3) \quad (2.14)$$

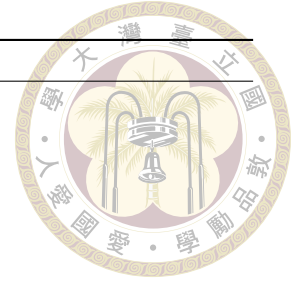
生成正確服從萊維分配的步長最常見的方式為 Mantegna 演算法，利用兩服從常態分配之變數 U 和 V 來產生 [24]：

$$s = \frac{U}{|V|^{\frac{1}{\lambda}}} \quad (2.15)$$

其中

$$U \sim N(0, \sigma^2), \quad V \sim N(0, 1) \quad (2.16)$$

$$\sigma^2 = \left[\frac{\Gamma(1 + \lambda)}{\lambda \Gamma(\frac{1+\lambda}{2})} \cdot \frac{\sin(\frac{\pi\lambda}{2})}{2^{\frac{\lambda-1}{2}}} \right]^{\frac{1}{\lambda}}$$



Algorithm 1 Cuckoo Search via Lévy Flights

- 1: 目標函數 $f(x)$, $x = (x_1, \dots, x_d)^T$
 - 2: 初始化 n 個鳥巢的位置
 - 3: **while** ($t < \text{Max Generation}$) or (Stop Criterion) **do**
 - 4: 隨機選取一隻布穀鳥
 - 5: 透過 Lévy flight 產生一個解 x_i
 - 6: 評估解的 quality/fitness F_i
 - 7: 從 n 個巢中隨機選取一個 (假設為第 j 個巢)
 - 8: **if then** $f_i > f_j$
 - 9: 用 i 的解取代 j 的解
 - 10: 一部分 (p_a) 糟糕的巢被拋棄
 - 11: 新巢/解由式 (2.12) 建立
 - 12: 排列所有解並保存最佳解 (或是部分優質解)
 - 13: $t \leftarrow t + 1$
 - 14: 後處理與視覺化
-

2.4.2 自動調整遞迴類神經網路的超參數

研究人員在調整 RNN 的超參數時，必須根據目標任務為模型的超參數選擇合適的值。但由於極難解釋的模型行為，會讓搜尋空間變得相當巨大，導致此任務非常具有挑戰性。對於特定序列資料集手動調整超參數的 RNN 模型，通常在另外一個資料集上的表現不佳，且手動調整既曠日費時也不具有泛化能力，因此提出一套能夠不依賴於資料集內容的自動調整方法有其必要。

布穀鳥搜尋演算法具有參數少、收斂速度對參數變化不敏感、不易陷入局部最佳解和易於其他演算法相耦合等優點，特別適合用來調適類神經網路中的超參數組合。在 [31] 中，作者提出以布穀鳥搜尋演算法調整 RNN 的超參數之方法，使用 6 種不同類型的 RNN 優化器分別測試，包括 SGD、Adagrad、RMSProp、

Adadelta、Adam 和 Nadam，隱藏層的總數限制為 10 到 50，節點數則限制為 20 到 200，並使用經修改過的布穀鳥搜尋演算法來進行實驗，其中鳥巢數和 p_a 分別設為 50 和 0.25，最終在 EEG Eye State Dataset 等資料集上獲得高精確度的成果。

布穀鳥搜尋演算法的原始版本僅適用於連續空間的最佳化問題，而 RNN 的超參數調整中包含了神經元個數、類神經網路層數、優化器和激勵函數等離散數值，因此仍需稍作修改才能加以應用。

經修改後之布穀鳥搜尋演算法，其宿主巢初始化的過程，改由下式來進行：

$$x_{ij} = \text{round} \{ \text{LL}_j + u \cdot [\text{UL}_j - \text{LL}_j] \} \quad (2.17)$$

其中， x_{ij} 是第 i 個巢的第 j 維， LL_j 和 UL_j 是第 j 個參數範圍的上下限。而產生新解的方式，可將式 (2.13) 改為如下：

$$x_{ij}^{\text{new}} = \text{round} [x_{ij} + \alpha \otimes \text{Lévy}(s, \lambda)] \quad (2.18)$$

並將步長縮放控制量 α 設定為 0.4，與一般連續空間最佳化問題常用接近 0.01 的設定不同， α 必須保持足夠大的值，以避免四捨五入時布穀鳥無法更新到新的位置之情況發生。



