

國立中央大學

資訊管理研究所

碩士論文

(初稿)

混合式機器學習於數據預測之應用

Hybrid Machine Learning in Data Prediction for
Applications

研究生：連芷潑

指導教授：李俊賢 博士

中華民國 107 年 6 月

混合式機器學習於數據預測之應用

研究生：連芷潑

指導教授：李俊賢 博士

國立中央大學資訊管理學系碩士班

摘要

本論文研究中之預測模型為複數模糊類神經模型，藉由複數模糊集 (Complex fuzzy set) 取代傳統模糊類神經模型 (Fuzzy neural network) 中使用的傳統模糊集，並採用混合式機器學習進行模型參數學習，包含提出以粒子群最佳化 (Particle Swarm Optimization, PSO) 與隨機最佳化 (Random Optimization, RO) 並行運作之演算法，以及遞迴最小平方估計法 (Recursive least squares estimator, RLSE)。另外，資料進入模型學習前，先經由基於夏農資訊熵 (Shannon Entropy) 的特徵選取方法，選出對目標有影響力之特徵作為模型輸入。特徵選取方法是藉由計算特徵對於目標提供的資訊量多寡，來進行特徵的挑選。複數模糊集合比傳統的模糊集合具有更多的空間以附載更多的資訊，運用於模糊類神經網路時，使在神經網路內部傳遞訊息時，能夠包含更大量的資訊，提升模型預測準確度，且藉用複數的性質，模型能夠進行多目標的處理。在機器學習階段，藉由粒子群最佳化與隨機最佳化的並行運作，並透過競爭與學習的策略，增加找到更佳解的機率，再加上與遞迴最小平方估計法作結合，來提升模型運算效率。在模型預測方面，本論文以股票與匯率作為實驗對象，並從實驗結果顯示本論文提出之混合式機器學習、特徵選取與模型都有良好的表現。

關鍵字： 複數模糊集合；複數模糊類神經網路；粒子群最佳化演算法；隨機最佳化演算法；遞迴最小平方估計法；特徵選取；時間序列預測。

Hybrid Machine Learning in Data Prediction for Applications

Graduate student: Jhih-Ying Lian

Advisor: Dr. Chunshien Li

Department of Information Management, National Central University, Taiwan

Abstract

In the study, the predictive model is a complex fuzzy neural model. The complex fuzzy sets are used to replace the traditional fuzzy set used in the traditional fuzzy neural network. Based on parallel operation with the particle swarm optimization (PSO) algorithm and random optimization algorithm, an improved algorithm is proposed, and combined with the recursive least squares estimation (RLSE) into a hybrid machine learning algorithm, called the RoPso-RLSE learning method. In addition, a feature selection method based on Shannon entropy is presented to select useful information by significant features which will be used as model inputs in modeling. In this study, the feature selection, complex neural fuzzy system and hybrid machine learning algorithm are used for time series prediction of stock price and exchange rate.

The feature selection selects features by calculating the information provided by the features for the targets. Complex fuzzy sets have better description capabilities than tradition fuzzy sets to explain the membership of set elements. They can be used in neural fuzzy networks to transmit more information and increasing the prediction performance of model. Moreover, due to the property of CFSs, the model can perform multi-target forecasting simultaneously. In the machine learning stage, the hybrid algorithm RoPso, compared to use single PSO or RO only, can increase the probability of finding the optimal solution, with fast learning convergence. In addition, combining the RLSE with RoPso can reduce the loading of machine learning by the RoPso alone. Several real-world data sets of stock prices and exchange rates have been used to test the proposed approach in the experiments for multi-objective prediction. Through the experimental results, the proposed approach has shown good performance. In the experiments, this study uses the stock and exchange rate as experimental targets and from the experimental results shows that the proposed approach has a good performance.

Keywords : Complex fuzzy set; Complex fuzzy neural system; Particle swarm optimization algorithm; Random optimization; Recursive least squares estimation; Feature selection; Time series prediction.

目錄

章節	頁碼
摘要.....	i
Abstract	ii
目錄.....	iii
圖目錄.....	v
表目錄.....	vii
第一章 緒論.....	1
1.1 研究背景與目的.....	1
1.2 研究方法.....	1
1.3 論文架構.....	2
第二章 文獻探討.....	3
2.1 特徵選取.....	3
2.2 模糊集合.....	4
2.3 複數模糊集合.....	6
2.4 模糊類神經網路.....	7
2.5 機器學習演算法.....	9
2.5.1 隨機最佳化演算法.....	10
2.5.2 粒子群最佳化演算法.....	11
第三章 系統設計與架構.....	13
3.1 特徵選取.....	13
3.2 複數模糊類神經模型.....	16
3.3 混合式機器學習演算法.....	19
3.3.1 並行式最佳化 RoPso.....	19
3.3.2 遞迴最小平方估計法.....	22
3.4 研究整體流程.....	24
第四章 實驗實作與結果.....	27

4.1 實驗 1: 雙目標股票指數預測-.....	27
4.2 實驗 2: 三目標股票指數預測-.....	33
4.3 實驗 3: 四個目標匯率預測-.....	39
第五章 實驗結果討論	49
第六章 結論	52
6.1 結論	52
6.2 未來研究方向	53
參考文獻	55

圖目錄

圖	頁次
圖 1 溫度冷熱的明確集合	5
圖 2 溫度冷熱的模糊集合	5
圖 3 常見的歸屬函數。	6
圖 4 複數高斯模糊集合圖。	7
圖 5 類神經網路圖	9
圖 6 模糊類神經網路圖	9
圖 7 資訊影響矩陣範例，特徵變數 f_2 提供給目標 Target 的影響資訊量為 0.25，特徵變數 f_3 提供給特徵變數 f_2 的影響資訊量為 0.53.....	15
圖 8 複數模糊類神經系統模型	17
圖 9 PSO 之粒子移動變化比較示意圖。	21
圖 10 RoPso 流程圖。	22
圖 11 研究模型架構圖.....	26
圖 12 特徵對目標 Intel 與 Microsoft 所提供之累積增益量	27
圖 13 模型學習曲線 (實驗一).....	30
圖 14 模型訓練與預測結果-Intel.....	30
圖 15 訓練結果與目標之誤差- Intel.....	31
圖 16 模型訓練與預測結果-Microsoft	31
圖 17 訓練結果與目標之誤差-Microsoft	31
圖 18 特徵對目標 British Airlines、Southwest Airlines 與 Ryanair Holdings Ltd 所提供之累積增益量(實驗二)	33
圖 19 模型學習曲線	37
圖 20 模型訓練與預測結果-British Airlines.....	37
圖 21 訓練結果與目標之誤-British Airlines.....	37
圖 22 模型訓練與預測結果-Ryanair Holdings Ltd	38
圖 23 訓練結果與目標之誤差-Ryanair Holdings Ltd	38
圖 24 模型訓練與預測結果-Southwest Airlines.....	39
圖 25 訓練結果與目標之誤差-Southwest Airlines.....	39
圖 26 特徵對目標 CAD/USD、CHF/USD、DEM/USD 與 GBP/USD 提供之累積增益量.....	40
圖 27 模型學習曲線(實驗三).....	43
圖 28 模型訓練與預測結果(CAD/USD)	43
圖 29 訓練結果與目標之誤差(CAD/USD)	43
圖 30 模型訓練與預測結果(CHF/USD).....	44
圖 31 訓練結果與目標之誤差(CHF/USD).....	44

圖 32 模型訓練與預測結果(DEM/USD).....	45
圖 33 訓練結果與目標之誤差(DEM/USD).....	45
圖 34 模型訓練與預測結果(GBP/USD).....	46
圖 35 訓練結果與目標之誤差(GBP/USD).....	46

表目錄

表	頁次
表格 1 模型參數配置(實驗一).....	28
表格 2 RoPso-RLSE 學習演算法參數設定(實驗一).....	29
表格 3 與其他文獻之比較(實驗一).....	32
表格 4 10 次實驗之統計結果(實驗一).....	32
表格 5 模型參數配置(實驗二).....	34
表格 6 RoPso-RLSE 學習演算法參數設定(實驗二).....	35
表格 7 與其他文獻之比較(實驗二).....	36
表格 8 10 次實驗之統計結果(實驗二).....	36
表格 9 模型參數配置(實驗三).....	41
表格 10 RoPso-RLSE 學習演算法參數設定(實驗三).....	42
表格 11 與其他文獻之比較(實驗三).....	47
表格 12 10 次實驗之統計結果(實驗三).....	48

第一章 緒論

1.1 研究背景與目的

在數據增長快速的時代，資料的應用日趨重要，在目前與資料相關的研究與應用當中，時間序列的相關預測是主要議題。時間序列是按時間順序的排列的一組資料序列，如股票、溫度等。目前時間序列廣泛應用於各領域，如財經[16][17]、資源分配最佳化[15]、季風降水量預測[16]、風速預測[18]、地震預測[19]、短期電力負荷預測[20]等。對於時間序列預測和建模，許多研究採用了智慧型方法，尤以類神經網路為基底是目前人工智慧最常使用的方法。當類神經網路進行學習時需要透過機器學習，最常用於類神經網路的是倒傳遞演算法(Backward-Propagation)，而其它型式與類型的機器學習演算法，如粒子群最佳化、隨機最佳化、蜂群演算法、蟻群演算法等，皆可運用於最佳化神經網路中之參數。此外，隨著資料量增長迅速，對於以資料為學習基礎的機器學習，是個良好發展與應用的機會。但是，並非資料越多，機器學習就能表現越好，有時過多的資料反而會造成機器學習的阻礙。其一，機器學習的時間會隨資料量增長而增加；其二，資料中除了資訊也隱含許多雜訊，若將過多雜訊進行學習容易導致學習效果不佳。因此，如何從資料中選出對目標有高度影響力的資料作為學習，也是目前重要的議題之一。而本篇論文研究也是在人工智慧的領域進行研究，將提出一個智慧型的計算模型，此模型結合複數模糊集與模糊類神經網路，利用複數的特性提升模型輸出的準確度與數量。在機器學習演算法方面，論文中提出使用混合式的機器學習演算法，結合不同的演算法的運作，來提升找尋更佳參數的機會與模型訓練效率。在資料處理部分，本論文中將透過特徵選取的方法，用於模型輸入的資料，來篩選出與目標高度相關的特徵。

1.2 研究方法

首先，針對原始資料先進行特徵選取，透過夏侖資訊熵的概念，計算原始資料對於目標提供的資訊量，篩選出對目標有影響力的特徵作為模型輸入，以降低訓練的資

料量與減少訓練資料中的雜訊。本論文模型主要是以模糊類神經為基礎，在模糊集合上改採用複數模糊集(Complex fuzzy set, CFS)，藉由複數模糊集的特性，除了能夠增加傳遞的資訊量且能同時進行多目標的預測。模型中的 If-Then 模糊規則採用 Takagi-Sugeno (T-S) 的型式，而 If 與 Then 分別又稱作前鑑部與後鑑部。之後透過混合式的機器學習演算法來更新模型參數，其中，利用並行粒子群最佳化(Particle Swarm Optimization, PSO)與隨機最佳化(Random Optimization, RO)的最佳化演算法 RoPso 更新前鑑部參數，遞迴最小平方估計(Recursive least squares estimator, RLSE)則負責更新後鑑部參數。結合上述形成本論文之模型，用於數據預測以期達到良好的預測效果。

1.3 論文架構

本篇論文一共分作六章，第一章為緒論，介紹本論文的背景、動機、研究目的與本論文將使用的方法。第二章為文獻探討，包含特徵選取、模糊集合論、複數模糊集合、模糊類神經網路與機器學習演算法，將針對研究中使用到的相關理論、方法進行文獻回顧。第三章為系統設計與架構，介紹如何將特徵選取用於輸入資料的篩選、本論文的複數模糊類神經模型如何運作以及如何利用並行式演算法 RoPso 和遞迴式最小平方方法分別更新前鑑部與後鑑部的參數。第四章為實驗實作與結果，實驗將採用財金市場的歷史資料進行模型的訓練與預測，預測結果將與其他文獻研究結果相互比較。第五章為實驗結果的討論與研究方法的討論。第六章則是本論文的結論、貢獻以及未來的研究方向。

第二章 文獻探討

2.1 特徵選取

特徵選取 (feature selection) 的目標是要從原有的特徵中挑選出最好的部分特徵，由於資料集中通常包含著比建立模型所需的資訊還要更多，若資料集裡的資訊過於鬆散，將資料加入模型運算後，所能獲得的效果將會受限，以及在建立模型時保留過多不需要的資料，將導致消耗過多的運算資源。因此，對於模型的學習與預測而言，擁有良好的學習資訊是關鍵，而根據特徵選取的形式，可以將特徵選取的方法分作三種：包裝法，嵌入法和過濾法。

包裝法，為利用特徵子集合來訓練模型，透過對訓練過的模型替特徵子集合打分數，來判斷是否新增或刪除新特徵，藉由搜尋的策略來逐一將能夠提高適應值的特徵納入特徵子集合中。常見的包裝法包括循序向前選取法 (sequential forward selection, SFS)、循序向後選取法 (sequential backward selection, SBS) 與疊代剔除法等等，循序向前選取法採用循環的方式，從一開始不包含任何特徵的模型來訓練，在每一次循環都持續加入能夠提升模型的特徵，直到任何特徵都無法在提升模型的表現。循序向後選取法，在一開始先採用所有的特徵來訓練模型，在逐一的刪除掉對模型表現沒有提升益處的特徵，直到模型表現收斂。疊代剔除法，是一種貪心算法，藉由剔除每次循環中最優或最劣特徵來訓練模型，再利用剩餘的特徵於下次循環中建模，直至所有特徵被刪除完，最後利用被刪除的順序給所有特徵排序作，為特徵重要性的依據。整體來說，包裝器的特徵選取方法通常需要較多的計算量，且容易導致學習模型過適 (overfitting) 的結果，但此方法在搜尋有效的特徵子集合上有較好的表現。

嵌入法，相對於包裝法，嵌入法是將特徵選取的方法嵌入選定的模型中，選出對模型訓練表現有益的特徵，這類的典型方法是構建線性模型的 LASSO，此方法給回歸係數加入了 L1 懲罰項，導致模型中的一些參數趨近於 0，而這些參數即為模型中各個特徵的權重係數，當權重係數為 0 時，則表示該特徵對模型表現結果沒有貢獻，因此

將被剔除，而任何係數不為零的特徵都會被 LASSO 算法選中。其它種嵌入式方法，如由 Guyon 等人在對癌症分類時提出來的支持向量機的遞歸特徵消除 (SVM-REF) [2]，其最初只能對兩類數據進行特徵提取，透過使用 SVM 模型來評估特徵，反覆地構建同一個模型來移除低權重的特徵。

過濾法，過濾法的特徵選取方式，與包裝法類似，都是運用搜尋的機制來找到適當的特徵子集合。不同於包裝法使用預測模型給特徵子集合打分數來作為指標，此方法的常用指標包括互資訊、資訊熵或相關係數等等，透過間接衡量資料自身特性來評斷是否為重要特徵來進行建模，透過逐一變化特徵，評估該特徵使用與不使用時模型表現的差別，若當不使用某個特徵時，導致模型的表現有明顯的差異，則表示該特徵重要性之大。其中一種典型的過濾法，Relief，是效率較高的維數簡化方法，Relief 系列算法主要包括最初由 Kira 和 RenMicrosoft 提出的 Relief 和 ReliefF，其中 ReliefF 主要用於解決多分類，數據缺失和存在噪聲等問題[1]。Relief 計算複雜度較低，不使用分類精度作為評價函數，但基於特徵權重的計算方式，因此在進行特徵選擇時，只能提高與標籤關聯度高的特徵的權重值，剔除權重值低的特徵，所以無法有效的去除冗餘特徵。而在本篇論文所使用的特徵選取法同樣也是過濾法。

2.2 模糊集合

模糊集合(Fuzzy set)為學者 Zadeh 於 1965 年所提出的[3]，是傳統集合的一種擴展。傳統集合，又可稱為明確集合(Crisp set)，在明確集合中的元素只能屬於或不屬於某個集合，換言之，每個元素對於每個集合而言，其歸屬程度只有 0 或 1 的二元表示法。若一任意元素 x 屬於集合 S ，記為 $x \in S$ ，對於明確集合中的元素於集合的映射關係，可用特徵函數(Characteristic function)來表示。定義 ϕ_S 為集合 S 的特徵函數，表示如下。

$$\phi_S(x): X \rightarrow \{0,1\} \quad (1)$$

其中， X 表示為集合 S 內元素 x 的來源， $x \in X$ ， X 稱之為論域(Universe of discourse)。此特徵函數透過 0 與 1 表示元素 x 屬不屬於集合 S 。如判斷溫度高低以氣溫 25 度來分界，

亦即只有冷與熱兩種可能，如圖 1 所示。

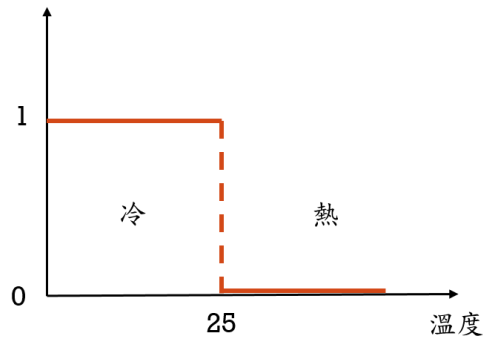


圖 1 溫度冷熱的明確集合

對於模糊集而言，則是由歸屬函數(Membership function)來表示。相對於明確集合的特徵函數只有兩個值，模糊集的歸屬函數其值範圍在 $[0,1]$ 閉區間之中，而歸屬度值的大小，反映元素對於模糊集的歸屬程度。定義 μ_A 為模糊集 A 的歸屬函數。元素 x 屬於模糊集 A 的程度可表示如下。

$$\mu_A(x) \rightarrow [0,1] \quad (2)$$

由於兩者之間的差異，因此模糊集合可以視為傳統集合的擴張。在日常生活中，人類在語意的表達上，常常是模糊不清楚，例如「距離很遠」、「速度很慢」、「身高很高」等等語意含糊的句子，但距離多遠算遠、速度為時速多少算快、身高多高算高，這些句子都沒有明確的定義來說明其意義，因此模糊集合可以用來處理這類型的模糊描述，針對傳統集合對於一些需要明確描述的問題，模糊集合提供了有效的解決方式(如圖 2)。