第三章 實驗方法

3.1 模型概述

整體模型架構如圖3.1所示，首先將不同的時間序列分解為不同頻率之子訊號，並把子訊號各別用於訓練單獨的 LSTM，再將 LSTM 之輸出通過密集層 (fully connected layer)，最後通過 sigmoid 函數作下一時間點漲跌之分類，其中 LSTM 和密集層之層數可以複數疊加。

有關本實驗之整體流程如圖3.3表示。利用 SWT 將原始的股價訊號拆解為不同頻率之分量時，à trous 演算法所能接受的訊號長度須為 2^n ，其中 $n \in \mathbb{Z}$ 為欲分解之層數，否則須作零填充至最接近之 2 的幕次長度；而利用 EMD 來拆解訊號時，其分解出之 IMF 數量則會依據訊號不同有所差異。考慮模型輸入序列須為固定長度，需將訊號截斷處理，實驗中以尺寸 $L = 16$ 之矩形窗 (rectangular window) 作截斷：

$$S_i = f(i - L + 1), f(i - L + 2), \dots, f(i) \quad (3.1)$$

S_i 為第 i 組從原訊號 f 截斷後之子訊號，如圖3.2所示。

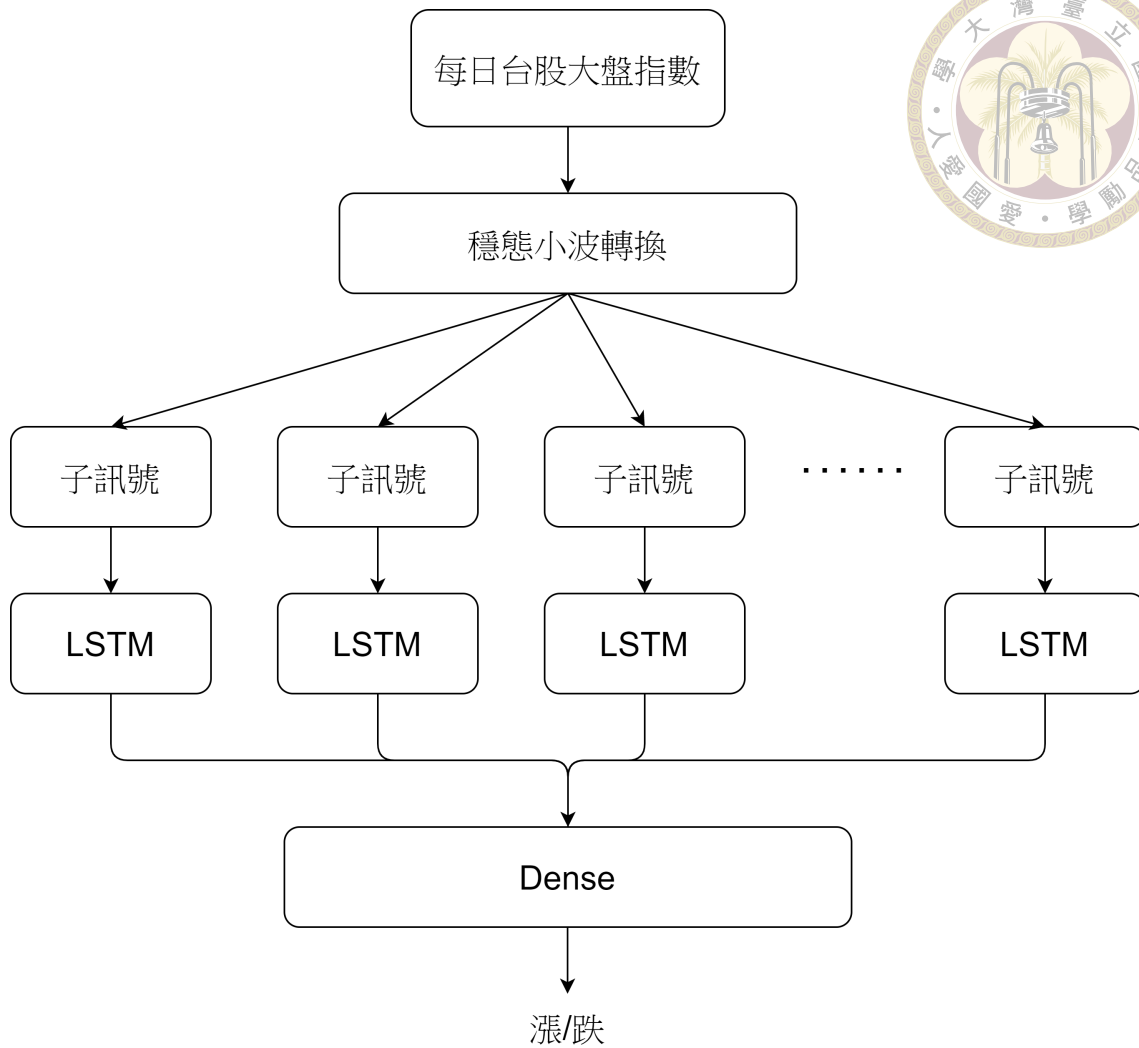


Figure 3.1: 結合小波轉換之 LSTM 展開圖

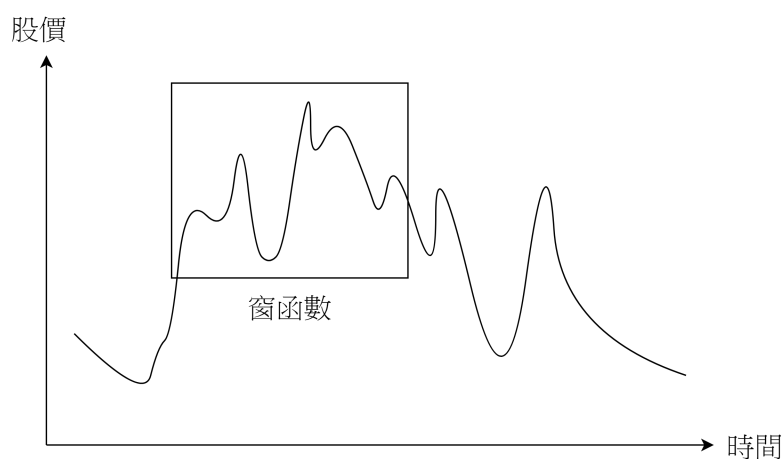


Figure 3.2: 利用窗函數截斷訊號示意圖

實驗中模型之訓練方式為監督式學習 (supervised learning)，為了創造學習過程中所需之漲跌標籤 (label)，可藉由對每日收盤價進行一階差分並取其正負號而



得，並將其與開盤價、最高價、最低價、收盤價和成交量經 SWT 分解後所形成之各子序列合併成為模型訓練之最小單位，即輸入前 L 天之資訊，預測第 $L + 1$ 天的收盤價漲跌。

此次實驗 SWT 採用 Haar、Daubechies-3、Coiflet-3 和 Symlet-3 四種不同的小波基底，與文獻回顧中金融資料相關任務常用之母小波相同。另為提高模型收斂速度和模型精準度，資料標準化採用最大絕對值標準化 (max abs normalization)，將分解完之子訊號各自除以其絕對值最大者，藉以進行特徵縮放並保留訊號之正負號。

另外，實驗中使用 Adam 方法作為權重最佳化之優化器，二元交叉熵 (binary cross entropy) 作為損失函數，若訓練中訓練集和驗證集之錯誤率相差開始擴大時，使用早停法 (early stopping) 終止訓練，並輔以使用放棄法 (dropout) 來緩解模型的過擬合問題。其中，二元交叉熵之計算公式如下：

$$loss = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \quad (3.2)$$

y 為漲跌之標籤， \hat{y} 為模型預測下一時間點收盤價上漲之機率。

Table 3.1: 布穀鳥演算法之參數

參數	值
Iteration	30
鳥巢數量	50
布穀鳥數量	30
p_a	0.25
連續部分 α	0.02
離散部分 α	0.4

LSTM 模型之超參數使用布穀鳥搜尋演算法於表3.1之搜尋範圍內選擇最佳參數值；布穀鳥搜尋演算法本身之參數於表3.2中詳細列出，而優化過程中評估一個解的品質高低，則由該解作為超參數所建立之模型經訓練後，其預測下一時間點