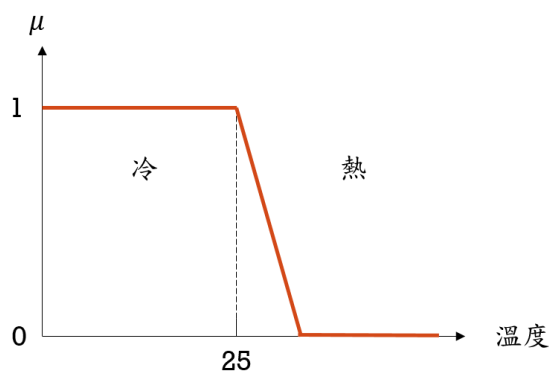


圖 2 溫度冷熱的模糊集合

在模糊集中，常見的歸屬函數主要為三角型函數(圖 3-a)、梯型函數(圖 3-b)、高斯型函數(圖 3-c)、鍾型函數(圖 3-d)等。

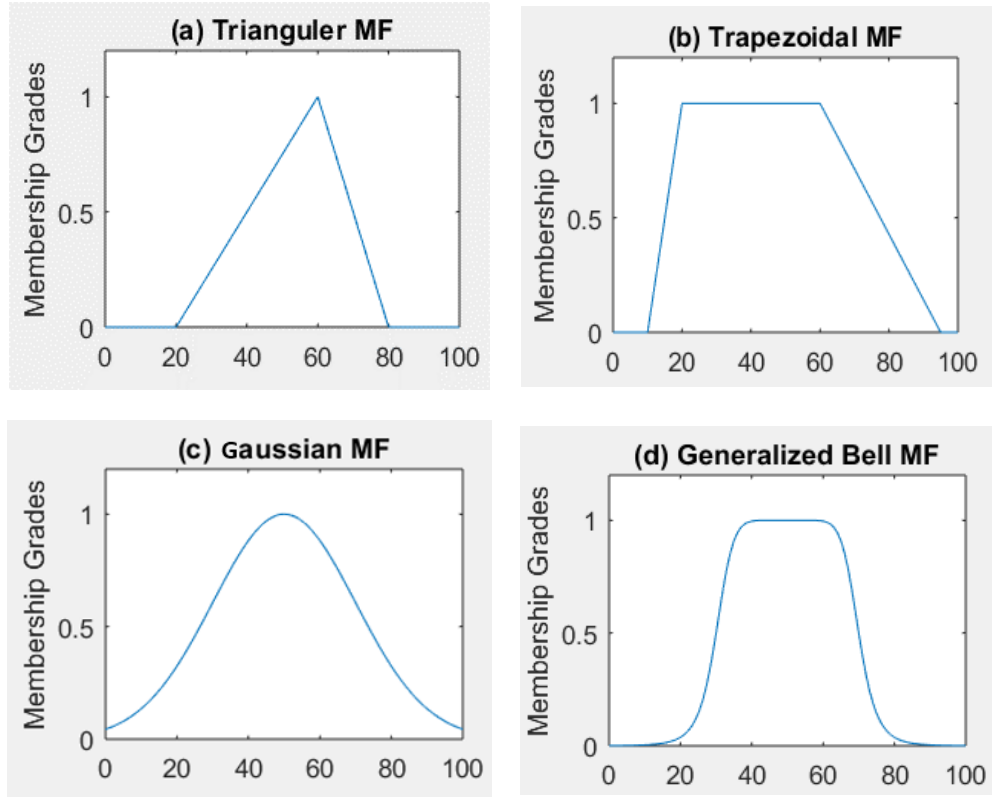


圖 3 常見的歸屬函數。

左上 a 為三角型函數；右上 b 為梯型函數；左下 c 為高斯型函數；右下 d 為鍾型函數。

2.3 複數模糊集合

Ramot et al. [4] 在 2002 年提出了複數模糊集合 (Complex fuzzy set, CFS) 的概念，複數模糊集可以視為模糊集合的延伸，將原始模糊集合中介於 0 到 1 的歸屬程度，延伸到複數平面上，其歸屬程度包含在平面的二維單位圓盤之中，由於進到複數空間中，歸屬程度比起一般模糊集多了虛部，使得複數型歸屬程度能夠負載更多的資訊，進而提升模糊集合的適應力與映射能力。設一複數模糊集 F 其歸屬函數 $\mu_F(h)$ 表示如下。

$$\begin{aligned}\mu_F(h) &= r_F(h) \exp(j\omega_F(h)) \\ &= r_F \cdot \cos(\omega_F(h)) + j \cdot r_F(h) \cdot \sin(\omega_F(h))\end{aligned}\tag{3}$$

其中， F 為論域 U 上的模糊集合，定義為 $F = \{(h, \mu_F(h)) | h \in U\}$ 。 $r_F(h)$ 為振幅函數； $\omega_F(h)$ 為相位函數。當 $\omega_F(h)=0$ 時，複數模糊集將退化成傳統集合，換言之，傳統模糊集合可視為複數模糊集的一種特例。將高斯模糊集使用於複數模糊集合，其歸屬函數表示如下。

$$\text{GaussianComplexType}(h, m, \sigma, \lambda) = r_F(h, m, \sigma, \lambda) \exp(j\omega_F(h, m, \sigma, \lambda)) \quad (4)$$

$$r_F(h, m, \sigma, \lambda) = \text{Gaussian}(h, m, \sigma) = \exp\left(\frac{(h-m)^2}{2\sigma^2}\right) \quad (5)$$

$$\omega_F(h, m, \sigma, \lambda) = -\exp\left(\frac{(h-m)^2}{2\sigma^2}\right) \cdot \frac{(h-m)}{\sigma^2} \cdot \lambda \quad (6)$$

其中， $j = \sqrt{-1}$ ； h 是數值變數； m 為中心值； σ 為延展程度； λ 為是相位變數。

在 2013 年時 Li 和 Chiang [25] 等人成功的將複數模糊集合運用於模糊類神經系統 (Complex neuro-fuzzy system, CNFS)，並且使用複數模糊類神經系統結合 ARIMA 模型，從該文獻的實驗結果證實了複數模糊集合用於模糊類神經系統確實能夠提升模型的預測準確性，因此，在本篇論文中也將傳統模糊集合替代為複數的模糊集合。

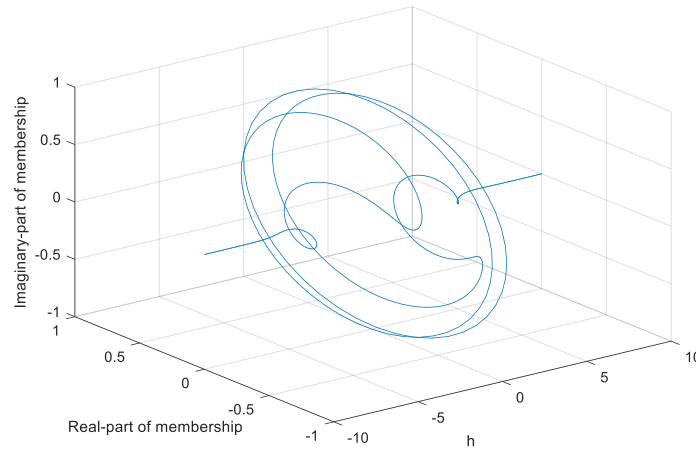


圖 4 複數高斯模糊集合圖。

2.4 模糊類神經網路

類神經網路是一種使用相連的人工神經元來模仿生物神經網路的計算模型，一般常見的多層結構網路由三部分組成，輸入層、隱藏層與輸出層，每一層皆由一些神經

元建構而成，而同一層的神經元彼此不相連，不同層間的神經元彼此相連，訊號方向為前導式的，由輸入層傳輸至輸出層，如圖 5。類神經網路具有優異的學習能力與適應能力，能夠對未知的資料進行正確的預測輸出，因此類神經網路被廣泛的應用在各種領域中，尤其是在時間序列的預測上，但在類神經網路經由學習後，會得到一個如黑盒子般的模型，難以用人類的知識來解釋模型的意義。

此外，模糊系統(Fuzzy system)，是一種將輸入、輸出和狀態定義在模糊集合上的系統，為確定性系統的推廣。模糊系統廣義上而言，較符合人腦思維的模糊性特質，在描述知識方面有其優勢，可以模仿人類的推斷來解決模糊訊息處理的問題。模糊系統由於其能夠解決非線性問題，因此已廣泛應用於各領域上，如自動控制等。在類型方面，模糊系統中主要分作 Mamdani 型[13][14]與 Takagi-Sugeno (T-S)型[12]。Mamdani 型模糊系統，其模糊規則的前鑑部(If part)和後鑑部(Then part)均為模糊語言值，該系統會在輸入和輸出的部分，分別進行模糊化與解模糊化的動作，系統的輸入與輸出皆為精確值，因此可以直接在實際工程中應用。由於該系統應用廣泛，又稱為模糊系統的標準模型。T-S 型模糊邏輯系統，其模糊規則不同於一般的模糊規則形式。特色是輸出可以用輸入值的線性組合來表示，因此系統的輸出值是明確數值，不同於 Mamdani 需透過解模糊化。由於線性組合的特性，能充分利用參數估計的方法來確定系統的參數。T-S 系統的缺點是規則的輸出部分不具有模糊語言值的形式，因而不能充分利用專家知識，模糊邏輯的各種原則在此系統中的應用的自由度也受到限制。

學者 Jang [5]在 2003 年結合模糊系統與類神經網路，提出了模糊類神經系統，如圖 6。其改善了過去類神經網路不易被理解的特性，再加入模糊系統後，便可具有接近人類的思考行為，並且以 If-Then 的法則來進行推論，讓模糊類神經網路同時具備兩者的優點。在 G. S. Atsalakis 與 K. P. Valavanis 的研究中[6]，使用了神經模糊推理系統對股票進行走勢的預測，並以紐約證券交易所(NYSE)的數據來訓練和評估該系統，在短期股票市場趨勢中與其他方法相比，提出的方法有很大的改進和更好的預測。

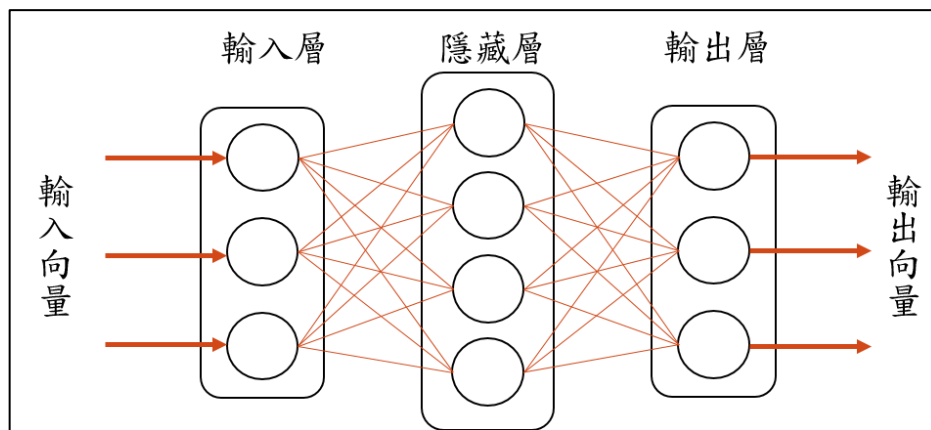


圖 5 類神經網路圖

Rule¹: IF 室溫 is 高 AND 溫差 is 大 THEN 風速 is 強

Rule²: IF 室溫 is 低 AND 溫差 is 小 THEN 風速 is 弱

Rule³: IF 室溫 is 高 AND 溫差 is 小 THEN 風速 is 弱

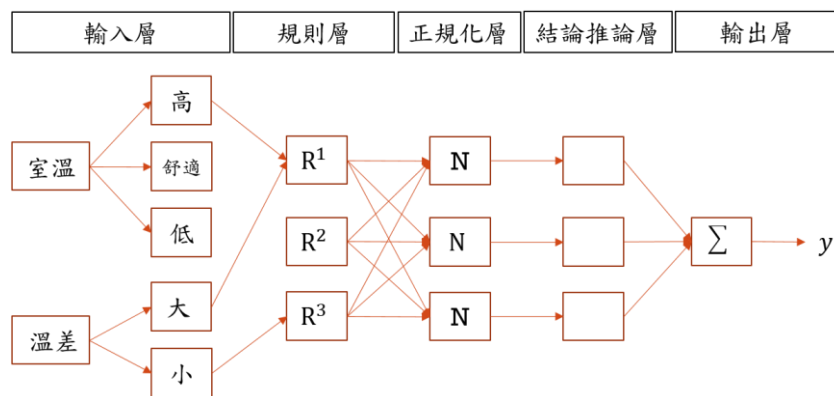


圖 6 模糊類神經網路圖

2.5 機器學習演算法

機器學習演算法對於模型扮演了重要的腳色，會直接的影響到模型的效能，因此使用何種機器學習演算法是非常重要的，常見到的機器學習演算法像是基因演算法（Genetic Algorithm, GA）[33]、蟻群最佳化（Ant Colony Optimization, ACO）[32]、粒子群最佳化（Particle Swarm Optimization, PSO）[7]與隨機最佳化（Random Optimization, RO）[8]等，粒子群最佳化藉由觀察鳥類族群覓食行為所啟發，概念上，一群鳥群在搜尋空間中搜尋食物，過程中，單一個體移動時會同時依據自身過去經驗與其他個體所提供的資訊進行比對，來修正下一次的行進方向，使鳥群能夠往離食物最近的個體周

遭移動，以找到食物。隨機最佳化透過迭代地隨機移動到搜索空間中更好的位置來運作，能夠在不需計算梯度的情況下進行問題的最優化，為一種簡單且易用的方法。蟻群最佳化，源自於螞蟻的行為模式，當螞蟻在搜尋食物時會在路徑上釋放費洛蒙，費洛蒙是一種利用化學分子傳遞訊息的溝通方式，其他螞蟻便藉由路徑上的費洛蒙濃度調整移動方向。基因演算法，便是根據生物演化而來。透過模擬基因染色體的複製、選擇、變異後，逐漸變化，較好的染色體便會被保留下來。而在類神經網路中，最常使用的演算法是倒傳遞法(Backpropagation)，以計算模型的性能指標函數之梯度，來決定搜索的方向以尋找最佳解。雖然倒傳遞法能夠在短時間計算出梯度找到解，但倒傳遞法並不容易找到最佳解並且易陷入局部最小化，因此在本篇論文將使用並行隨機最佳化與粒子群最佳化之最佳化演算法，並與遞迴最小平方估計法結合，用於更新本篇論文的複數模糊類神經網路模型之參數。

2.5.1 隨機最佳化演算法

隨機最佳化(RO)的初始版本是由 Matyas 於 1965 年提出[8]，是數值優化方法之一，可以在不需計算梯度的情況下進行問題的最優化，因此 RO 可用於不連續或不可微的函數。這種優化方法也被稱為直接搜索、無梯度或黑箱法。RO 透過迭代地移動到搜索空間中更好的位置來運作，並於搜索空間中使用常態分佈對當前位置週圍進行取點。RO 算法步驟如下：令 $f(x)$ 為愈最小化的目標函數， x 是目前考慮的位置。RO 演算法通過以下四個步驟來尋找最佳解：

步驟 1：選擇起始點 x 作為當前點。

步驟 2：使用常態分佈產生的隨機向量 dx 添加到參數空間中的當前點 x ，並在 $x + dx$ 的新點處評估目標函數。

步驟 3：如果 $f(x + dx) < f(x)$ ，將當前點 x 設置為 $x + dx$ 。

步驟 4：如果達到最大疊代次數(MaxIteration)，則停止。否則，返回步驟 2 尋找下一個新的點 x 。

當在 $\text{MaxIteration} = \infty$ 的假設下，此演算法能夠在緊湊集合(Compact set)上以機率 1 收斂到全域最小值[9][10][11]。於 1981 年，Solis 和 Wets 提出了改良的隨機最佳化演算法[9]，以少量步驟找到目標函數的全域最小值。改良後的隨機演算法步驟如下：

設 b 為常態分佈之中心

步驟 1：選擇起始點 x 作為當前點， $b = 0$ 。

步驟 2：使用常態分佈產生隨機向量 ξ

步驟 3：

(i) 如果 $f(x + \xi) < f(x)$ ，則將當前點 x 設置為 $x + \xi$ ，且更新 $b = 0.4\xi + 0.2b$

(ii) 如果 $f(x + \xi) > f(x)$ 且 $f(x - \xi) < f(x)$ ，則將當前點 x 設置為 $x - \xi$ ，並更新 $b = b - 0.4\xi$

(iii) 否則， $b = 0.5b$

步驟 4：如果達到最大疊代次數(MaxIteration)，則停止。否則，返回步驟 2 尋找下一個新的點 x 。

2.5.2 粒子群最佳化演算法

粒子群最佳化(PSO)，於西元 1995 年由 Kennedy 與 Eberhart 兩位學者提出[7]，從觀察鳥類族群的覓食行為中的訊息傳遞獲得的啟發，以一顆粒子代表鳥群中的單一個體，多粒子表示為鳥群，概念上，是一群鳥群在搜尋空間中搜尋食物，過程中，粒子移動時會同時依據自身過去經驗與其他個體所提供的資訊進行比對，來修正下一次的行進方向，使鳥群能夠往離食物最近的個體周遭移動，以找到食物。假設搜索空間是 J 維。粒子的速度(Velocity, V)和位置(Location, L)更新公式如下：

$$V_i(t+1) = \omega \cdot V_i(t) + c_1 \cdot \xi_1 \cdot (Pbest_i(t) - L_i(t)) + c_2 \cdot \xi_2 \cdot (Gbest(t) - L_i(t)) \quad (7)$$

$$L_i(t+1) = L_i(t) + V_i(t+1), \quad i = 1, 2, \dots, S_{swarm} \quad (8)$$

$V_i(t) = [v_{i,1}(t), v_{i,2}(t), \dots, v_{i,J}(t)]$ 是粒子 i 在時間 t 時的速度向量， c_1 與 c_2 為 PSO 的參數， ξ_1 與 ξ_2 是介於 $[0,1]$ 的隨機數， ω 是慣性權重； $L_i(t) = [l_{i,1}(t), l_{i,2}(t), \dots, l_{i,J}(t)]$ ，

粒子 i 在時間 t 時的位置向量； s_{swarm} 表示粒子群數量； $Pbest_i(t)$ 代表粒子 i 最好的位置； $Gbest(t)$ 代表粒子群中最好的位置。

粒子群最佳化學習步驟如下：

步驟 1：初始化粒子位置與速度。

步驟 2：以適應函數估計每顆粒子的適應值。

步驟 3：更新每顆粒子之適應值表現有比自身過去經驗中的 $Pbest$ 更好，則將 $Pbest$ 更新為粒子目前位置。

步驟 4：根據每顆粒子之 $Pbest$ 選出粒子群中最好位置 $Gbest$

步驟 5：依速度與位置更新公式，更新粒子。

步驟 6：檢查是否達到停止條件，是則停止；否則，返回步驟 2。

但是，粒子群最佳化中存在著一些因素會導致其表現不佳，因此在粒子群最佳化被提出後也出現了許多改良式的粒子群最佳化。2000 年，Kennedy 與 Clerc [31]由基本的數學與解析觀點，提出壓縮因子式粒子群最佳化演算法(Particles Swarm Optimization With Constriction Factor, PSO-CF)，有效控制各個粒子的搜尋軌跡。2001 年，Shi 與 Eberhart [28]以 PSO-CF 參數和其特性，調整 PSO-IW 中的慣性權重，並用於動態性系統問題的最佳化。2004 年，Chatterjee 與 Siarry [29]提出非線性動態慣性權重，依照不同世代的慣性權重關係選擇相對應的慣性權重，以增加粒子群的離散度。2006 年，Shu-Kai 與 Erwie [30]提出 PSO 結合單純形演算法概念，使粒子的移動模式取代 PSO 原始粒子的移動模式，來避免 PSO 落於區域最佳解。由文獻可發現會導致粒子群最佳化表現不佳的因素，其中之一的因素是粒子移動的活性，若粒子移動的活性不足，將易使得粒子群在最佳化的過程提早進入收斂並且陷入區域最佳解，產生早熟(Premature)的現象。因此，本論文提出了結合 PSO 與 RO 的演算法 RoPso，透過兩演算法之不同運作模式，提高單一使用 RO 或 PSO 之粒子移動變化，以降低粒子提早收斂，提升找到更佳解之機率，並且結合遞迴最平方估計法作為模型之機器學習最佳化演算法。

第三章 系統設計與架構

本章節將說明運用於研究中之方法設計與架構。本論文結合複數模糊集、Takagi-Sugeno 模糊系統與類神經網路，形成複數模糊類神經模型。並透過混合式機器學習演算法 RoPso-RLSE 來優化模型參數，且對資料進行特徵選取以提升學習成效與減少資料量對模型的負擔。