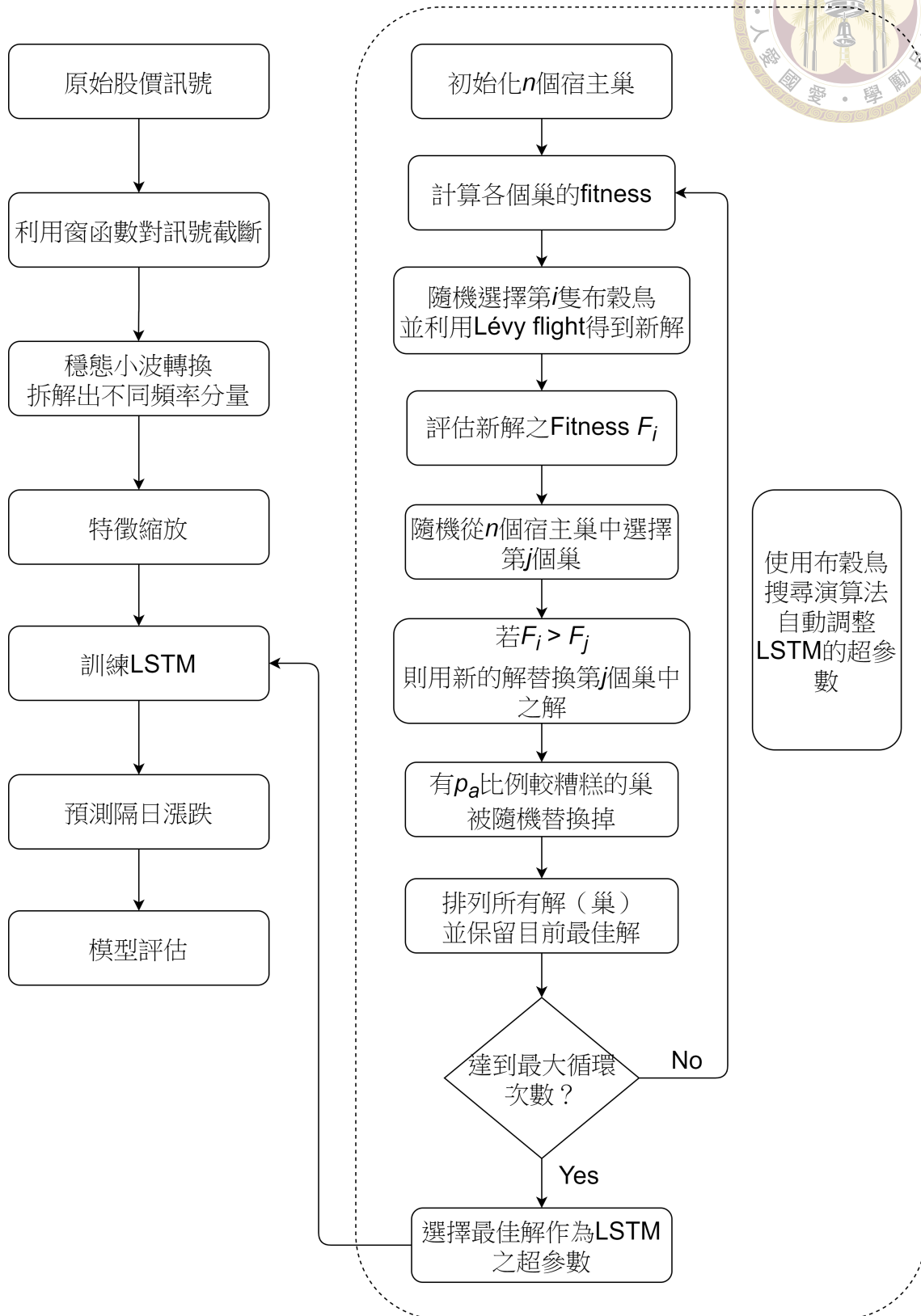
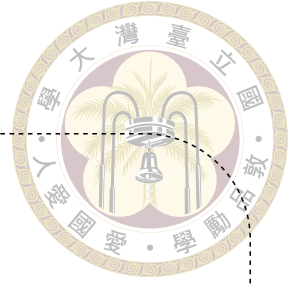


Figure 3.3: 整體實驗架構流程圖

Table 3.2: 超參數搜尋範圍

	搜尋範圍	離散或連續
是否為雙向 LSTM	{True, False}	離散
LSTM L2 正規化	[0, 0.3]	連續
LSTM 神經元數	[64, 256]	離散
LSTM 層數	[1, 3]	離散
LSTM Dropout Rate	[0, 0.3]	連續
LSTM Recurrent Dropout Rate	[0, 0.3]	連續
Dense 層數	[0, 3]	離散
Dense L2 正規化	[0, 0.3]	連續
Dense 神經元數	[4, 128]	離散
Dense 激勵函數	{ReLU, SELU, ELU, Leaky ReLU, Tanh}	離散
Dense Dropout Rate	[0, 0.3]	連續
是否採用 Batch Normalization	{True, False}	離散
Batch Size	[80, 400]	連續