3.1 特徵選取

特徵選取的目標是要從原來的特徵中挑選出較有影響力的特徵。若不在模型學習前，先將資料進行特徵挑選，容易造成模型的運算負荷量較高，且也易使模型連同資料中的雜訊一起學習，使得模型後續預測的準確度受到影響。由於提出的計算模型為多目標型式，因此在特徵挑選時，本論文針對多目標情況設計特徵選取的方式，以便挑選出對所有目標有較多影響力的特徵。透過特徵挑選的前處理程序，將資料進行篩選，降低資料中雜訊函量，以選出有助於模型學習的資訊。研究中的特徵選取是採用過濾法，基於夏農資訊熵(Shannon Entropy) [21]以資訊密度的方式來估算資料集的亂度，資訊熵的計算定義如下：

$$H(Y) = \int_y p_d(y) \log\left(\frac{1}{p_d(y)}\right) dy \quad (9)$$

其中， Y 是事件之隨機變數； $H(Y)$ 為隨機變數 Y 之資訊熵； $p_d(y)$ 為事件 y 的機率密度。另外我們可以利用條件式資訊熵，來衡量若隨機變數 X 為已知的條件下隨機變數 Y 的資訊熵 $H_X(Y)$ ，定義如下：

$$H_X(Y) = \int_x \int_y p_d(x, y) \log\left(\frac{1}{p_d(y|x)}\right) dx dy \quad (10)$$

其中， X 是事件之隨機變數； $p_d(y|x)$ 為已知事件 x 發生的條件下事件 y 發生的機率密度； $p_d(x, y)$ 是事件 x 和事件 y 發生共同事件的機率密度。計算出已知隨機變數 X 情況下的隨機變數 Y 的資訊熵後，可透過(9)式減掉(10)式計算出隨機變數 X 與隨機變數 Y 的互資訊(Mutual information) [21]，互資訊代表兩個隨機變數間共有的訊息，在此代表隨機變數 X

與隨機變數 Y 之間共享的資訊，如(11)式

$$I(X, Y) = H(Y) - H_X(Y) \quad (11)$$

另外在(9)式與(10)式中需要先知道該隨機變數的機率密度函數(Probability density function, PDF)，接著再由機率密度函數進行取樣計算出該隨機變數的資訊熵，在研究中是利用核密度估計(Kernel density estimation)的方式，透過平滑的核函數(例如，高斯函數(Gaussian function))來擬合觀察到的數據，從而對真實的機率密度函數進行模擬並估算，核密度估計公式如下。

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right), \quad -\infty < x < \infty, \quad (12)$$

其中， n 為樣本大小； h 為平滑參數； $K(\cdot)$ 為是核函數(Kernel function)； x_i 為樣本； $i = 1, 2, \dots, I$ ； I 為樣本數量。

本論文模型在訓練前會將原始資料透過差分運算取得差量資料，在後續計算 $H(Y)$ 以及 $H_X(Y)$ 時，將前述之隨機變數 Y 令為目標變數，而隨機變數 X 為候選特徵變數且分作兩類：其一，在資料 x 為正值下對於目標 y 的事件；其二，在 x 為負值下對於目標 y 的事件，以這兩種情況計算出兩個互資訊量，最後將這兩個互資訊量分別乘上 x 資料機率分佈密度的面積比例再加總，即為候選特徵變數 X 對目標變數 Y 提供的資訊，如(13)及(14)式，稱為影響資訊(Influence information) [35]。

$$I_{X \rightarrow Y} = I(X^+, Y) \int_0^\infty p_d(x) dx + I(X^-, Y) \int_{-\infty}^0 p_d(x) dx \quad (13)$$

$$\begin{aligned} I(X^+, Y) &= H(Y) - H_{X^+}(Y) \\ I(X^-, Y) &= H(Y) - H_{X^-}(Y) \end{aligned} \quad (14)$$

其中， X^+ 表示在資料 x 是正值的條件下的隨機變數； $I(X^+, Y)$ ， X^+ 與 Y 的互資訊； X^- 表示在資料 x 是負值的條件下的隨機變數； $I(X^-, Y)$ 為 X^- 與 Y 的互資訊； $H(Y)$ 為 Y 本身的資訊亂度； $H_{X^+}(Y)$ 為在 X^+ 的條件下， Y 的資訊亂度(條件資訊熵)； $H_{X^-}(Y)$ 為在 X^- 的條件下， Y 的資訊亂度； $p_d(x)$ 為連續型事件 x 的機率密度。

以上述公式就可以計算出候選特徵變數 X 提供給目標變數 Y 的資訊量，亦可計算出不同候選特徵對於目標所提供的資訊量，以及計算出候選特徵彼此間的資訊量，由這些

候選特徵的資訊量及候選特徵提供給目標的資訊量整理出一個影響資訊矩陣(Influence information matrix, IIM) [35]。

	f_1	f_2	f_3	Target
f_1	0	0.37	0.41	0.54
f_2	0.42	0	0.22	0.25
f_3	0.58	0.53	0	0.31
Target	0.57	0.29	0.30	0

圖 7 資訊影響矩陣範例

由於特徵之間會有交互影響的關係，因此當同時要挑選多個特徵做為模型輸入時，除了需要計算特徵對於目標的影響資訊外，也必須考量到特徵之間的影响資訊。特徵之間的影响資訊在同時為模型輸入時，特徵間的影响資訊將可能轉變為冗餘資訊量，需要考慮到冗餘資訊量，將影响資訊排除掉冗餘資訊量後，以計算出特徵對目標真正提供的影響資訊。本論文中，特徵資訊量扣除冗餘資訊量後所提供給目標之影響資訊，稱其為增益量(Selection gain)。計算得影响資訊矩陣後，接著將說明如何選出對目標變數提供有明顯貢獻的特徵變數，來做為後續模型的輸入。假設一個候選特徵池 (Candidate-feature pool, CP)，定義為 $CP = \{f_i, i = 1, 2, \dots, |CP|\}$ ，其中， f_i 代表第幾個候選特徵變數， $|CP|$ 代表候選特徵池裡總共有幾個候選變數；已選特徵池 (Selected pool, SP)，定義為 $SP = \{s_i, i = 1, 2, \dots, |SP|\}$ ，其中， s_i 代表第幾個已選特徵變數。選取流程如下：

- Step:1.** 首次挑選時，會先從 CP 中挑出資訊量最大的特徵進入 SP 中，並將挑選出的特徵從候選特徵中移除。
- Step:2.** 第二次挑選時，從 CP 中選擇影响資訊量最大值的特徵，但由於 SP 中已經有存在特徵，因此需考慮冗餘資訊量 R ，冗餘資訊量計算方式，如(15)式。將特徵提供給目標的資訊量減掉冗餘資訊量就是特徵最後提供給目標的資訊增益量 G

[35]，如(16)式，若該特徵之增益量為正值，則選入 SP 中，並將該特徵從 CP 中移除。

$$R_{f \rightarrow SP} = \left(\frac{\sum_{k=1}^{|SP|} I(f \rightarrow s_k^{(SP)}) + I(s_k^{(SP)} \rightarrow f)}{2 \cdot |SP|} \right) \quad (15)$$

$$G(f \rightarrow t) = I_{f \rightarrow t} - R_{f \rightarrow SP} \quad (16)$$

其中， $R_{f \rightarrow SP}$ 為候選特徵 f 對 SP 中所有特徵的冗餘資訊量[35]； $|SP|$ 為 SP 中的特徵數量； $I(f \rightarrow s_k^{(SP)})$ 為候選特徵 f 對 SP 中所有特徵的影響資訊總量[35]； $I(s_k^{(SP)} \rightarrow f)$ 為 SP 中所有特徵對於候選特徵 f 的影響資訊總量[35]； $I_{f \rightarrow t}$ 為候選特徵 f 對於目標提供的影響資訊量[35]。

Step:3. 後續挑選皆如同 Step2，直到 CP 中特徵之值不再為正，挑選過程中每個放入 SP 中的特徵，所計算出來的增益量即為特徵真正提供給目標的資訊量，依照 SP 中的增益量之大小，來挑選將會做模型輸入的特徵。

由於本論文之模型為多目標預測，經過如上述之計算，取得各目標之 SP。後續分別針對出現於 SP 中的特徵進行對所有目標的增益值加總，並進行升冪排序後，從此排序由前面選擇特徵。最後，原始資料集經由特徵選取後，將產生訓練資料集(Training dataset, TD)，並運用於訓練模型。

$$TD = \{(\vec{x}, \vec{y})^{\{i\}}, i = 1, 2, \dots, |TD|\},$$

$$\vec{x} = [x_1 \ x_2 \ \dots \ x_M]^T, \quad (17)$$

$$\vec{y} = [y_1 \ y_2 \ \dots \ y_m]^T,$$

其中， $\vec{x}^{\{i\}}$ 為第 i 對資料集的輸入向量， $\vec{y}^{\{i\}}$ 為 $\vec{x}^{\{i\}}$ 所對應的目標向量； M 為輸入向量的維度； m 為實數型目標個數。

3.2 複數模糊類神經模型

本論文中，將運用複數模糊集合於模糊類神經模型中，由於複數模糊集合相較於傳統模糊集合，在其歸屬程度呈現在二維的複數單位圓盤之上，有實數部與虛數部的

值域，讓此歸屬程度擁有較多的描述空間，來容納更多的資訊。本論文之模型主要分作兩部分，分別為前鑑部與後鑑部。前鑑部負責將模型輸入透過複數模糊集合，轉換成歸屬程度，再依據模糊規則計算得啟動強度；後鑑部則是負責將啟動強度與每條規則進行計算，以取得最後模型輸出，此處の後鑑部採用 Takagi-Sugeno(T-S)模糊推論系統的型態。本論文將結合複數模糊集、模糊系統以及類神經網路模型，設計可用於多目標預測之複數類神經網路模型。模型說明如下：

設有一訓練資料集 $TD = \{(\vec{x}, \vec{y})^{\{i\}}, i = 1, 2, \dots, |TD|\}$ ， $\vec{x}^{\{i\}}$ 為第 i 對資料集的輸入向量， $\vec{y}^{\{i\}}$ 為 $\vec{x}^{\{i\}}$ 所對應的預測目標向量。假設研究模型由 K 條模糊規則所組成，則第 k 條模糊規則如下表示。

$$\begin{aligned} \text{Rule}^{(k)}: & \text{IF } x_1 = A_1^{(k)}(h_1) \text{ and } x_2 = A_2^{(k)}(h_2) \text{ and } \dots x_M = A_M^{(k)}(h_M) \\ & \text{THEN } y^{(k)} = a_0^{(k)} + \sum_{i=1}^M a_i^{(k)} h_i \text{ for } k = 1, 2, \dots, K \end{aligned} \quad (18)$$

其中， h_M 為第 M 個模型輸入； x_M 為第 M 個模型輸入的語意變數； $A_M^{(k)}$ 為第 k 個前鑑部的複數模糊集； $y^{(k)}$ 為第 k 條模糊規則的後鑑部， $\{a_0^{(k)}, a_1^{(k)}, \dots, a_M^{(k)}\}$ 為其後鑑部參數。本論文模型為多層架構的類神經網路模型，如圖 8 所示。

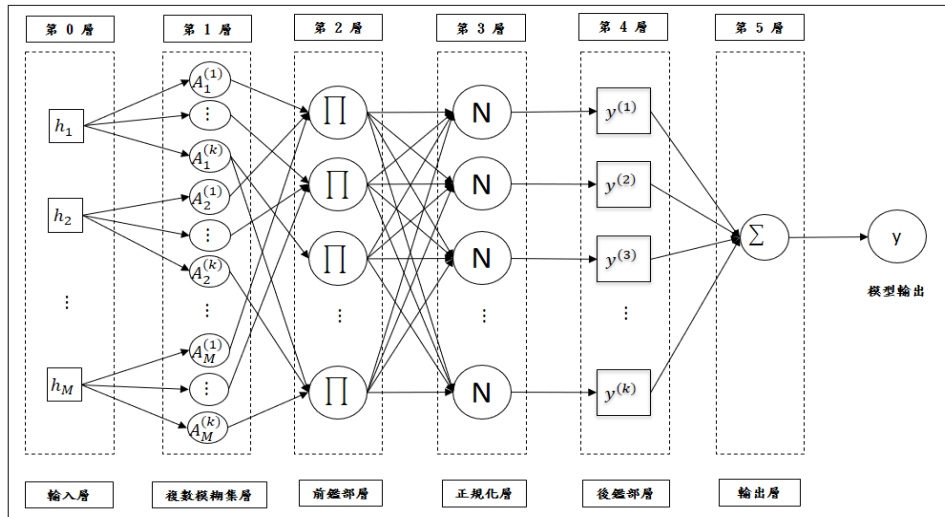


圖 8 複數模糊類神經系統模型

以下將分層進行說明。設研究模型具有 M 個實數型態輸入：

第 0 層：稱為輸入層(Input layer)，每一個神經元(以下泛稱為節點)都對應至一個數值變數，且該輸入將直接輸出至下一層，輸入變數之向量表示如下。

$$\vec{h} = [h_1, h_2, \dots, h_M]^T \quad (19)$$

第 1 層：稱為模糊集層(Fuzzy-set layer)，本層中每一個節點都對應至一個複數模糊集(CFS)，本論文在複數模糊集合的歸屬函數採用高斯型複數歸屬函數，其中 $\{m, \sigma, \lambda\}$ 是高斯型態 CFS 的參數集，為每一個節點的參數。經由歸屬函數計算可得知輸入資料屬於該複數模糊集合的歸屬程度 $\mu(h)$ ，歸屬程度又可以經由分解其值，得到三個不同的歸屬程度，進行多目標運算。複數型態歸屬程度表示如下。

$$\mu(h) = a + bi, \quad i = \sqrt{-1}; \quad a, b \in \mathbb{R} \quad (20)$$

將 $\mu(h)$ 分解出多個歸屬程度輸出，表示如下：

$$\begin{aligned} {}_1\mu(h) &= \mu(h) \\ {}_2\mu(h) &= \text{real}(\mu(h)) \\ {}_3\mu(h) &= \text{imag}(\mu(h)) \end{aligned} \quad (21)$$

其中， $\text{real}(\cdot)$ 是取實數； $\text{imag}(\cdot)$ 則是虛數。

第 2 層：稱為前鑑部層(Premise layer)，每一個節點代表一個前鑑部，負責進行積運算，每個節點將從前一層輸出的歸屬程度進行相乘得啟動強度，第 k 個節點表示為第 k 個前鑑部的啟動強度，表示為 $\vec{\beta}_{N \times 1}^{(k)}$ ，其中， $\vec{\beta}_{N \times 1}^{(k)}$ 的第 i 個成份表示如下。

$${}_i\beta^{(k)} = \prod_{j=1}^M {}_i\mu_j^{(k)}(h_j), i = 1, 2, 3. \quad (22)$$

其中 ${}_i\mu_j^{(k)}(h_j)$ 表示第 j 個輸入變量對應至第 j 個 CFS 的歸屬程度向量中第 i 個成份。

第 3 層：稱為正規化層(Normalization layer)，用於正規化前一層輸出的啟動強度值，

而第 k 個前鑑部正規化的值表示為 $\vec{\lambda}^{(k)}$ ，其中 $\vec{\lambda}^{(k)}$ 中的第 i 個成份表示如下：

$${}_i\lambda^{(k)} = \frac{{}_i\beta^{(k)}}{\sum_{k=1}^K {}_i\beta^{(k)}}, i = 1, 2, 3. \quad (23)$$

第 4 層：稱為後鑑部層(Consequence layer)，本層將計算每個後鑑部的結果 $\vec{y}^{(k)}$ ，第 k 個後鑑部之第 i 個成分，表示如下。

$$i\hat{y}^{(k)} = i\lambda^{(k)} \cdot \left(a_0^{(k)} + \sum_{j=1}^M a_j^{(k)} h_j \right) \quad (24)$$

其中， $a_j^{(k)}$ 為第 k 個後鑑部參數。

第 5 層：稱為輸出層(Output layer)，本層將加總第 4 層輸出的所有結果，來得到模型的輸出，表示如下。

$$\vec{\hat{y}} = \sum_{k=1}^K \vec{\hat{y}}^{(k)} \quad (25)$$

當模型計算出最終輸出後，可以利用模型的輸出與目標之間的誤差計算出性能指標，性能指標是用來評估模型輸出的性能，並讓機器學習演算法根據性能指標來調整模型內部的參數，本論文中是使用均方根誤差(Root-mean-square error, RMSE)來做為性能指標，其定義如下：

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n \vec{e}^{(i)*} \cdot \vec{e}^{(i)}}{n}}, \quad \vec{e}^{(i)} = \vec{y}^{(i)} - \vec{\hat{y}}^{(i)} \quad (26)$$

其中* 為共軛轉置運算； $\vec{e}^{(i)}$ 為模型輸出 $\vec{y}^{(i)}$ 與目標值 $\vec{\hat{y}}^{(i)}$ 之誤差。

3.3 混合式機器學習演算法

在本篇論文中所使用到的機器學習法是利用粒子群最佳化與隨機最佳化的並行運作，產生之最佳化演算法(RoPso)，並且結合遞迴最小平方估計法(RLSE)。並行式演算法 RoPso 主要是用來調整前鑑部的參數，而遞迴最小平方估計法則是依據 RoPso 演算法調整後前鑑部的輸出來調整後鑑部的參數。

3.3.1 並行式最佳化 RoPso

由 2.5.1 隨機最佳化演算法與 2.5.2 粒子群最佳化演算法可知，隨機最佳化與粒子群最佳化，都有各自的特點。隨機最佳化(RO)，對於任何連續的目標函數作修正，無

論初始位置於何處，理論上能以機率 1 來收斂到全域最小值。除其收斂性外，RO 特色為其隨機性，且整體計算流程快速、簡單，使用上具有易於應用的特性。粒子群最佳化(PSO)，具有篤定性與自動調整步伐的特色。篤定性：由於粒子群中的個體本身會記憶歷史移動過程中最好之位置，稱之 Pbest，且當進行方向修正時，還會參考群體粒子中最佳移動經驗，稱之 Gbest，因此更新行進方向時，將帶領粒子群往較佳的解前進。自動調整步伐：每顆粒子在移動時，會同時參考三個元素，分別為當前行進方向向量、該顆粒子之 Pbest 的方向向量以及群體 Gbest 的方向向量，透過每次疊代中改變 ω 、 c_1 與 c_2 ，來調整粒子移動的步伐，使粒子在移動時保有彈性。由於上述特性，因此，本論文提出透過並行 RO 與 PSO 兩者的運作方式，融入競爭與學習的機制，將此兩種演算法進行結合，藉由結合 RO 與 PSO 的特性，以產生一個運算快速、兼具隨機性與方向性，提升找到全域解機率之演算法。以下針對競爭與學習機制以及如何並行運作之方面進行說明。

競爭與學習機制

RO 與 PSO 兩演算法透過並行的方式運作，於兩者獨立的運作過程中，各自都將產生所屬的最佳解，稱之 Gbest。當每次疊代中，RO 與 PSO 的 Gbest 皆取得之時，將兩演算法之該次疊代最佳解進行比對，並將表現較佳者視為整體之最佳解代表，此為競爭機制。當取得整體最佳解後，將其回傳至 RO 與 PSO 中取代兩者各自的最佳解，表示後續的粒子將以兩者表現較佳之最佳解位置作為參考來進行移動，為學習機制。透過競爭與學習的機制，若當 PSO 之 Gbest 較佳時，將影響 RO 起始點位置之決定，使其下次進行粒子取點時，會以較佳之位置為基準，讓 RO 的移動中加入方向性的元素；若為 RO 之 Gbest 較佳時，PSO 中之 Gbest 將被取代，意即 PSO 中的粒子移動方向將進行調整，如圖 9 所示。讓 PSO 中粒子的移動時，減少受到粒子本身過去之經驗與群體之經驗的影響力，提升 PSO 粒子移動的隨機性。因此，此一並行式演算法 RoPso，整體而言，將同時具備方向性與隨機性，藉由提升粒子移動的彈性與變化性，以提升全域解搜尋之機率。

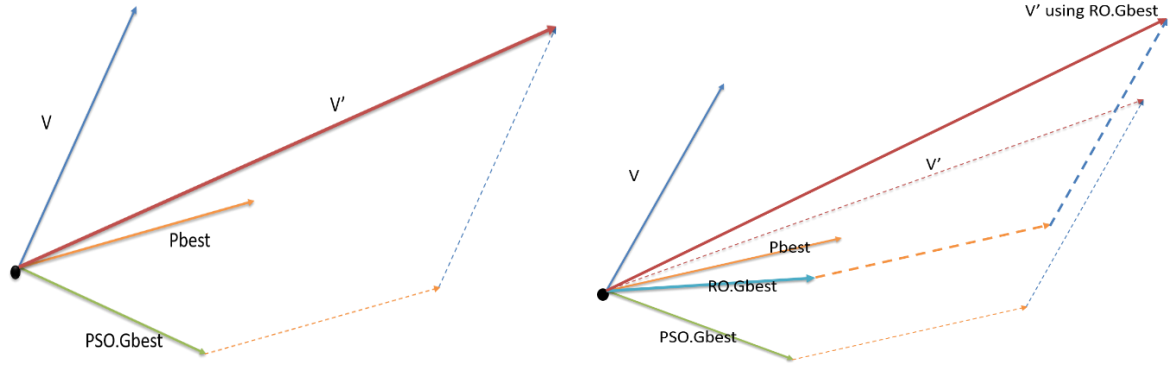


圖 9 PSO 之粒子移動變化比較示意圖。

左圖為參考 PSO 之 Gbest 進行粒子移動示意圖；右圖為 PSO 中以 RO 之 Gbest 為參考時粒子移動示意圖。

RoPso 演算法其運作方式如下(如圖 10 所示):

步驟一：粒子初始化，PSO 進行粒子位置與速度的初始化，而 RO 則是初始化一個位置 x 作為起始點(start point)，並令常態分佈之中心參數 $b = 0$ 。

步驟二：在 RO 的起始點為中心，以常態分佈生成多組隨機向量 $\xi^{(i)}$ ， i 表示為第 i 組隨機向量。分別以下列方式產生 RO 的新位置(Position)置：

- (i) $P_i(k+1) = \text{start point}(k) + \xi^{(i)}, i = 1, 2, \dots, \left\lfloor \frac{s_{\text{swarm}}}{2} \right\rfloor$
- (ii) $P_i(k+1) = \text{start point}(k) - \xi^{(i)}, i = \left\lfloor \frac{s_{\text{swarm}}}{2} \right\rfloor + 1, \left\lfloor \frac{s_{\text{swarm}}}{2} \right\rfloor + 2, \dots, s_{\text{swarm}}$ (27)

$$\xi = N(b, \sigma), \sigma = \text{Min}(D) + (\text{Max}(D) - \text{Min}(D)) * \text{random}(0,1)$$

其中， $\text{start point}(k)$ 為在時間 k 時的起始點； s_{swarm} 為 RO 之粒子數； D 為資料集。

步驟三：利用適應函數評估 RO 與 PSO 每顆粒子的適應值。

步驟四：藉由每顆粒子的適應值，RO 從所屬的粒子中，挑選出本次疊代中適應值最好的粒子作為 RO 的 Gbest (RO.Gbest)，且中心參數 b 更新如下：

設 $f(\cdot)$ 為適應函數

- (i) 若 RO.Gbest 屬於(27)中第(i)部分之粒子位置，
則更新 $b = 0.4\xi^{(i)} + 0.2b$ (28)
- (ii) 若 RO.Gbest 屬於(27)中第(ii)部分之粒子位置，
則更新 $b = b - 0.4\xi^{(i)}$

(iii) 否則， $b = 0.5b$

PSO 同理，從所屬的粒子中，挑選出適應值最好的粒子作為 PSO 的 Gbest

(PSO.Gbest)，且每顆粒子也會選出該顆粒子移動路徑中，最好的一次位置作為 Pbest (PSO.Pbest_{*i*}, $i = 1, 2, \dots, s_{swarm}$)。

步驟五：比對 RO 與 PSO 的 Gbest 的適應值，何者表現較佳。

步驟六：選擇較佳的 Gbest 做為 RoPso.Gbest，並回傳至 PSO 與 RO 中取代兩者之 Gbest，使其採用 RoPso.Gbest 做為學習參考

步驟七：RO 以 RoPso.Gbest 去更新的起始點，而 PSO 則依公式(7)來更新粒子速度，並依公式(8)更新粒子位置。

步驟八：依據學習疊代次數或門檻值條件，若達成，則結束學習將 RoPso.Gbest 作為最佳解，否則，回到步驟二繼續學習。

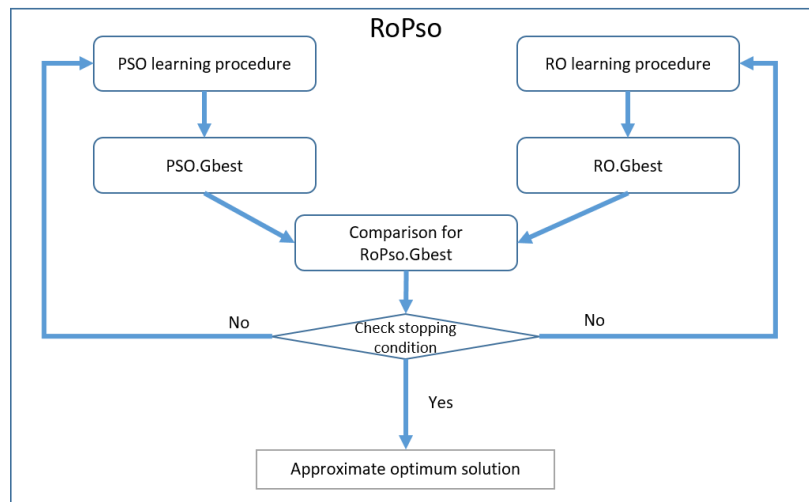


圖 10 RoPso 流程圖。

3.3.2 遞迴最小平方估計法

遞迴最小平方估計法[26][27]演算法是由最小平方估計法（LSE）加入遞迴的概念而來。最小平方估計法的計算概念則是從已知的訓練資料中以數學計算式直接找出一條最佳的方程式解，以平方誤差求解找到訓練資料的最佳近似解。LSE 適用於線性的參數表示，如下所示：

$$y = f_1(u)\theta_1 + f_2(u)\theta_2 + \cdots + f_m(u)\theta_m + \varepsilon \quad (29)$$

其中， u 是模型的輸入； y 為目標； $\{f_i(u), i = 1, 2, \dots, m\}$ 為 u 的函數； $\{\theta_i, i = 1, 2, \dots, m\}$ 是期望被估計的模型參數，而 θ 可視為模型中的後鑑部參數。

假設 TD 為一個 n 組的資料集，表示為 $TD = \{(\vec{h}^{(i)}, \vec{y}^{(i)}) , i = 1, 2, \dots, n\}$ ，其中 \vec{h} 為輸入向量； \vec{y} 為目標向量。將每一組資料帶入(29)中，以矩陣形式表示結果如下。

$$Y = B\theta + \varepsilon \quad (30)$$

Y 為模型輸出矩陣， B 為輸入矩陣， θ 為尋找的參數矩陣， E 為誤差矩陣。

$$Y = [y_1 \quad \cdots \quad y_n]^T, \quad (31)$$

$$B = \begin{bmatrix} f_1(h_1) & \cdots & f_m(h_1) \\ \vdots & \cdots & \vdots \\ f_1(h_n) & \cdots & f_m(h_n) \end{bmatrix}, \quad (32)$$

$$\theta = [\theta_1 \quad \cdots \quad \theta_n]^T, \quad (33)$$

$$\varepsilon = [\varepsilon_1 \quad \cdots \quad \varepsilon_n]^T \quad (34)$$

為取得最小化 $\|\varepsilon^2\|$ ， θ 可利用最小平方估計法取得，如下所示。

$$\theta = (B^T B)^{-1} B^T Y \quad (35)$$

由於有新資料產生時，採用最小平方估計法則需將所有資料重新計算一次，考量計算量問題，因此本論文中藉由遞迴式最小平方估計法(RLSE)來計算參數 θ ，透過以遞迴方式循序漸進計算，利用前一次估計出的結果，加入當次輸入以及輸出值，來計算當次的估計結果，此方式可以降低運算量以及當有新資料時可根據新資料更新參數。在遞迴次數為 r 時，參數矩陣為 θ_r ，遞迴最小平方估計法定義如下。

$$P_{r+1} = P_r - \frac{P_r b_{r+1}}{1 + b_{r+1}^T P_r b_{r+1}} b_{r+1}^T P_r \quad (36)$$

$$\theta_{r+1} = \theta_r + P_{r+1} b_{r+1} (y_{r+1} - b_{r+1}^T \theta_r) \quad (37)$$

其中， $r = 0, 1, 2, \dots, (n-1)$ ， $[b_r^T, y_r]$ 代表矩陣 $[B, Y]$ 中的第 r 列； θ_r 是第 r 次遞迴計算出的估計參數值； $P(r)$ 為 $\theta(r)$ 的增益矩陣。在使用遞迴式最小平方估計法公式前，需先將 P_0 與 θ_0 進行初始化。 θ_0 初始化為零向量， P_0 的初始化定義如下：

$$P_0 = \alpha I, \quad (38)$$

其中， α 是個極大數； I 則為單位矩陣。

由於原始 RLSE 公式是適用於單值輸出，為配合本論文中所提出之模型為多輸出型式，需將(36)式中分母的單值 1 改寫為單位矩陣 I ，以便能計算出多輸出之值。公式改寫如下所示：

$$P_{r+1} = P_r - \frac{P_r b_{r+1}}{I + b_{r+1}^T P_r b_{r+1}} b_{r+1}^T P_r \quad (39)$$

3.4 研究整體流程

本論文模型架構如圖 11。模型建構完成後，將進行模型參數的優化，本論文中運用前鑑部與後鑑部參數分開優化的方式，可減少模型訓練時間以提升效率。採用混合式機器學習演算法 RoPso-RLSE 更新模型參數，由 RoPso 負責優化前鑑部參數，後鑑部參數則由 RLSE 進行更新。藉由將模型輸出與目標值計算取得誤差值並轉換成誤差指標，使計算模型根據誤差指標之大小來調整模型參數，RoPso-RLSE 演算法流程如下：

Step:1. 將原始資料，經由特徵挑選後取得模型輸入資料集，並將此資料集分為兩部分，一為訓練資料集(TD)，另為則是測試資料集。

Step:2. 初始化 RoPso 之參數，其中參數包含 $\{m, \sigma, \lambda\}$ 。初始化 RO 的初始點位置與 PSO 粒子的初始位置和速度，初始位置由輸入資料的最大與最小值為範圍來隨機生成。第 i 顆粒子初始化位置如下所示。

$$\text{particle}_{\text{position}}^{(i)} = [\rho \quad \rho \quad \dots \quad \rho], \quad (40)$$

$$\rho = \text{Min}(TD) + (\text{Max}(TD) - \text{Min}(TD)) * \text{random}(0,1)$$

Step:3. 每顆粒子都視作一個模型，針對每個模型計算訓練資料輸入模型後，對每條規則所產生出的啟動強度 $\vec{\lambda}_{N \times 1}^{(k)}$ 。

Step:4. 利用遞迴式最小平方估計法更新模型後鑑部參數，根據(36)與(37)式，其中

$$\mathbf{b}_{r+1} = [bb^{(1)}(r+1) \ bb^{(2)}(r+1) \ \dots \ bb^{(k)}(r+1)]^T, k = 1, 2, \dots, K \quad (41)$$

$$bb^{(k)}(r+1) = \begin{bmatrix} {}_1\lambda^{(k)} & {}_1\lambda^{(k)}h_1(r+1) & \dots & {}_1\lambda^{(k)}h_M(r+1) \\ \vdots & \vdots & \ddots & \vdots \\ {}_i\lambda^{(k)} & {}_i\lambda^{(k)}h_1(r+1) & \dots & {}_i\lambda^{(k)}h_M(r+1) \end{bmatrix}, i = 1, 2, 3 \quad (42)$$

當中 k 表示為規則， ${}_i\lambda^{(k)}$ 是第 k 條規則第 i 個成分之正規化啟動強度，經過(39)式計算得 θ ，即後鑑部參數。

Step:5. 計算模型輸出 $\hat{\mathbf{y}}$ ，將 $\hat{\mathbf{y}}$ 與目標值 \mathbf{y} 進行誤差運算

$$\bar{e}^{\{i\}} = \mathbf{y}^{\{i\}} - \hat{\mathbf{y}}^{\{i\}}, i = 1, 2, \dots, |\text{TD}| \quad (43)$$

Step:6. 利用誤差值 $\bar{e}^{\{i\}}$ 計算均方根誤差(Root-mean-square error, RMSE)，RMSE 計算如下。

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{|\text{TD}|} \bar{e}^{\{i\}*} \cdot \bar{e}^{\{i\}}}{|\text{TD}|}} \quad (44)$$

其中， $e(t)^*$ 為 $e(t)$ 的共軛轉置。

Step:7. RO 與 PSO 先各自比對自己群中之粒子，以 RMSE 最低的粒子當作 Gbest，即更新 RO 中的 RO.Gbest 與 PSO 中的 PSO.Pbest_{*i*}及 PSO.Gbest。

Step:8. 比較 RO 與 PSO 的 Gbest 何者較佳，選擇較佳者來當作 RoPso.Gbest，並回傳至 RO 與 PSO 中取代各自之 RO.Gbest 與 PSO.Gbest。

Step:9. PSO 根據更新速度(7)與位置公式(8)來更新各粒子; RO 將 RO.Gbest 取代其 start point，並根據(27)(28)式更新粒子位置。

Step:10. 判斷終止條件是否符合，若尚未達到學習迭代次數，則回到 Step3；若已完成，則取出 RoPso.Gbest，此粒子位置即為複數模糊類神經模型優化後的前鑑部參數。

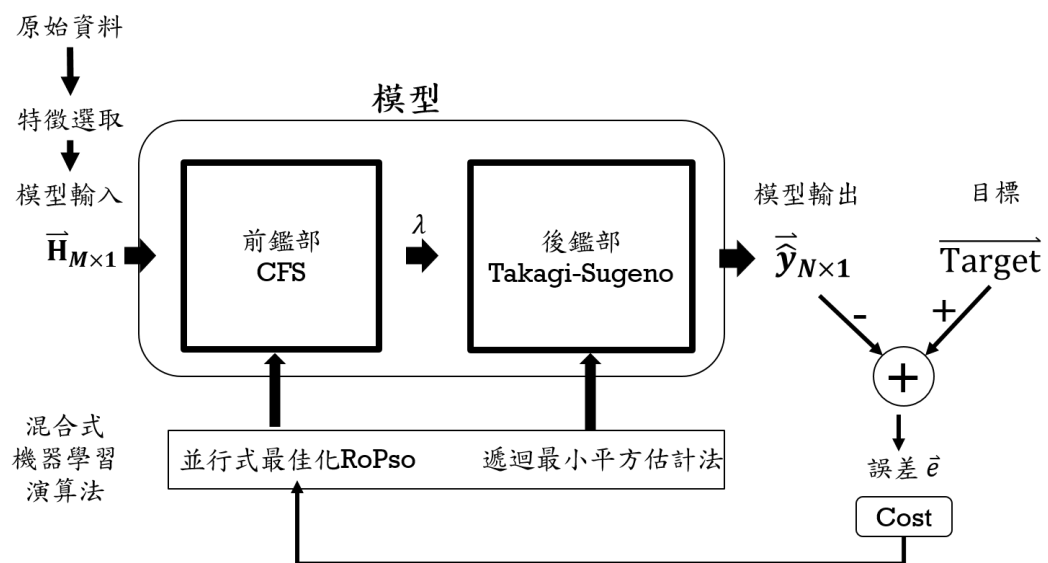


圖 11 研究模型架構圖

第四章 實驗實作與結果

4.1 實驗 1: 雙目標股票指數預測-

在實驗 1 中，為測試模型雙目標輸出的表現能力，實驗目標分別使用 Intel 和 Microsoft 兩間公司的每日股價，資料來源取自 <https://finance.yahoo.com/>，資料蒐集範圍為在 2007 年 9 月 12 日至 2010 年 11 月 11 日期間的每日收盤價，共 800 筆。為了訓練研究模型，我們將資料集分成訓練和測試兩組資料集，使用 2007 年 9 月 12 日至 2009 年 9 月 14 日(共 506 筆)進行訓練，並於 2009 年 9 月 15 日至 2010 年 11 月 11 日(共 294 筆)進行測試。在原始資料集的前處理時，會將資料集處理成差分資料(漲跌)，接著會將 Intel 和 Microsoft 的資料集前 30 天的差分資料做為候選特徵 $\{f_i, i = 1, 2, \dots, 30\}$ ，兩個資料集共為 60 個候選特徵變數(f_1 至 f_{30} 屬於 Intel; f_{31} 至 f_{60} 屬於 Microsoft)，其中， i 值越大表示越接近目標。在實驗中，將使用一個複數型態的目標變數來表示兩個實數型態目標，分別通過實部和虛部代表第一實數目標(Intel)和第二實數目標(Microsoft)後續在經由本論文中的特徵選擇後，將選出對目標提供較多資訊量之特徵作為模型輸入，以進行訓練與測試，並將實驗重複進行十次，最後將實驗結果與其它文獻方法比較。

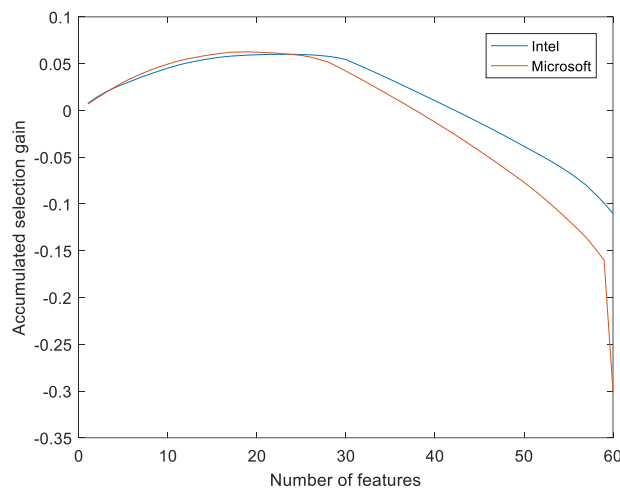


圖 12 特徵對目標 Intel 與 Microsoft 所提供之累積增益量

圖 12 為特徵提供給目標的累積增益量，經由特徵挑選後，由 $\{f_{57}, f_{13}, f_{17}, f_{54}, f_{27}, f_{21}\}$ 被選用作模型的輸入， $f_{13}, f_{17}, f_{21}, f_{27}$ 分別為 Intel 之第 13、17、21 與第 23 個特徵； f_{57}, f_{54} 為 Microsoft 之第 27 與第 24 個特徵，形成之訓練資料(TD)表示為 $TD = \{(\vec{x}, \vec{y})^{(i)}, i = 1, 2, \dots, |TD|\}$ ，其中， $\vec{x}^{(i)} = [f_{57}, f_{13}, f_{17}, f_{54}, f_{27}, f_{21}]^T$ ， $\vec{y}^{(i)}$ 則為對應目標值。於實驗中建構模型時，將每個模型輸入採用減法分群演算法(subtractive clustering algorithm)進行分群來取得群數，每個分群即代表一個高斯複數模糊集。本實驗中，每個模型輸入皆分作三群，即每個輸入皆有三個模糊集，將產生 729 個前鑑部 (同條模糊規則數)。由於考量研究模型計算效率，因此在實驗中，將透過篩選出前 15 個前鑑部，以資料導向(Data-oriented)的方式決定資料較為密集的前鑑部作為候選者以提升運算效率，挑選過程如下。