股價漲跌的精確度決定。

3.2 資料來源

為評估深度學習模型之性能，訓練資料的規模必須提供適當的資料量，才足以擬合一個良好的模型，並避免因樣本過少而造成的過擬合，本次實驗使用資料為：

台灣證券交易所 (Taiwan Stock Exchange, TWSE) 公布自 2003 年 1 月 2 日至 2020 年 4 月 30 日之台灣發行量加權股價報酬指數之收盤價與成交量，其中開盤價、最高價和最低價可藉由台灣發行量加權股價指數等比例縮放而得，共 4282 日。

上述資料中，前 4032 日用以訓練模型，其中 2184 天收盤價為漲，1847 天為跌；最後 250 日作為測試集 (testing set)，其中 140 天收盤價為漲，110 天為跌。



3.3 驗證資料處理

在最佳化超參數時，若應用傳統的交叉驗證 (cross-validation) 在時間序列的資料上，可能會有未來資料洩漏之問題存在 [18]，此處選擇使用滾動視窗驗證 (rolling window validation) 方式來避免，如下圖所示：

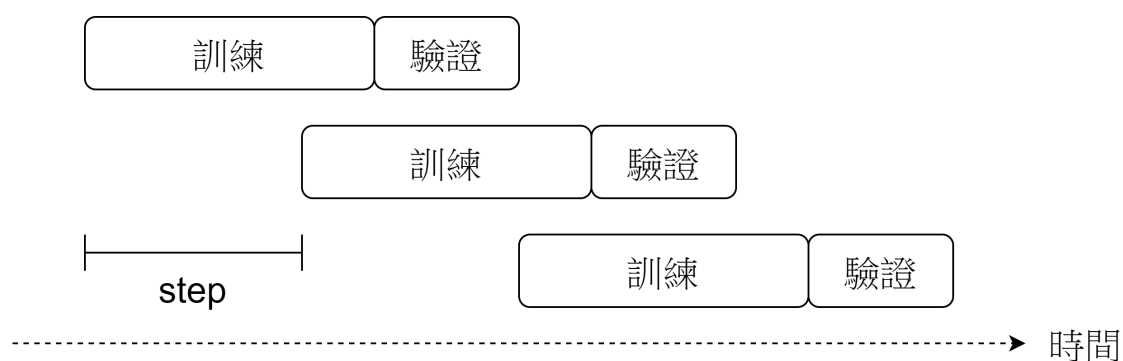


Figure 3.4: 滾動視窗驗證

高度非穩態資料之訓練視窗大小需要謹慎選擇，太大容易學習到過多的無關特徵，太小則容易使模型過度擬合。本研究採訓練視窗大小為 1000 個資料點，驗證視窗大小為 250 個資料點，每次在時間軸上移動步距 (step) 為 1000 個資料點。



第四章 實驗結果

4.1 實驗一

實驗一考慮使用兩組手動設定之超參數，模型輸入為台灣發行情加權股價報酬指數之開收高低與成交量，共 5 時間序列，再分別經 SWT 或 EMD 拆解為數個子序列，預測隔日收盤價之漲跌。其使用超參數如表 4.1 中所示，準確率如表 4.2 至表 4.5 所示。

Table 4.1: 超參數

	第一組	第二組
是否為雙向 LSTM	False	False
LSTM L2 正規化	0	0.01
LSTM 神經元數	128	256
LSTM 層數	1	1
LSTM Dropout Rate	0.25	0.1
LSTM Recurrent Dropout Rate	0.25	0.05
Dense 層數	1	2
Dense L2 正規化	0	0.01
Dense 神經元數	128	256
Dense 激勵函數	ReLU	ReLU
Dense Dropout Rate	0.25	0.1
是否有作 Batch normalization	True	True
Batch Size	400	400

Table 4.2: 準確率 (不分解)

	—
第一組	54.8%
第二組	52%

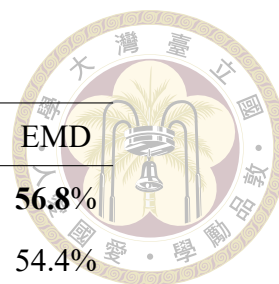


Table 4.3: 準確率 (分解 1 次)