Step:1. 將輸入資料 $\{\vec{x}^{(i)}, i = 1, 2, \dots, |TD|\}$ 重新帶回輸入空間，透過高斯函數計算與累計每個資料點在各前鑑部上產生的累積數量強度值。

Step:2. 加總所有前鑑部的累積數量強度值，取其平均與標準差，並以平均值與一倍標準差之和，作為挑選門檻(Threshold)。選出所有大於門檻值的候選前鑑部，並進行排序。最後，選取前 15 個候選前鑑部作為模型的前鑑部。

經由挑選後，本實驗每個模型輸入皆留下三個高斯複數模糊集，且共有 15 條模糊規則，下表為模型相關配置。

表格 1 模型參數配置(實驗一)

Model Inputs	$f_{57}, f_{13}, f_{17}, f_{54}, f_{27}, f_{21}$
Number of Gaussian complex fuzzy sets (GCFSs) of each input	3,3,3,3,3,3
Number of GSCFS premises	15
Number of Takagi-Sugeno consequences	15
Complex -valued target(s)*	1
Model output(s)	1

實驗中採用 RoPso-RLSE 的複合式機器學習演算法，於 RoPso 的設定中，RO 與 PSO 各別皆使用 8 顆粒子，每顆粒子皆為 54 個維度，粒子的維度表示為模型的前鑑部參數。學習總迭代次數為 50 次，其中 RO 與 PSO 的初始位置採用輸入資料的最大與最小值為範圍來隨機生成。詳細學習演算法相關設定如表格 2。

表格 2 RoPso-RLSE 學習演算法參數設定(實驗一)

Iterations	50
Number of parameter in premises	54
Number of parameter in consequences	105
PSO setting	
Size of swarm	8
Particle dimensions	54
ω	Random in [0.6,0.95]
c_1 、 c_2	Random in [1.8,2.2]
ξ_1 、 ξ_2	Random in [0,1]
RO setting	
Size of swarm	8
Particle dimensions	54
New random vectors generation	$N(b, \sigma)^{54}$
RLSE setting	
Number of parameter in consequence	105
P_0	$\alpha \cdot I$
α	1×10^{10}
I	105×105 identity matrix

本實驗總共進行 10 次試驗，圖 13 為 10 次中表現最佳之學習曲線，圖 14 與圖 16 為模型預測結果，圖 15 與圖 17 為模型預測結果與目標之誤差。為比較本論文提出之混合式學習演算法的效能，因此於表格 3 中分別列出 RO-RLSE、PSO-RLSE、RoPso-RLSE 以及其他文獻方法之比較。其中，RO-RLSE 與 PSO-RLSE 之粒子顆數設定為 RoPso -RLSE 粒子顆數之總和。

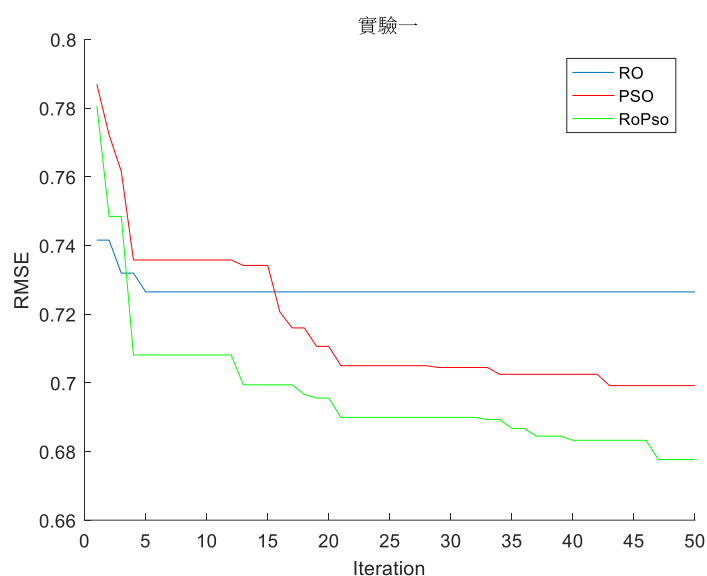


圖 13 模型學習曲線 (實驗一)

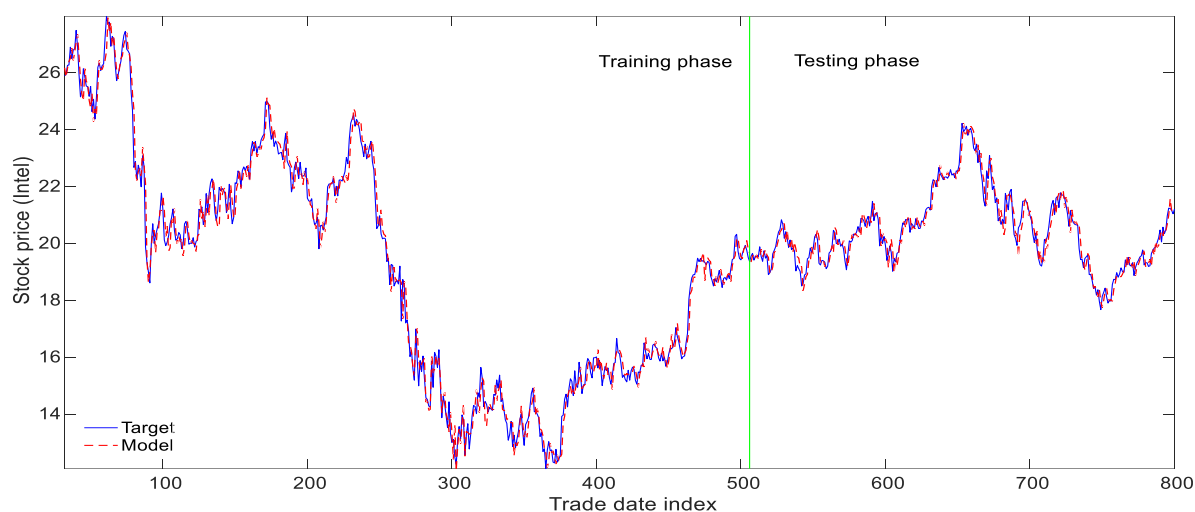


圖 14 模型訓練與預測結果-Intel

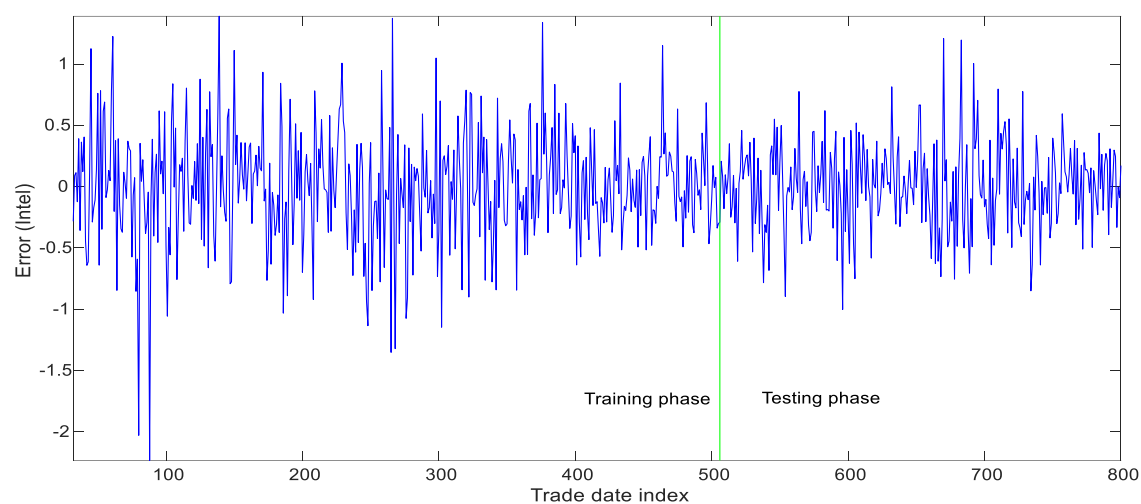


圖 15 訓練結果與目標之誤差- Intel

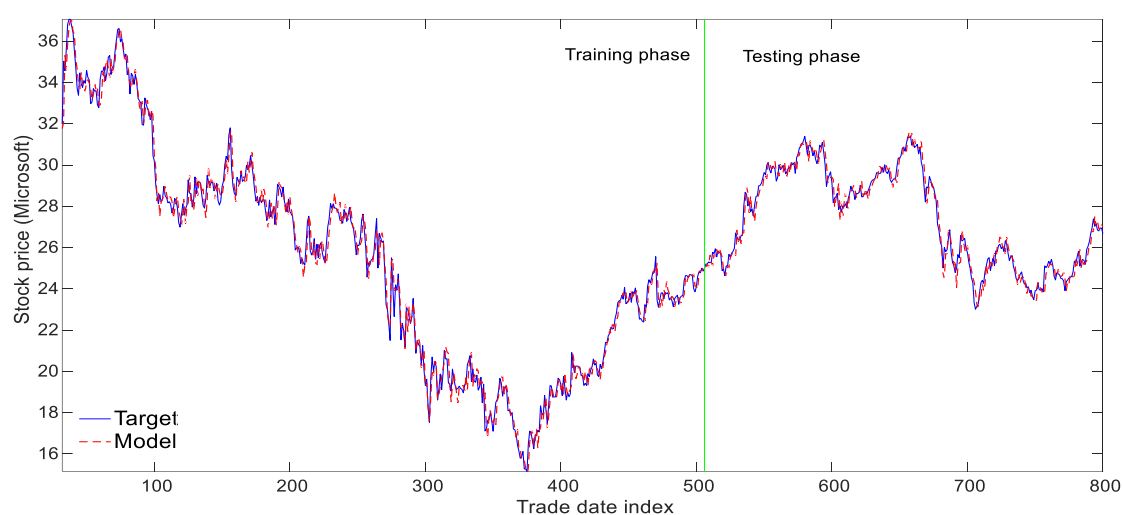


圖 16 模型訓練與預測結果-Microsoft

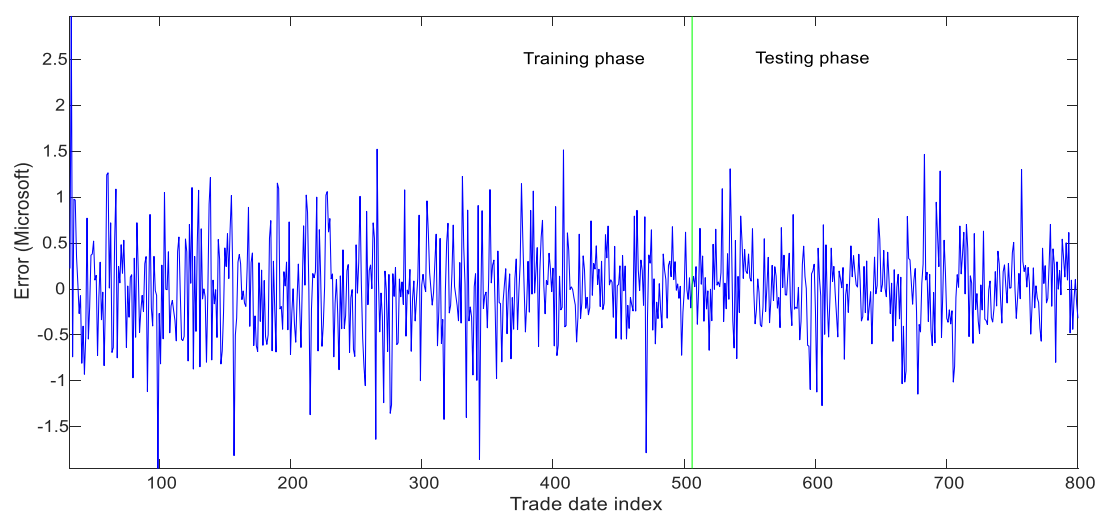


圖 17 訓練結果與目標之誤差-Microsoft

表格 3 與其他文獻之比較(實驗一)

Method	MAPE	
	Intel	Microsoft
SVR-GA [34]	0.04738	0.066795
SVR-CGA [34]	0.04670	0.061031
SVR-FA [34]	0.04562	0.052653
SVR-CFA [34]	0.04459	0.051907
ANN [34]	0.04708	0.054386
ANFIS [34]	0.04635	0.056344
Proposed approach(RO-RLSE) (testing phase)	0.0180	0.0146
Proposed approach(PSO-RLSE) (testing phase)	0.0164	0.0130
Proposed approach(RoPso-RLSE) (testing phase)	0.0144	0.0121

表格 4 10 次實驗之統計結果(實驗一)

# trials:10	RMSE					
	Intel			Microsoft		
	RO-RLSE	PSO-RLSE	RoPso-RLSE	RO-RLSE	PSO-RLSE	RoPso-RLSE
Best	0.4137	0.4947	0.3687	0.5097	0.5077	0.4257
Worst	23.5417	6.8002	5.8384	12.8525	3.9464	2.7567
Average	3.7755	2.3054	1.0028	2.3448	1.5703	0.7356
Std	7.3925	2.1987	1.7137	4.0327	0.9815	0.7222
MAPE						
Best	0.0150	0.0157	0.0137	0.0117	0.0130	0.0121
Worst	0.3312	0.0350	0.0492	0.1779	0.0270	0.0258
Average	0.0527	0.0239	0.0190	0.0323	0.0184	0.0146
Std	0.0985	0.0078	0.0113	0.0515	0.0041	0.0045

4.2 實驗 2: 三目標股票指數預測-

在實驗 2 中，將測試模型針對三個目標之效能表現，三個目標分別使用 British Airlines、Southwest Airlines 和 Ryanair Holdings Ltd 在 2003 年 5 月 6 日至 2005 年 3 月 17 日的每日股價，共 471 筆，資料蒐集來源取自 <https://finance.yahoo.com/>。為了訓練研究模型，我們將資料集分成訓練和測試兩組資料集，使用 2003 年 5 月 6 日至 2004 年 7 月 23 日對模型進行訓練(共 307 筆)，並於 2004 年 7 月 26 日至 2005 年 3 月 17 日進行測試(共 164 筆)。在原始資料集的前處理時，會將資料集處理成差分資料(漲跌)，接著會將三目標的資料集之前 30 天的差分資料做為候選特徵 $\{f_i, i = 1, 2, \dots, 30\}$ ，三個資料集共為 90 個候選特徵變數(f_1 至 f_{30} 屬於 British Airlines, f_{31} 至 f_{60} 屬於 Ryanair Holdings Ltd, f_{61} 至 f_{90} 屬於 Southwest Airlines) 其中， i 值越大表示越接近目標。在實驗 2 中，有三個目標，因此使用兩個複數型態的目標變數來表達。第一複數目標分別通過其實部和虛部處理第一實數目標(British Airlines)和第二實數目標(Ryanair Holdings Ltd); 第二個複數目標的實部則是第三個實數目標(Southwest Airlines)，虛部未被使用。後續經特徵選擇後，將選出對目標提供較多資訊量之特徵作為模型輸入，以進行訓練與測試，並將實驗重複進行十次，最後將實驗結果與其它文獻方法比較。

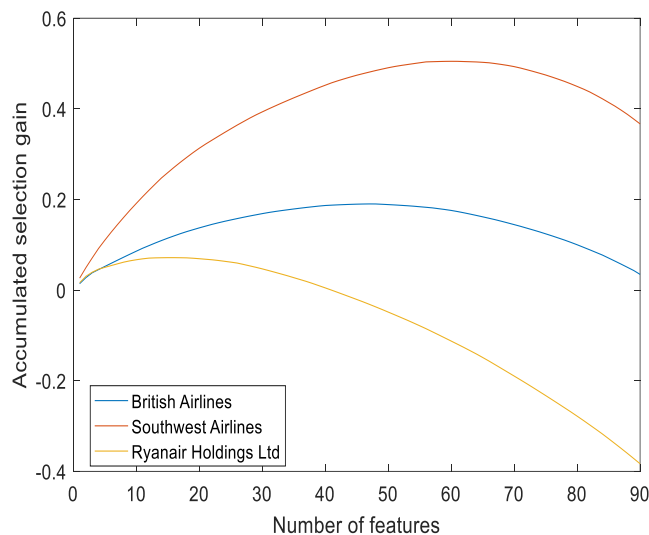


圖 18 特徵對目標 British Airlines、Southwest Airlines 與 Ryanair Holdings Ltd 所提供之累積增益量(實驗二)

圖 18 為特徵提供給目標的累積增益量，經由特徵挑選後，由 $\{f_{30}, f_{18}, f_{61}, f_{47}, f_{83}, f_{40}\}$ 被選用作模型的輸入， f_{18} 與 f_{30} 為 British Airlines 第 18 與第 30 個特徵； f_{40} 與 f_{47} 分別為 Ryanair Holdings Ltd 第 10 與第 17 個特徵； f_{61} 與 f_{83} 則分別為 Southwest Airlines 第 1 與第 23 個特徵，形成之訓練資料(TD)表示為 $TD = \{(\vec{x}, \vec{y})^{[i]}, i = 1, 2, \dots, |TD|\}$ ，其中， $\vec{x}^{[i]} = [f_{30}, f_{18}, f_{61}, f_{47}, f_{83}, f_{40}]^T$ ， $\vec{y}^{[i]}$ 則為對應目標值。於實驗中建構模型時，將每個模型輸入採用減法分群演算法(subtractive clustering algorithm)進行分群取得群數，每個分群即代表一個高斯複數模糊集。本實驗中，每個模型輸入分別分作 5、4、4、4、4 與 2 群，將產生 2560 個前鑑部 (同條模糊規則數)。由於考量研究模型計算效率，因此在實驗中，將透過篩選出前 15 個前鑑部，選取方式同時驗 1。經由挑選後，本實驗每個模型輸入分別留下 2、2、3、2、2 與 1 個高斯複數模糊集，且共有 15 條模糊規則，下表為模型相關配置。

表格 5 模型參數配置(實驗二)

Model Inputs	$f_{30}, f_{18}, f_{61}, f_{47}, f_{83}, f_{40}$
Number of Gaussian complex fuzzy sets (GCFSSs) of each input	2、2、3、2、2、1
Number of GSCFS premises	15
Number of Takagi-Sugeno consequences	15
Complex -valued target(s)*	2
Model output(s)	2

實驗中採用 RoPso-RLSE 的複合式機器學習演算法，於 RoPso 的設定中，RO 與 PSO 使用 8 顆粒子，每顆粒子皆 36 個維度，粒子的維度表示為模型的前鑑部參數。學習總迭代次數為 50 次，其中 RO 與 PSO 的初始位置同實驗一之方式來隨機生成。詳細學習演算法相關設定如表格 6。本實驗總共進行 10 次試驗，圖 19 圖 13 為 10 次中表現最佳之學習曲線，圖 20、圖 22 與圖 24 為模型預測結果，圖 21、圖 23 與圖 25

為模型預測結果與目標之誤差。為比較本論文提出之混合式學習演算法的效能，因此於表格 7 中分別列出 RO-RLSE、PSO-RLSE、ROPSO-RLSE 以及其他文獻方法之比較。

表格 6 RoPso-RLSE 學習演算法參數設定(實驗二)

Iterations	50
Number of parameter in premises	36
Number of parameter in consequences	105
PSO setting	
Size of swarm	8
Particle dimensions	36
ω	Random in [0.6,0.95]
c_1 、 c_2	Random in [1.8,2.2]
ξ_1 、 ξ_2	Random in [0,1]
RO setting	
Size of swarm	8
Particle dimensions	36
New random vectors generation	$N(b, \sigma)^{36}$
RLSE setting	
Number of parameter in consequence	105
P_0	$\alpha \cdot I$
α	1×10^{10}
I	105×105 identity matrix

表格 7 與其他文獻之比較(實驗二)

Method	MAPE		
	British Airlines	Ryanair Holdings Ltd	Southwest Airlines
ANN [22]	2.283	1.673	1.492
HMM [22]	2.629	2.011	1.928
Combination of HMM–fuzzy model [23]	1.529	1.356	-
ARIMA [23]	1.573	2.283	-
ANN [23]	1.504	1.504	-
Fusion HMM–ANN–GA with weighted average [23]	1.646	1.377	-
Proposed approach (RO-RLSE) (testing phase)	1.6532	1.8006	1.5139
Proposed approach (PSO-RLSE) (testing phase)	1.5721	1.5415	1.5168
Proposed approach (RoPso-RLSE) (testing phase)	1.4905	1.5033	1.3802

表格 8 10 次實驗之統計結果(實驗二)

# trials:10	RMSE								
	British Airlines			Ryanair Holdings Ltd			Southwest Airlines		
	RO-RLSE	PSO-RLSE	RoPso-RLSE	RO-RLSE	PSO-RLSE	RoPso-RLSE	RO-RLSE	PSO-RLSE	RoPso-RLSE
Best	0.4605	0.4142	0.3979	0.5109	0.4630	0.4490	0.2966	0.3047	0.2542
Worst	6.3242	2.6066	0.4313	5.5249	2.9013	0.4771	4.6239	2.0213	0.2862
Average	1.5276	0.9283	0.4123	1.5690	1.1140	0.4626	0.9711	0.7217	0.2721
Std	1.9874	0.7021	0.0131	1.8868	0.8633	0.0084	1.3172	0.5339	0.0114

# trials:10	MAPE								
	British Airlines			Ryanair Holdings Ltd			Southwest Airlines		
	RO-RLSE	PSO-RLSE	RoPso-RLSE	RO-RLSE	PSO-RLSE	RoPso-RLSE	RO-RLSE	PSO-RLSE	RoPso-RLSE
Best	1.6532	1.5721	1.4758	1.7601	1.5415	1.5033	1.5139	1.5168	1.3769
Worst	10.0559	4.9074	1.6137	12.7029	5.8697	1.6379	6.2528	5.2222	1.5575
Average	3.2895	2.1805	1.5399	3.6943	2.5161	1.5583	2.8214	2.2227	1.4626
Std	2.7240	1.0376	0.0452	3.3739	1.4798	0.0486	1.5809	1.1348	0.0566

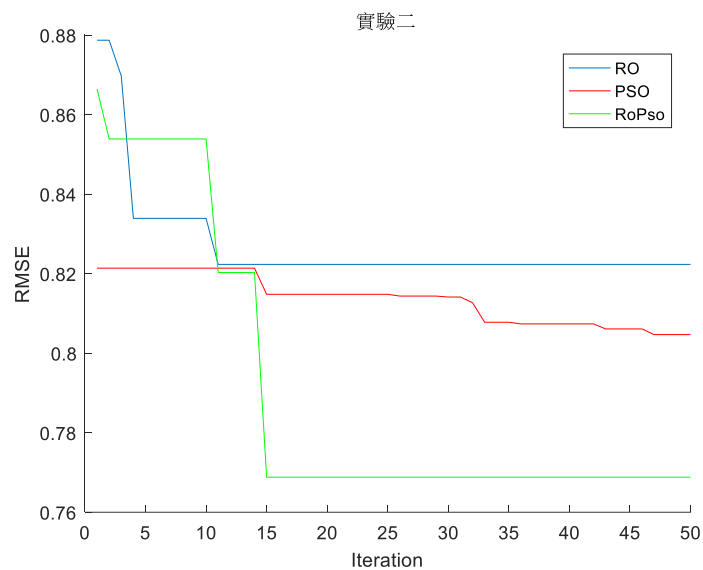


圖 19 模型學習曲線

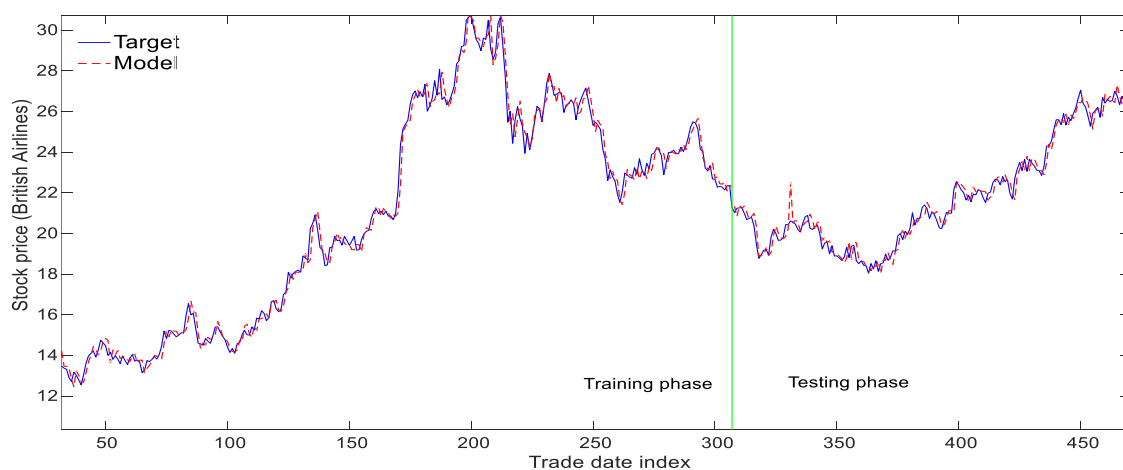


圖 20 模型訓練與預測結果-British Airlines

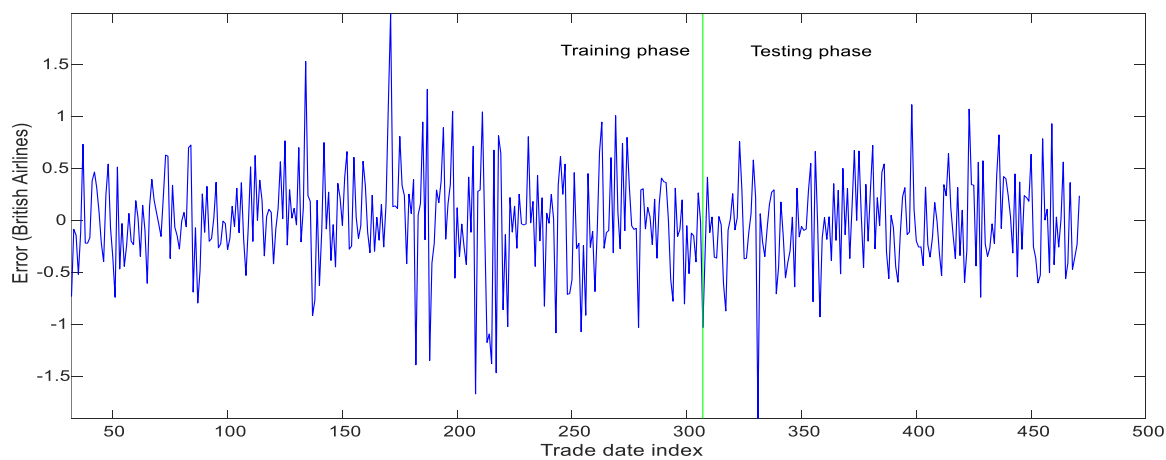


圖 21 訓練結果與目標之誤-British Airlines

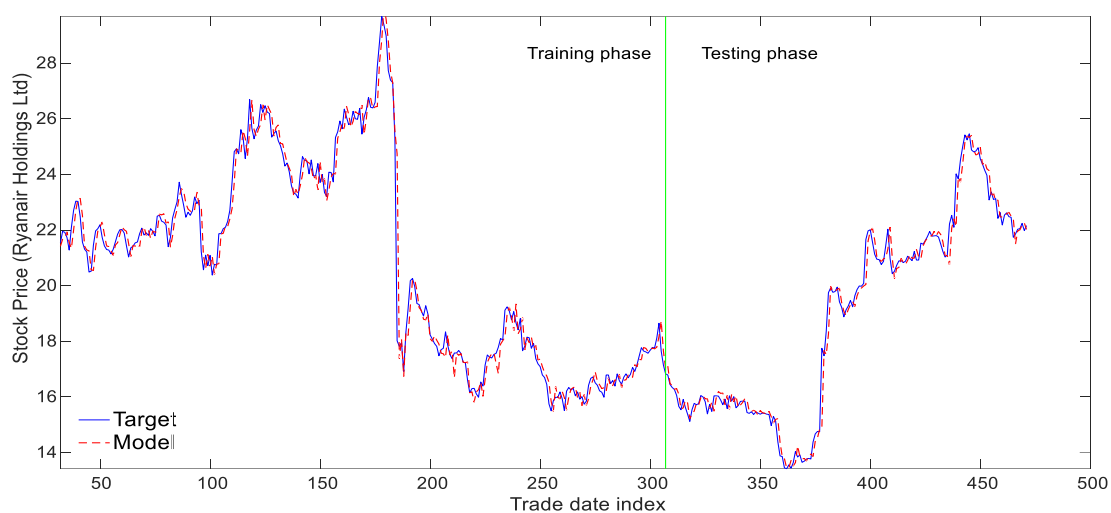


圖 22 模型訓練與預測結果-Ryanair Holdings Ltd

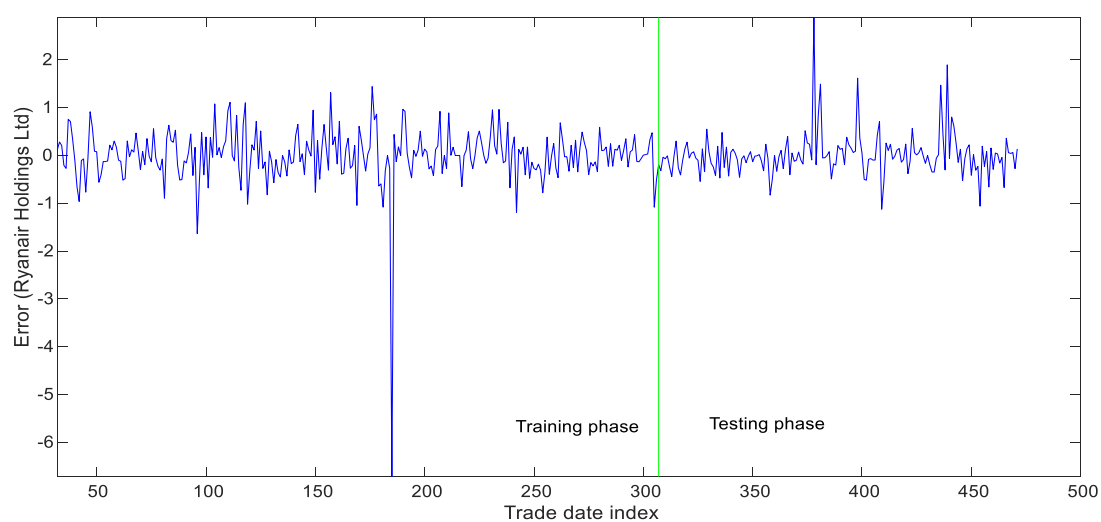


圖 23 訓練結果與目標之誤差-Ryanair Holdings Ltd

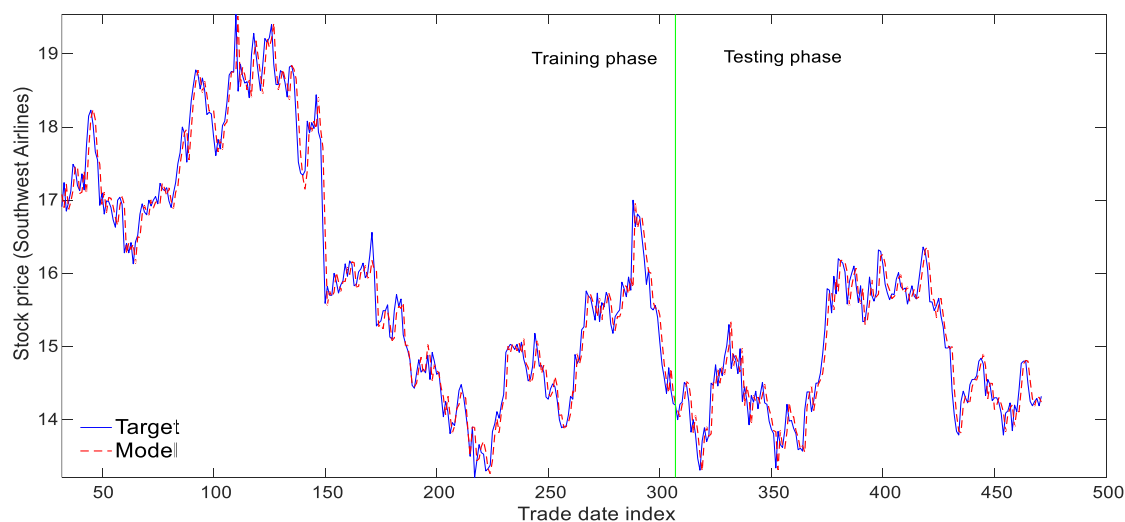


圖 24 模型訓練與預測結果-Southwest Airlines

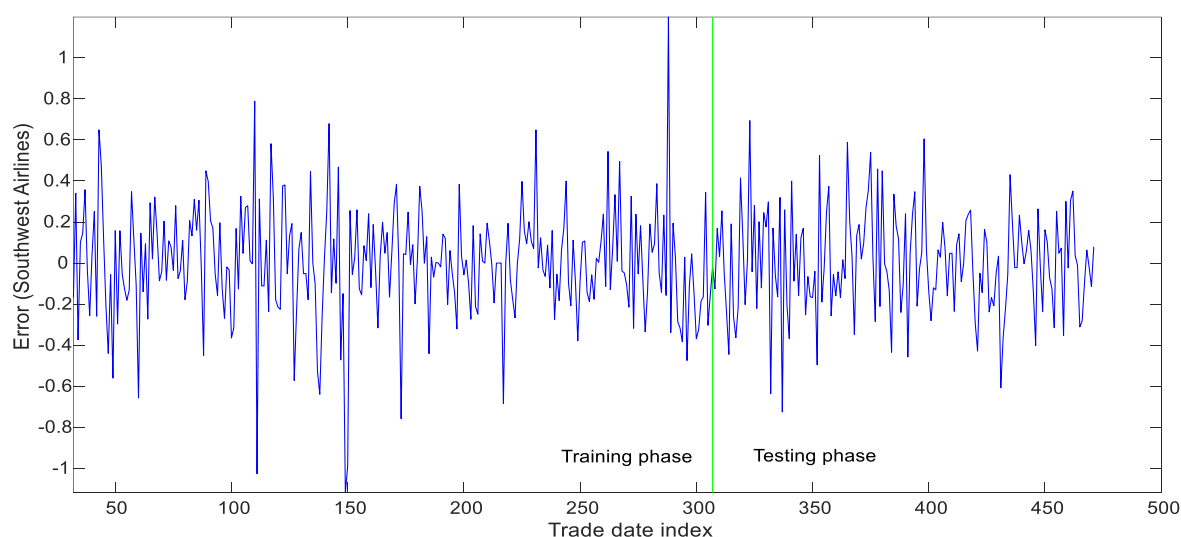


圖 25 訓練結果與目標之誤差-Southwest Airlines

4.3 實驗 3: 四個目標匯率預測-

在實驗 3 中，為測試模型四個目標的輸出表現能力，實驗目標採用匯率作為實驗對象，分別使用加拿大幣對美元(Canadian dollars, CAD)/ (U.S. Dollars, USD)、瑞士法郎對美元(Swiss Francs, CHF)/ (U.S. Dollars, USD)、德國馬克對美元(German Marks, DEM)/ (U.S. Dollars, USD)、英鎊對美元的匯率(British Pound, GBP)/ (U.S. Dollars, USD)等四種匯率。上述分別表示為 1 美元對應加拿大幣、瑞士法郎、德國馬克與英鎊之價格。

四種匯率之資料來源取自 <http://fx.sauder.ubc.ca/>，資料蒐集範圍為在 1971 年 1 月至 2006 年 11 月期間的每月平均匯率，共 431 筆。為了訓練研究模型，我們將資料集分成訓練和測試兩組資料集，使用 1971 年 1 月至 2000 年 12 月(共 360 筆)進行的訓練，並於 2000 年 1 月至 2006 年 11 月日(共 71 筆)進行測試。在原始資料集的前處理時，會將資料集處理成差分資料(漲跌)，接著會將四種匯率的資料集前 30 天的差分資料做為候選特徵，四個資料集共為 120 個候選特徵變數 $\{f_i, i = 1, 2, \dots, 120\}$ ， f_1 至 f_{30} 屬於 CAD/USD； f_{31} 至 f_{60} 屬於 CHF/USD； f_{61} 至 f_{90} 屬於 DEM/USD； f_{91} 至 f_{120} 屬於 GBP/USD，其中， i 值越大表示越接近目標。在此實驗中，有四個目標 (CAD/USD、CHF/USD、DEM/USD 和 GBP/USD)，我們使用兩個複數型態目標變數來表示它們。第一複數目標分別通過其實部和虛部處理第一實數目標(CAD/USD)和第二實數目標 (CHF/USD);第二個複數目標也分別通過其實部和虛部，處理第三個(DEM/USD)和第四個實數目標(GBP/USD)。後續經特徵選擇後，將選出對目標提供較多資訊量之特徵作為模型輸入，以進行訓練與測試，並將實驗重複進行十次，最後將實驗結果與其它文獻方法比較。

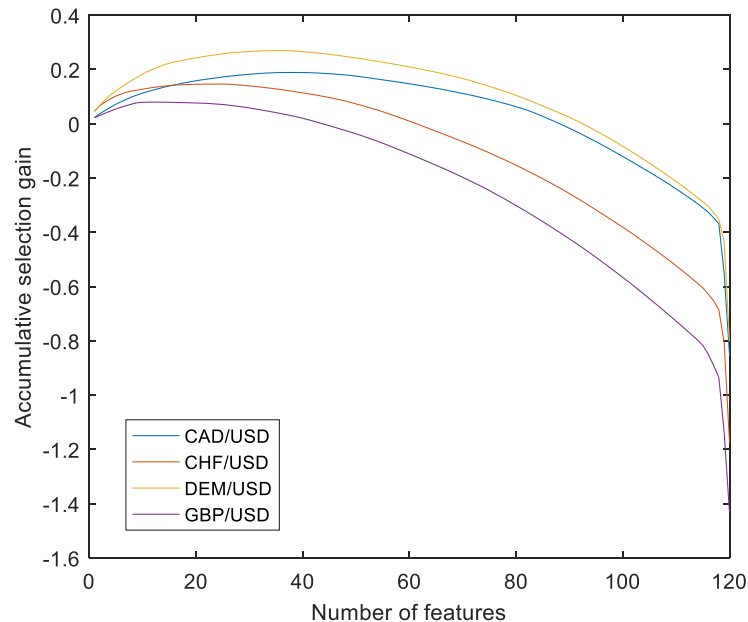


圖 26 特徵對目標 CAD/USD、CHF/USD、DEM/USD 與 GBP/USD 提供之累積增益量

圖 26 為特徵提供給目標的累積增益量，經由特徵挑選後，由 $f_{90}, f_{83}, f_{60}, f_{119}, f_{39}, f_{112}$ 被選用作模型的輸入， f_{39} 與 f_{60} 為 CHF/USD 之第 9 與第 30 個特徵； f_{83} 與 f_{90} 為 DEM/USD 之第 23 與第 30 個特徵； f_{112} 與 f_{119} 為 GBP/USD 之第 22 與第 29 個特徵，形成之訓練資料(TD)表示為 $TD = \{(\vec{x}, \vec{y})^{(i)}, i = 1, 2, \dots, |TD|\}$ ，其中， $\vec{x}^{(i)} = [f_{90}, f_{83}, f_{60}, f_{119}, f_{39}, f_{112}]^T$ ， $\vec{y}^{(i)}$ 則為對應目標值。於實驗中建構模型時，將每個模型輸入採用減法分群演算法(subtractive clustering algorithm)進行分群取得群數，每個分群即代表一個高斯複數模糊集。本實驗中，每個模型輸入分別分作 5、5、4、3、4 與 3 群，將產生 3600 個前鑑部（同條模糊規則數）。由於考量研究模型計算效率，因此在實驗中，將透過篩選出前 15 個前鑑部，選取方式同時驗 1。經由挑選後，本實驗每個模型輸入分別留下 3、3、3、3、3 與 2 個高斯複數模糊集，且共有 15 條模糊規則，下表為模型相關配置。