	Haar	Daubechies-3	Symlet-3	Coiflet-3	EMD
第一組	54.8%	54.4%	55.6%	53.2%	56.8%
第二組	55.6%	55.8%	54.4%	55.2%	54.4%

Table 4.4: 準確率 (分解 2 次)

	Haar	Daubechies-3	Symlet-3	Coiflet-3	EMD
第一組	56.4%	56.8%	56%	56.4%	54.4%
第二組	54.4%	53.2%	55.2%	52.8%	54%

Table 4.5: 準確率 (分解 3 次)

	Haar	Daubechies-3	Symlet-3	Coiflet-3	EMD
第一組	53.6%	55.2%	55.2%	54%	56%
第二組	52.4%	51.2%	54.4%	54.8%	55.2%

Table 4.6: 準確率 (分解 4 次)

	Haar	Daubechies-3	Symlet-3	Coiflet-3	EMD
第一組	51.6%	51.2%	52%	49.2%	53.6%
第二組	50.8%	47.2%	50.4%	51.2%	52%

經由實驗一發現，在分解階層超過 2 之後，準確率會隨著分解次數增加而有降低之趨勢。另外，可以觀察到在不同超參數之情況下，會造成相同的分解方式擁有截然不同的結果，因此在未對超參數進行最佳化時不容易比較何種分解方式對於預測指數漲跌更具優勢。



4.2 實驗二

實驗二採用布穀鳥搜尋演算法分別搜尋不同分解方式在分解次數限制為2的情況下之最佳超參數組合，其優化後之超參數結果如表4.7所示，準確率則由表4.8所示。

Table 4.7: 優化後之超參數

	Haar	Daubechies-3	Symlet-3	Coiflet-3	EMD	不分解
是否為雙向 LSTM	True	True	False	False	True	False
LSTM L2 正規化	9.56e-5	0	1.32e-4	4.98e-4	9.3e-6	3.12e-4
LSTM 神經元數	197	182	201	197	127	237
LSTM 層數	1	3	3	1	3	1
LSTM Dropout Rate	0.218	0.025	0.041	0.04	0.228	0.093
LSTM Recurrent Dropout Rate	0.047	0.077	0.171	0.045	0.075	0.156
Dense 層數	2	1	3	2	0	1
Dense L2 正規化	0	1.54e-06	9.59e-05	0	0	0
Dense 神經元數	128	99	118	111	–	128
Dense 激勵函數	Tanh	ELU	ReLU	ELU	–	Tanh
Dense Dropout Rate	0	0.25	0.076	0	–	0
是否有作 Batch normalization	False	False	True	True	False	False
Batch Size	357	274	400	400	317	400

Table 4.8: 優化後超參數下之準確率

	Haar	Daubechies-3	Symlet-3	Coiflet-3	EMD	不分解
準確率	55.6%	54.8%	57.2%	57.6%	55.6%	56.8%

實驗二經過布穀鳥搜尋演算法計算之超參數組合的預測結果，以 Coiflet-3 作為穩態小波轉換基底之分解方法最佳，可達到 57.6% 的準確率。





第五章 結論與建議

5.1 結論

結合濾波器組拆解序列並搭配深度學習模型的研究一直有學者提出，然而對於使用何種拆解方法較好仍無定論，本研究亦受此啟發，嘗試在金融時間序列上應用各種不同的訊號拆解方法進行比較，希望能在不同的小波基底和分解階層數的選擇上提供一個客觀的測試結果，以利後續相關研究的開展。

經由實驗得知，將訊號經過更多次的分解，並不意謂著會有更好的準確率。究其原因，可能是因為較多的子序列會使本實驗設計的模型其參數數量跟著上升，然而過高的複雜度往往會讓模型過度匹配訓練樣本。這樣的訓練結果，反而會使模型在脫離訓練樣本後失去泛化能力，在測試集上得到較差的結果。

實驗也發現用 Coiflet-3 和 Symlet-3 作為穩態小波轉換的基底較佳，這兩者皆是 Daubechies 小波家族的延伸版本，其函數具有更加光滑和對稱的特性，但相對付出的計算成本也更高。不可忽視的是，僅用布穀鳥搜尋演算法進行超參數調整而不進行訊號分解的模型，其表現甚至打敗了一些經過訊號預處理之模型。