表格 9 模型參數配置(實驗三)

Model Inputs	$f_{90}, f_{83}, f_{60}, f_{119}, f_{39}, f_{112}$
Number of Gaussian complex fuzzy sets (GCFs) of each input	3、3、3、3、3、2
Number of GSCFS premises	15
Number of Takagi-Sugeno consequences	15
Complex -valued target(s)*	2
Model output(s)	2

實驗中採用 RoPso-RLSE 的複合式機器學習演算法，於 RoPso 的設定中，PSO 使用 10 顆粒子，RO 使用 10 顆粒子，每顆粒子皆為 51 個維度，粒子的維度表示為模型的前鑑部參數。學習總迭代次數為 50 次，其中 RO 與 PSO 的初始位置同前述實驗之方式來隨機生成。詳細學習演算法相關設定如表格 10。

表格 10 RoPso-RLSE 學習演算法參數設定(實驗三)

Iterations	50
Number of parameter in premises	51
Number of parameter in consequences	105
PSO setting	
Size of swarm	10
Particle dimensions	51
ω	Random in [0.6,0.95]
c_1 、 c_2	Random in [1.8,2.2]
ξ_1 、 ξ_2	Random in [0,1]
RO setting	
Size of swarm	10
Particle dimensions	51
New random vectors generation	$N(c, \sigma)^{51}$
RLSE setting	
Number of parameter in consequence	105
P_0	$\alpha \cdot I$
α	1×10^{10}
I	105×105 identity matrix

本實驗總共進行 10 次試驗，圖 27 為 10 次中表現最佳之學習曲線，圖 28、圖 30、圖 32 與圖 34 圖 16 為模型預測結果，圖 29、圖 31、圖 33 與圖 35 為模型預測結果與目標之誤差。為比較本論文提出之混合式學習演算法的效能，因此於表格 3 中分別列出 RO-RLSE、PSO-RLSE、ROPSO-RLSE 以及其他文獻方法之比較。

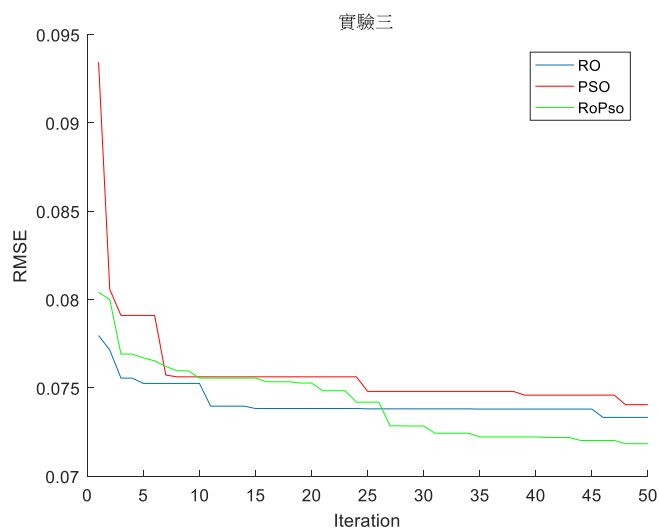


圖 27 模型學習曲線(實驗三)

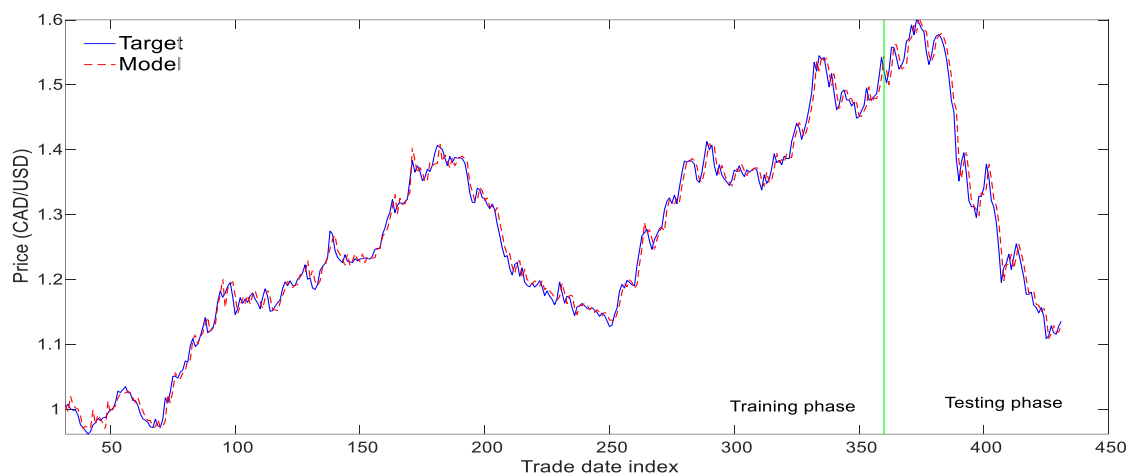


圖 28 模型訓練與預測結果(CAD/USD)

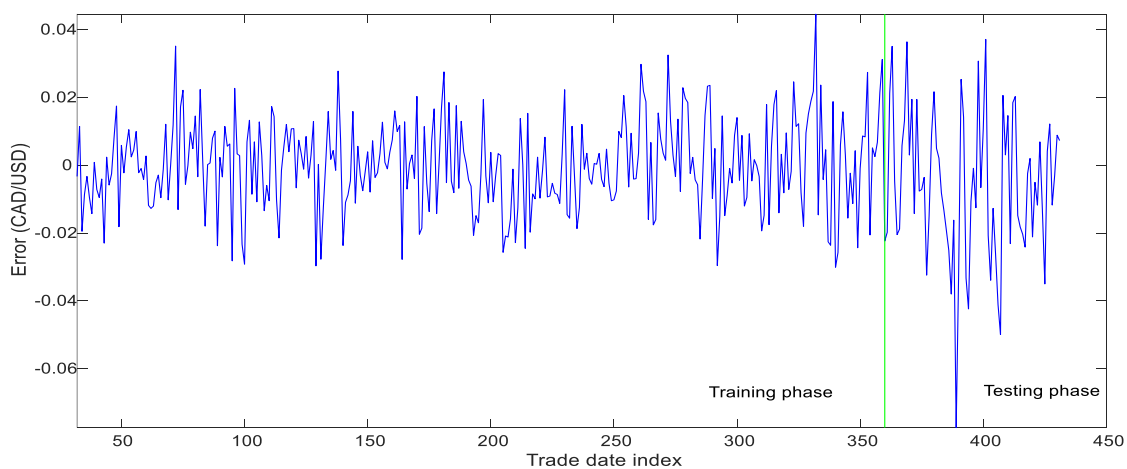


圖 29 訓練結果與目標之誤差(CAD/USD)

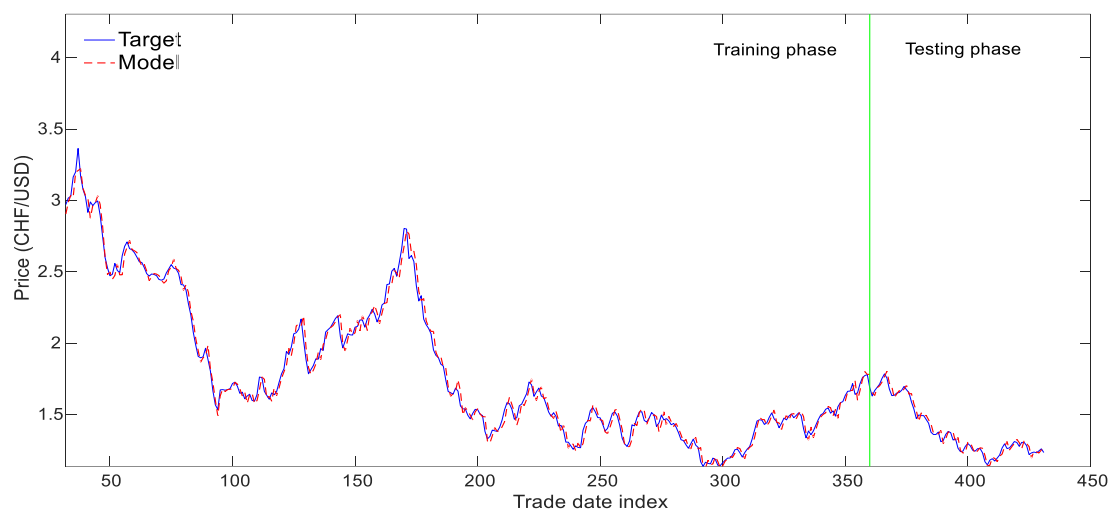


圖 30 模型訓練與預測結果(CHF/USD)

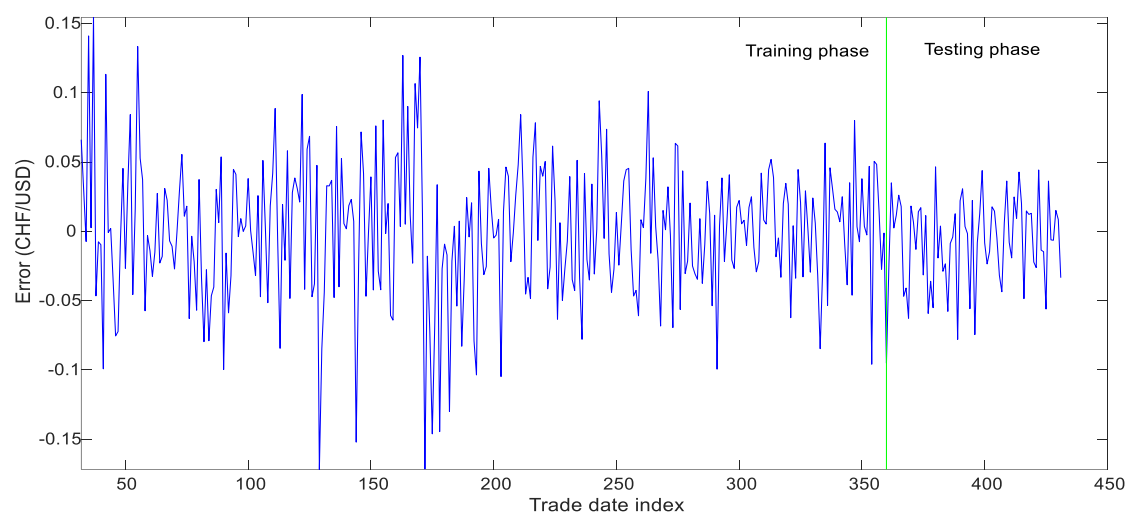


圖 31 訓練結果與目標之誤差(CHF/USD)

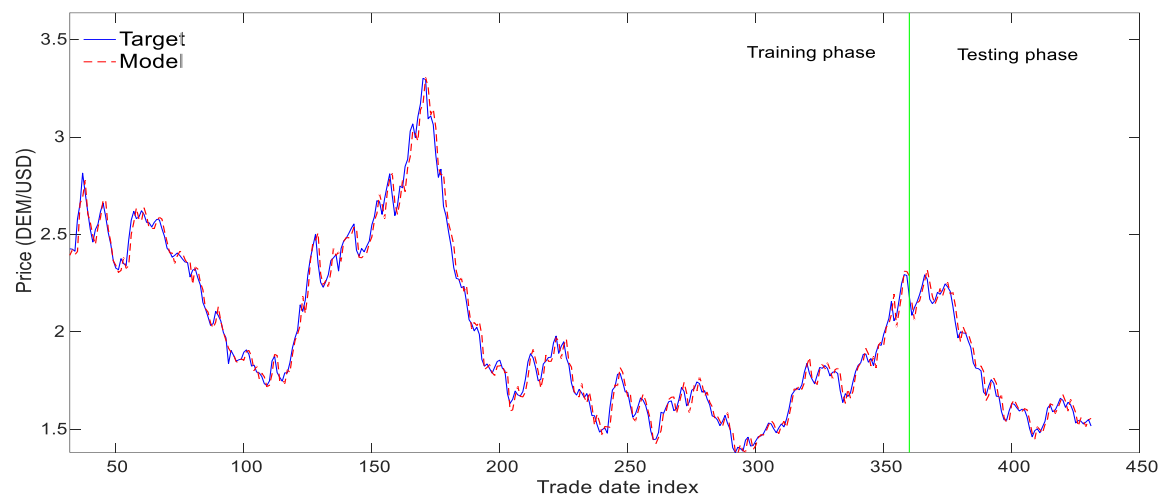


圖 32 模型訓練與預測結果(DEM/USD)

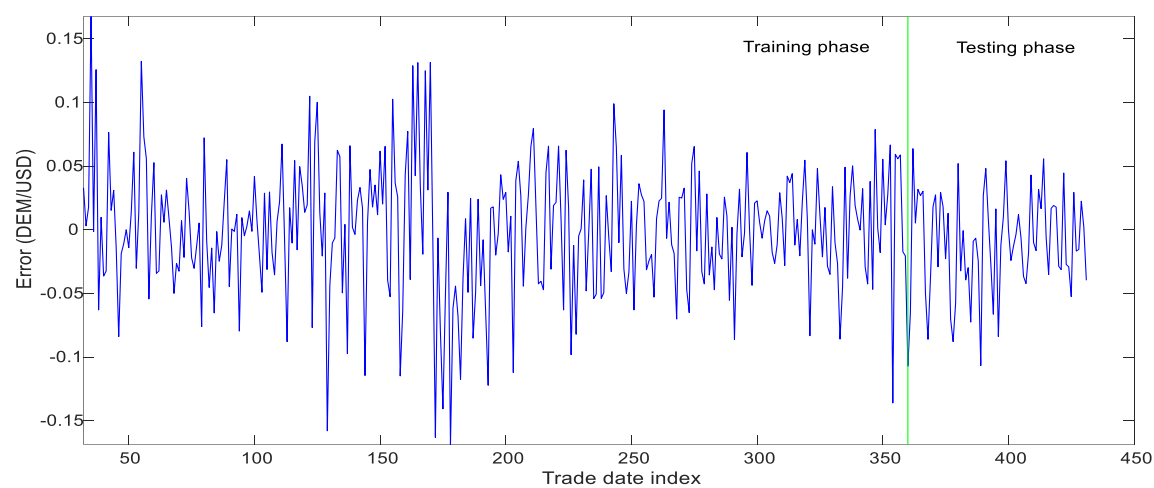


圖 33 訓練結果與目標之誤差(DEM/USD)

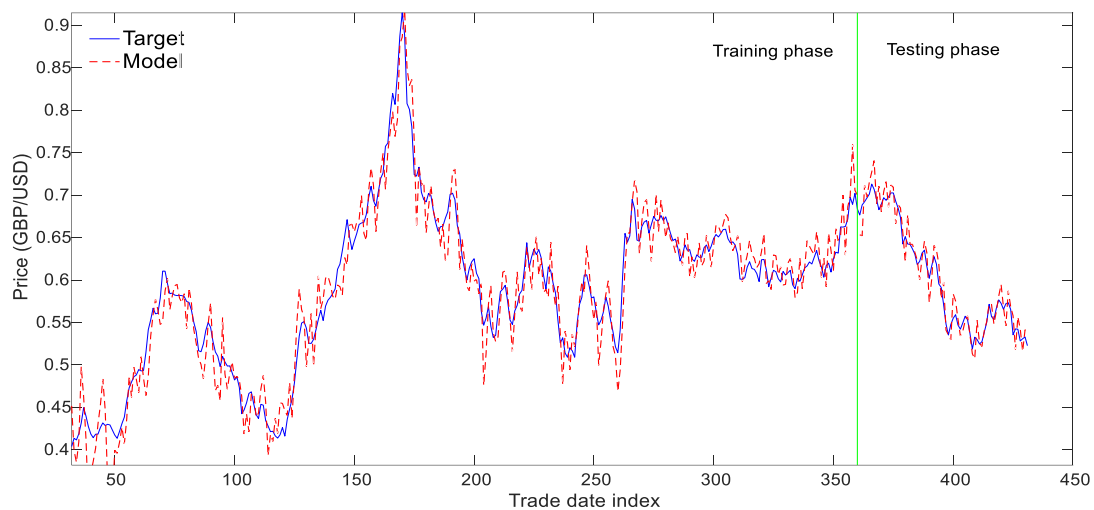


圖 34 模型訓練與預測結果(GBP/USD)

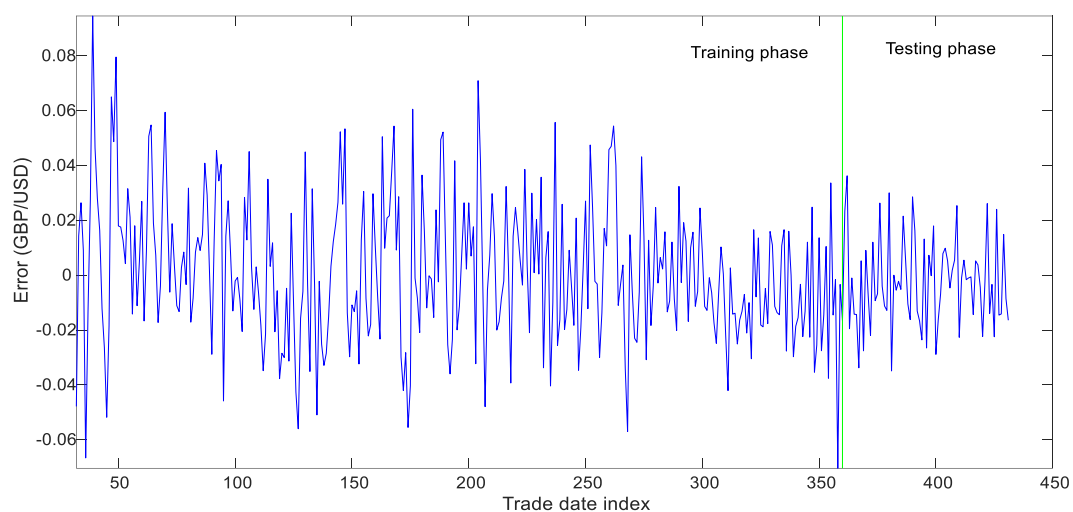


圖 35 訓練結果與目標之誤差(GBP/USD)

表格 11 與其他文獻之比較(實驗三)

Method	NMSE			
	CAD/USD	CHF/USD	DEM/USD	GBP/USD
Single RBF model [24]	-	-	0.0895	0.614
Simple averaging [24]	-	-	0.0812	0.686
Simple MSE [24]	-	-	0.0733	0.0789
Stacked regression [24]	-	-	0.0598	0.0484
Variance-based model [24]	-	-	0.0654	0.0467
RBF-based ensemble [24]	-	-	0.0462	0.0388
Proposed approach(RO-RLSE) (testing phase)	0.0330	0.0374	0.0391	0.0340
Proposed approach(PSO-RLSE) (testing phase)	0.0242	0.0539	0.0469	0.0588
Proposed approach(RoPso-RLSE) (testing phase)	0.0212	0.0299	0.0378	0.0278

表格 12 10 次實驗之統計結果(實驗三)

# trials:10	RMSE					
	CAD/USD			CHF/USD		
	RO-RLSE	PSO-RLSE	RoPso-ROSE	RO-RLSE	PSO-RLSE	RoPso-ROSE
Best	0.0232	0.0231	0.0215	0.0330	0.0342	0.0310
Worst	2.9208	0.5264	0.0281	3.4268	0.3576	0.0515
Average	0.3707	0.1170	0.0237	0.4562	0.1123	0.0345
Std	0.9078	0.1655	0.0019	1.0721	0.1291	0.0062
# trials:10	DEM/USD			GBP/USD		
	RO-RLSE	PSO-RLSE	RoPso-ROSE	RO-RLSE	PSO-RLSE	RoPso-ROSE
	RO-RLSE	PSO-RLSE	RoPso-ROSE	RO-RLSE	PSO-RLSE	RoPso-ROSE
Best	0.0379	0.0396	0.0376	0.0162	0.0140	0.0170
Worst	3.2404	0.7981	0.0415	3.2498	0.7870	0.0216
Average	0.4153	0.1992	0.0397	0.4016	0.1863	0.0185
Std	1.0026	0.2508	0.0012	1.0113	0.2548	0.0015
# trials:10	NMSE					
	CAD/USD			CHF/USD		
	RO-RLSE	PSO-RLSE	RoPso-ROSE	RO-RLSE	PSO-RLSE	RoPso-ROSE
Best	0.0245	0.0242	0.0209	0.0338	0.0364	0.0299
Worst	387.2824	12.5812	0.0357	365.0697	3.9760	0.0825
Average	39.9101	1.7404	0.0256	38.6295	0.8585	0.0380
Std	122.1042	3.9498	0.0043	114.8848	1.5657	0.0160
# trials:10	DEM/USD			GBP/USD		
	RO-RLSE	PSO-RLSE	RoPso-ROSE	RO-RLSE	PSO-RLSE	RoPso-ROSE
	RO-RLSE	PSO-RLSE	RoPso-ROSE	RO-RLSE	PSO-RLSE	RoPso-ROSE
Best	0.0351	0.0384	0.0347	0.0234	0.0173	0.0256
Worst	257.0568	15.5925	0.0422	937.5077	54.9747	0.0416
Average	26.3663	2.3579	0.0386	96.0250	8.2685	0.0304
Std	81.0779	4.8764	0.0023	295.7435	17.2483	0.0050

第五章 實驗結果討論

本論文使用之計算模型為複數模糊類神經網路，用混合式機器學習演算法 RoPso-RLSE 來進行模型參數之優化，並於資料輸入模型前，事先將資料進行特徵選取，選出對目標有影響力之特徵進入模型學習與預測。其中，計算模型結構主要分作前鑑部與後鑑部兩部分，於參數學習時，使用 RoPso 負責更新前鑑部參數，後鑑部參數優化則由 RLSE 負責。實驗部分，分別採用不同目標個數來進行實驗，以試驗本論文提出之混合式機器學習演算法與複數模糊類神經網路模型之表現能力。

實驗一中使用股票市場中之 Intel 與 Microsoft 兩間公司之股價進行預測，模型輸出為一複數值，其中實部與虛部分別表示為 Intel 與 Microsoft 之模型預測值。從實驗結果可知，不論採用 RO-RLSE、PSO-RLSE 與 RoPso-RLSE，本論文模型於 Intel 與 Microsoft 的預測皆表現良好，預測效果皆優於比對的文獻方法。Intel 於文獻方法中，最好之 MAPE 值為 0.04459，Microsoft 為 0.051907，而本論文之模型其 MAPE 值分別 0.0144 與 0.0121。顯示，本論文模型於同時預測兩個目標時有良好的預測表現。於機器學習演算法部分，由三者之實驗數據可知，採用 RoPso-RLSE 比使用 RO-RLSE 與 PSO-RLSE 有較佳之表現，且於表格 4 中 10 次試驗之數據顯示，RoPso-RLSE 其預測表現之平均值與標準差皆優於 PSO-RLSE 與 RO-RLSE，顯示 RoPso-RLSE 於模型中尋找參數有較佳之穩定性。

在實驗二中，採用三目標之股票預測，分別為 British Airlines、Ryanair Holdings Ltd 與 Southwest Airlines，此三目標皆屬於航空業的公司。此實驗之模型輸出為兩組複數值，第一組複數之實部與虛部分別表示為 British Airlines 與 Ryanair Holdings Ltd 的模型預測值；而第二組複數之實部則代表 Southwest Airlines 的模型預測值，虛部則未使用。從實驗結果可知，對於本論文所提出之模型而言，其使用 RoPso-RLSE 方法時，在 British Airlines 與 Southwest Airlines 的預測上表現良好，皆優於比對之文獻方法。Ryanair Holdings Ltd 則是表現稍差一些，最好之文獻方法其 MAPE 值為 1.356，而

本論文方法 Ryanair Holdings Ltd 的 MAPE 值為 1.5033，由於本論文模型為三目標輸出，預測結果中只有一個目標之表現沒有優於文獻方法，且其差距並不大。因此在多目標的預測上，本論文之模型可稱上具有競爭力。對於 RoPso-RLSE、RO-RLSE 與 PSO-RLSE 三者之間的表現差異，實驗中分別進行測試的結果來看，RoPso-RLSE 比起 RO-RLSE 或 PSO-RLSE 訓練出之模型表現更佳，顯示出本論文所提出之混合式機器學習演算法 RoPso-RLSE，相對於單獨使用 RO 或 PSO 演算法，在模型參數的最佳化上有較好的表現。

第三的實驗採用匯率作為實現對象，分別為分別使用加拿大幣對美元(Canadian dollars, CAD)/ (U.S. Dollars, USD)、瑞士法郎對美元(Swiss Francs, CHF)/ (U.S. Dollars, USD)、德國馬克對美元(German Marks, DEM)/ (U.S. Dollars, USD)、英鎊對美元的匯率(British Pound, GBP)/ (U.S. Dollars, USD)等四種匯率。實驗之模型輸出為兩組複數值，第一組複數之實部與虛部分別表示為 CAD/USD 與 CHF/USD 的模型預測值；而第二組複數之實部與虛部分別為 DEM/USD 與 GBP/USD 的模型預測值。在文獻比較上，本論文所提出之模型在搭配 RoPso-RLSE 的表現下，預測表現良好。DEM/USD 與 GBP/USD 的表現皆比文獻方法更佳，本論文模型之誤差指標 NMSE 值分別為 0.0378 與 0.0278，而文獻中最佳之方法其誤差指標 NMSE 值各自為 0.0462 與 0.0388，說明本論文之模型在多目標的預測能力有良好的表現。在演算法方面，為使用 RoPso-RLSE 之模型表現最好，再次顯示混合式學習演算法 RoPso-RLSE 可以有較佳的表現。

本論文中所提出之混合式機器學習演算法 RoPso-RLSE，藉由 RoPso 與 RLSE 的複合式運作，分別優化模型之前鑑部與後鑑部參數。其中，前鑑部透過並行式運作的演算法 RoPso 來找尋非線性參數空間的解，此一 RoPso 經由 RO 與 PSO 並行運作，加上競爭與學習之機制，來提升單獨使用 RO 或 PSO 之表現。若單獨使用 RO 於非線性參數空間尋找解，雖然 RO 具有高度隨機性與運算快速之特性，但其選取最佳粒子位置之方式，採用以起始點為中心，利用常態分佈隨機生成多個隨機向量於起始點運算，來取得多個新粒子位置，並取得該次迭代中最好之粒子作為下次的起始點。意即 RO

的粒子中只有進行比較的過程即決定下次起始點，而高度的隨機性將可能無法在有限的迭代次數中收斂。若為 PSO 於非線性參數空間尋找解，其特性為粒子間的移動皆會考慮本身與群體的移動經驗，再修正移動方向，由於參考因素包含該粒子目前位置、該粒子移動過程中最好之位置與群體最好之位置，因此運算較為耗時，且粒子移動的離散度也易造成過度早熟之問題。

由於上述原因，本論文提出之並行式演算法 RoPso 將同時具有高度隨機性與方向性，藉由競爭機制先比對 RO 與 PSO 兩者，再經學習機制將較佳者作為兩者下次之移動參考。從三個實驗結果中，比對單獨使用 RO-RLSE 以及 PSO-RLSE 於模型的效果，顯示 RoPso-RLSE 方法在表現上有更好的效果。而後鑑部採用 RLSE，利用其計算快速且適用於線性函數之性質，使模型後鑑部參數不須透過 RoPso 即可優化，達到參數簡化之效果。

第六章 結論

本篇論文中使用混合式機器學習來進行模型參數的優化，藉由並行式演算法 RoPso，讓模型在學習時能夠比單獨使用粒子群最佳化演算法或隨機最佳化演算法來達到更快速的收斂，並且提高找到全域最佳化的機率。模型方面，本篇論文使用之模型為複數模糊類神經網路，且使用特徵選取來挑出提供給目標大量資訊量的特徵做為模型輸入，並將以上方法實際運用於股票預測以及匯率的預測，從與其文獻的比較結果可能看出本篇研究所使用的機器學習演算法、模型、特徵選取的效能，在此整理了幾點本篇論文研究的貢獻，如下：

6.1 結論

特徵選取

由於資料量增長迅速，對於以資料為學習基礎的機器學習，有時過多的資料反而會造成機器學習的阻礙。因此，如何藉由特徵選取從資料中選出對目標有高度影響力的資料作為學習是個重要的議題。在本篇論文中透過基於夏農資訊熵的特徵選取方式，來找出提供給目標較多資訊量的特徵，來有效的減少過多的特徵數量與選出對目標具有影響力的特徵。

複數模糊類神經模型

本論文模型使用複數模糊集於模糊類神經網路，複數模糊集將一維介於0到1的歸屬程度，延伸到複數平面上，使其歸屬程度包含在平面的二維單位圓盤之中，由於進到複數空間中，使得複數型歸屬程度能夠負載更多的資訊，且運用於模型之時，複數模糊集合可以提升模糊類神經模型對於非線性問題的映射能力，達到較佳的模型彈性與適應能力。此外，藉由複數模糊集的特性使本論文模型能夠同時預測多個目標，相對於過去預測單一目標的模型而言，能夠預測多目標之模型將更具有競爭力。

混合式機器學習

本論文提出 RoPso-RLSE 混合式機器學習應用於最佳化複數模糊類神經模型的參數調整。其中，利用隨機最佳化演算法的易於運算與高度隨機性，以及粒子群最佳化在粒子移動時會參考本身與群體的移動經驗，結合兩者並採用競爭與學習的機制運作，來提高單一使用 RO 或 PSO 之粒子移動變化，以降低粒子提早收斂或落入局部最佳解的機率。在本論文中使用 RoPso 來調整模型前鑑部參數；而後鑑部參數則是採用遞迴式最小平方估計法，遞迴式最小平方估計法是以利用數學計算來取得參數，除了減少 RoPso 需要更新的參數數量，另外，其快速運算的特性，能在短時間求出參數解，因此能增加模型的學習效能。

6.2 未來研究方向

本論文中使用之模型與混合式機器學習來優化模型參數，根據實驗結果所示，都有不錯的預測表現。但仍然有幾點可以在未來加以延伸。以下提出未來可延續的研究方向：

1. 挑選進入模型的特徵數量

本篇論文中透過特徵選取的過程，將資料進行篩選，最後挑選出對目標有較多影響力之特徵作為模型輸入。而在論文中以使用者自行決定的方式，採用上限 6 個，下限 2 個的設定作為選取數量之依據，因此在未來可以藉由實驗的方式，來確認多少個特徵數量對於模型運算效能與其預測表現能夠呈現較佳的平衡點。

2. 模型規則數量之決定

本篇論文實驗中，針對大量的規則數問題，採用以資料較為密集的前鑑部作為候選者來提升運算效率，其中規則數量之決定為使用者自訂。因此，在未來研究中可透過實驗來檢視模型規則數與效能表現之關係，來制定更有效之選擇策略，以找出較適合的模型規則數。

3. RoPso 最佳化演算法的粒子維度數過高

在本篇論文中，將模型參數分作前鑑部與後鑑部兩部分。後鑑部參數由遞迴最小

平方法估計法來進行學習，因此能夠減少 RoPso 演算法中粒子的維度數，但當增加模型輸入與模糊集數量時，粒子的維度數也隨之增加，將導致 RoPso 最佳化演算法的效能降低，因此在未來當粒子維度過高時，或許能將粒子的維度數切割成多顆低維度的粒子，利用降低每個顆粒子負責的參數維度，來提升學習演算法的結果與效能。

4. 區域最小化問題

在本論文中提出了並行粒子群最佳化與隨機最佳化運作的 RoPso 來增加找到全域最小化的機率，但並不表示本論文中使用之最佳化演算法一定不會陷入區域最小化，因此，在未來研究中可藉由加入一些跳脫的相關機制來減少粒子群落入區域最小化的可能性。

參考文獻

- [1] O. Reyes, C. Morell and S. Ventura, “Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context.” *Neurocomputing*, Volume 161, pp. 168-182, August 2015.
- [2] I. Guyon et al., “Gene selection for cancer classification using support vector machines.” *Machine Learning*, Volume 46, Issue 1–3, pp. 389–422, January 2002.
- [3] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, Volume 8, Issue 3, pp. 338-353, June 1965.
- [4] D. Ramot, “Complex fuzzy sets,” *IEEE Transactions on Fuzzy Systems*, Volume 10, Issue 2, pp. 171-186, April 2002.
- [5] J.-S.R. Jang, “ANFIS: Adaptive-network-based fuzzy inference system,” *IEEE Transactions on Systems*, Volume 23, Issue 3, pp. 665-685, May 1993.
- [6] George S. Atsalakis and Kimon P. Valavanis, “Forecasting stock market short-term trends using a neuro-fuzzy based methodology,” *Expert Systems with Applications*, Volume 36, Issue 7, pp.10696-10707, September 2009.
- [7] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proc. IEEE International Conf. on Neural Networks*, Australia, 1995.
- [8] Matyas, J, “Random optimization,” *Automation and Remote Control*, Volume 26, pp. 244-251, 1965.
- [9] F. J. Solis & J. B. Wets, “Minimization by random search techniques. Mathematics of Operations Research”, *Mathematics of Operations Research*, Volume 6, No. 1, pp. 19-30, February 1981.
- [10] N. Baba, T. Shoman, & Y. Sawaragi, “A modified convergence theorem for a random optimization method,” *Information Sciences*, Volume13, Issue 2, pp. 159-166, 1977.
- [11] N. Baba, “Convergence of a random optimization method for constrained optimization problems,” *Journal of Optimization Theory and Applications*, Volume 33, Issue 4, pp. 451-461, April 1981.
- [12] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, Volume 15, Issue 1, pp. 116–132, Jan.-Feb. 1985.
- [13] E.H. Mamdani, “Applications of fuzzy algorithm for control a simple dynamic plant,” *Proceedings of the Institution of Electrical Engineers*, Volume 121, Issue 12, pp. 1585–1588, December 1974.
- [14] E.H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International Journal of Man–Machine Studies*, Volume 7, Issue 1, pp. 1–13, January 1975.
- [15] N. Laptev, “Time-series extreme event forecasting with neural networks at Uber,”

- International Conference on Machine Learning*, Australia, 2017.
- [16] P. Singh, "Rainfall and financial forecasting using fuzzy time series and neural networks based model", *International Journal of Machine Learning and Cybernetics*, Volume 9, Issue 3, pp. 491-506, March 2018.
- [17] D. Pradeepkumar and V. Ravi, "Forecasting financial time series volatility using Particle Swarm Optimization trained Quantile Regression Neural Network," *Applied Soft Computing*, Volume 58, pp. 35-52, September 2017.
- [18] A. Yadav and K. Sahu, "Wind forecasting using artificial neural networks: a survey and taxonomy," *International Journal of Research In Science & Engineering*, Volume 3, Issue 2, pp. 148-155, March-April 2017.
- [19] G. Asencio-Cortés, "A medium-large earthquake magnitude prediction in Tokyo with artificial neural networks," *Neural Computing and Applications*, Volume 28, Issue 5, pp. 1043–1055, May 2017.
- [20] K. Lang, "Short-term load forecasting based on multivariate time series prediction and weighted neural network with random weights and kernels," *Cluster Computing*, pp. 1-9, 2018.
- [21] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, Volume 27, Issue 3, pp. 379-423, July 1948.
- [22] M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: a new approach," *5th International Conference on Intelligent System Design and Application*, pp. 192–196, Warsaw, Poland, 2005.
- [23] M. R. Hassan, "A combination of hidden Markov model and fuzzy model for stock market," *Neurocomputing*, Volume 72, Issue 16-18, pp. 3439-3446, October 2009.
- [24] Lean Yua, Kin Keung Lai, and Shouyang Wang, "Multistage RBF neural network ensemble learning for exchange rates forecasting," *Neurocomputing*, Volume 71, Issues 16–18, Pages 3295–3302, October 2008.
- [25] C. Li and T. Chiang, "Complex neuro fuzzy ARIMA forecasting-a New approach using complex fuzzy sets," *IEEE Transactions on Fuzzy Systems*, Volume 21, Issue 3, pp. 567-584, June 2013.
- [26] K. J. Åström and B. Wittenmark, "Adaptive Control," Addison Wesley, 1989.
- [27] O. D. Crisalle, "Adaptive Control of Photolithography," *American Institute of Chemical Engineers Journal*, Volume 38, Issue 1, pp. 1-14. , January 1992.
- [28] R.C. Eberhart and Shi, Y. "Tracking and Optimizing Dynamic Systems with Particle Swarms." *Proceedings of the Congress on Evolutionary Computation*, Volume 1, pp. 94–100, 2001.
- [29] A. Chatterjee and P. Siarry, "Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization" *Computers and Operations Research*, Volume 33, pp. 859-871, 2004.

- [30] S. Fan, Shu-Kai, and Erwie Zahara, “A Hybrid Simplex Search and Particle Swarm Optimization for Unconstrained Optimization” *European Journal of Operational Research*, Volume 181, pp. 527-548, 2006.
- [31] M. Clerc and J. Kennedy, “The Particle Swarm: Explosion, Stability, and Convergence in a Multimodal Complex Space.” *Proceedings of the Congress of Evolutionary Computation*, Washington, DC, IEEE, Piscataway, NJ, USA, Volume 6, pp. 58-73, 2000.
- [32] A. Coloni, M. Dorigo, and V. Maniezzo, “Distributed Optimization by Ant Colonies,” *The European Conference on Artificial Life*, pp. 134-142, Paris, France, 1991.
- [33] J. H. Holland, “Adaptation in Natural and Artificial Systems,” *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press Cambridge, MA, USA.
- [34] A. Kazem, “Support vector regression with chaos-based firefly algorithm for stock market price forecasting,” *Applied Soft Computing*, Volume 13, no. 2, pp. 947-958, February 2013.
- [35] 國立中央大學資訊管理所李俊賢教授, 研究生訓練課程內容 2016-2018, 紀錄筆記。(未發表)