5.2 未來展望

小波轉換擁有許多不同的轉換基底，除了本論文採用的幾種小波母函數外，仍有許多小波家族 (wavelet families) 未被考慮到，也有許多小波分析的變體可以嘗試，如雙樹複小波變換 (dual-tree complex wavelet transform, DT-CWT)。

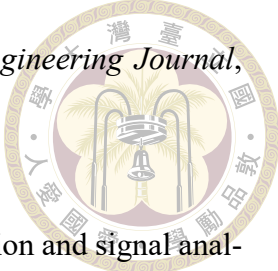
訊號截斷皆有所謂振鈴效應 (ringing artifacts)，選擇使用何種截斷方式與截斷之長度在此實驗中並未探討，僅採矩形窗作訊號截斷之用。而訊號截斷所用之窗函數除矩形窗外，仍有如 Hamming、Bartlett、Kaiser 和三角窗等選擇，也是未來可研究的著眼點之一。

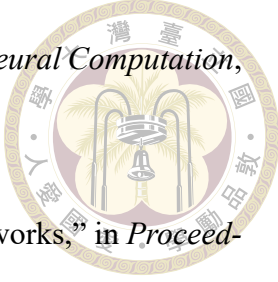
最後，本論文建置之模型，僅預測指數之漲跌，並沒有配套的交易策略，可考慮改採多目標最佳化演算法 (multi-objective optimization)，同時對模型的報酬和漲跌準確率作優化，嘗試建立自動化交易系統。

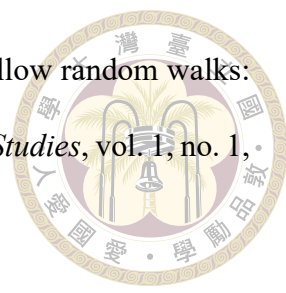


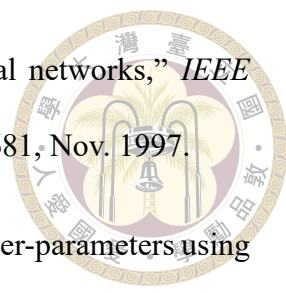
參考文獻

- [1] J. Acevedo, S. Maldonado-Bascón, P. Siegmann, S. Lafuente-Arroyo, and P. Gil-Jiménez, “Tuning L1-SVM hyperparameters with modified radius margin bounds and simulated annealing,” in *International Work-Conference on Artificial Neural Networks*, San Sebastián, Spain, Jun. 2007, pp. 284–291.
- [2] K. A. Althelaya, E. M. El-Alfy, and S. Mohammed, “Evaluation of bidirectional lstm for short-and long-term stock market prediction,” in *2018 9th International Conference on Information and Communication Systems*, Irbid, Jordan, Apr. 2018, pp. 151–156.
- [3] Y. Bengio, “Gradient-based optimization of hyperparameters,” *Neural Computation*, vol. 12, no. 8, pp. 1889–1900, Aug. 2000.
- [4] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, Feb. 2012.
- [5] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, Granada, Spain, Dec. 2011, pp. 2546–2554.
- [6] Chul Hwan Kim and Raj Aggarwal, “Wavelet transforms in power systems part

- 
- 1: General introduction to the wavelet transforms,” *Power Engineering Journal*, vol. 14, no. 2, pp. 81–87, 2000.
- [7] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis,” *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.
- [8] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for event-driven stock prediction,” in *Proceedings of the 24th International Conference on Artificial Intelligence*, Buenos Aires, Jul. 2015, pp. 2327–2333.
- [9] P. Dutilleux, “An implementation of the algorithm à trous to compute the wavelet transform,” in *Wavelets: Time-Frequency Methods and Phase Space*, J. M. Combes, A. Grossmann, and P. Tchamitchian, Eds., Berlin: Springer, 1989, pp. 298–304.
- [10] E. F. Fama, “Efficient capital markets: A review of theory and empirical work,” *Journal of Finance*, vol. 25, no. 2, pp. 383–417, May 1970.
- [11] I. Fister, X. S. Yang, D. Fister, and I. Fister, “Cuckoo search: A brief literature review,” in *Cuckoo Search and Firefly Algorithm: Theory and Applications*, X. S. Yang, Ed., Cham, Switzerland: Springer, 2014, pp. 49–62.
- [12] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, Dec. 2016, pp. 1027–1035.
- [13] C. L. Giles, S. Lawrence, and A. C. Tsoi, “Noisy time series prediction using recurrent neural networks and grammatical inference,” *Machine Learning*, vol. 44, nos. 1–2, pp. 161–183, Jul. 2001.

- 
- [14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [15] H. Hu and G. Qi, “State-frequency memory recurrent neural networks,” in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Aug. 2017, pp. 1568–1577.
- [16] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N. Yen, C. C. Tung, and H. Liu, “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*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.
- [17] F. Hutter, J. Lücke, and L. Schmidt-Thieme, “Beyond manual tuning of hyperparameters,” *KI - Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, Jul. 2015.
- [18] R. J. Hyndman and G. Athanasopoulos, “Forecasting: Principles and practice,” 2019. [Online]. Available: <https://otexts.com/fpp3/>
- [19] N. Jegadeesh and S. Titman, “Returns to buying winners and selling losers: Implications for stock market efficiency,” *Journal of Finance*, vol. 48, no. 1, pp. 65–91, Mar. 1993.
- [20] Y. Kara, M. A. Boyacioglu, and Ömer Kaan Baykan, “Predicting direction of stock price index movement using artificial neural networks and Support Vector Machines: The sample of the Istanbul Stock Exchange,” *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311–5319, May 2011.
- [21] C. L. Liu, “A tutorial of the wavelet transform,” 2010. [Online]. Available: <http://disp.ee.ntu.edu.tw/tutorial/WaveletTutorial.pdf>

- 
- [22] A. W. Lo and A. C. MacKinlay, “Stock market prices do not follow random walks: Evidence from a simple specification test,” *Review of Financial Studies*, vol. 1, no. 1, pp. 41–66, Feb. 1988.
- [23] P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor, “Particle swarm optimization for hyper-parameter selection in deep neural networks,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, Berlin, Jul. 2017, pp. 481–488.
- [24] R. N. Mantegna, “Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes,” *Phys. Rev. E*, vol. 49, no. 5, pp. 4677–4683, May 1994.
- [25] G. P. Nason and B. W. Silverman, “The stationary wavelet transform and some statistical applications,” in *Wavelets and Statistics*, A. Antoniadis and G. Oppenheim, Eds., New York: Springer, 1995, pp. 281–299.
- [26] L. Peng, S. Liu, R. Liu, and L. Wang, “Effective long short-term memory with differential evolution algorithm for electricity price prediction,” *Energy*, vol. 162, pp. 1301–1314, Nov. 2018.
- [27] J. Qiu, B. Wang, and C. Zhou, “Forecasting stock prices with long-short term memory neural network based on attention mechanism,” *PLoS ONE*, vol. 15, no. 1, p. e0227222, Jan. 2020.
- [28] N. Reimers and I. Gurevych, “Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks,” 2017. [Online]. Available: arXiv:1707.06799
- [29] A. M. Reynolds and M. A. Frye, “Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search,” *PLoS ONE*, vol. 2, no. 4, pp. 1–9, Apr. 2007.

- 
- [30] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [31] D. Srivastava, Y. Singh, and A. Sahoo, “Auto tuning of RNN hyper-parameters using cuckoo search algorithm,” in *2019 Twelfth International Conference on Contemporary Computing*, Noida, India, Aug. 2019, pp. 1–5.
- [32] K. Yan, W. Li, Z. Ji, M. Qi, and Y. Du, “A hybrid LSTM neural network for energy consumption forecasting of individual households,” *IEEE Access*, vol. 7, pp. 157633–157642, Oct. 2019.
- [33] X. S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” *International Journal of Mathematical Modelling & Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, May 2010.
- [34] X. S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in *2009 World Congress on Nature Biologically Inspired Computing*, Coimbatore, India, Dec. 2009, pp. 210–214.
- [35] X. S. Yang and S. Deb, “Multiobjective cuckoo search for design optimization,” *Computers & Operations Research*, vol. 40, no. 6, pp. 1616–1624, Jun. 2013.
- [36] S. R. Young, D. C. Rose, T. P. Karnowski, S. Lim, and R. M. Patton, “Optimizing deep learning hyper-parameters through an evolutionary algorithm,” in *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments*, New Austin, Texas, Nov. 2015, pp. 1–5.
- [37] B. Zhang, “Foreign exchange rates forecasting with an EMD-LSTM neural networks model,” *Journal of Physics: Conference Series*, vol. 1053, no. 1, p. 012005, Jul. 2018.