Paper Title\* **(**use style: ***paper title***)

Subtitle as needed ***(paper subtitle)***

Chi Feng Lina and Chunshien Lib

Information Management. National Central University, National Central University, Taoyuan, TAIWAN

aEmail: j8888888871@yahoo.com.tw

bEmail: jamesli@mgt.ncu.edu.tw (corresponding author)

*Abstract*—本篇論文主要探討不同演算法在時間序列上的預測效能，並透過模擬的投資策略做效能測試。時間序列的預測是一個很廣泛且重要的議題。我們提出了球型複數類神經模糊系統 (SCNFS)來對時間序列進行預測。一般的複數類神經模糊系統 (CNFS)，擁有一組複數型態的輸出，實數部分和虛數部分可以個別地針對不同目標做預測。本篇論文，模型對原本的CNFS中的複數模糊集 (CFSs)進行改良，歸屬程度仍然是在複數型態的值，但是能擁有多組輸出，使得模型可以同時預測兩個以上的目標。模型設計上，前鑑部使用高斯型態的球型複數神經模糊集 (SCFSs)，後鑑部則使用Takagi-Sugeno的線性函式，兩者使用IF-THEN規則型式連結。此外，為了最佳化模型的預測結果，本論文使用分而治之的概念，前鑑部個別使用不同的演算法Particle Swarm Optimization (PSO)、Artificial Bee Colony Optimization (ABCO)進行參數優化，後鑑部則使用Recursive Least Squares Estimator (RLSE)進行參數優化。最後，三個實驗將被用來測試，不同學習演算法的效能差異，並將模型輸出配合本論文提出的投資策略，模擬實際投資的效果。

Keywords— Sphere complex neurofuzzy system (SCNFS); complex fuzzy set (CFS); complex neurofuzzy system (CNFS), time-series forecasting.

# Introduction

人工智慧在現實中有很多的領域，從18、19世紀就有許多相關的研究，但是由於硬體的設備問題，導致成果一直無法應用在一般的生活中。如今隨著硬體設備的發展，人工智慧已經對人類帶來不少的幫助，研究更是多不勝數，像是財經方面的預測，於2017年，Jardin et al. [7] 提出量化的方法定義公司的健康狀態藉此預測公司是否破產；Ravi et al. [26]使用混沌理論、多層感知機和多目標優化的演算法，對匯率進行預測；Heaton et al. [12] 提出一套深度學習的模型應用於財金投資組合。圖像辨識方面Feng et al. 提出Discriminative Locality Alignment Network (DLANet)進行圖像辨識[9]，此方法提升了對於場景的辨識能力；Howard et al. [14]提出了MobileNets，透過流線型結構給予類神經網路權重，此概念專門提供給行動裝置做影像辨識使用；Zbontar et al. 提出一方法，透過卷積神經網路應用於圖像補釘上，使得立體圖的配對更加容易。能源預測方面，Fumo et al. [10]使用回歸分析對住宅的能源消耗做預測；Jovanović et al. [16]使用倒傳遞類神經網路等進行加熱時能源的預測，有著不錯的精準度。

從上述可以得知人工智慧的應用非常廣泛。其中，財務金融最被廣泛研究，因為它是一個牽扯到多因素的問題，包含了全球趨勢、地方文化、政治影響等，因此匯率、股票投資等是最難被預測的，這些問題的資料是根據時間所排序，又可視為時間序列的預測問題，過去已經有很多學者提出不同的方法預測時間序列的預測問題，像是ARIMA[21, 22]、模糊理論、類神經網路運算、類神經模糊系統等。其中，最常被提出的就是類神經模糊系統[13, 28, 30, 40]，也是效果表現最優異的。

類神經模糊系統 (NFSs)是基於類神經網路 (Neural Network)的延伸，所謂的神經網路是一種用多層神經元所串連起來的網路，透過層與層之間的資料傳輸來得到輸出，起源於1943年McCulloch et al.[24] 使用閥值邏輯 (Threshold logic)模擬出人類腦中的神經元，意即當一個神經元接收到資訊，會判斷是否要產生興奮反應的機制。於1956年Rochester [27]更進一步的創造出感知機，但仍然無法解決非線性問題，直到1975年Werbos [36]提出倒傳遞演算法，使得多層的類神經網路可以被訓練，不但解決了非線性問題，也造就日後類神經網路中訓練的概念。現今火紅的深度學習即是類神經網路的一種，在此領域中已有不少的研究[9, 11, 38]。

類神經模糊系統 (Neuro-fuzzy Systems, NFSs)是由模糊系統與類神經網路所組合而成，其中，模糊系統有所謂的IF-THEN規則，這些規則如同我們人類的經驗法則，較易被人類所理解。現今，被大量使用的類神模糊系統都是混合型類神經模糊系統[34]，意即將模糊系統中的元素使用於類神經網路之中，使整體的架構更有彈性。如上述所提，類神經模糊系統的特性使他對時間序列的預測有著不錯的效果。所以現在有關時間預測的研究，大多採用類神經網路為模型架構。

本論文採用類神經模糊系統為架構，當中包含了IF-THEN規則，建造多層神經元。在模型實作方面，我們將模糊系統結合類神經網路，IF-part使用高斯型態的球型複數神經模糊集，使得模型可有多組複數型態輸出，進行多目標預測，Then-part使用Takagi-Sugeno的線性函式[31]，兩者神經元透過IF-THEN規則概念結合，為一對一的關係。透過此模型和機器學習我們預期對時間序列的預測可以有優異的表現。

關於模糊集，在1965年由Zadeh學者，最先提出模糊集的概念[39]，使資料可以透過某個函式，得到介於0到1之間的歸屬程度 (Membership degree)。之後在2002年，有另一篇研究提出了複數模糊集 (CFSs)[25]的概念，歸屬程度可以透過函式得到一個複數型態的值，這使得歸屬程度可以呈現在一個半徑為1的複數單位圓盤中。這個概念使原本能表示的歸屬程度更加的豐富。一般來說我們可以透過複數型神經模糊集系統 (CNFS)[19, 20]，得到一組複數型態的輸出，而實數部分和虛數部分可針對不同目標做預測，所以可針對兩個不同目標。目前兩個目標的預測已經有很多的研究產出[19, 20, 22]。而為了同時對更多的目標做預測，本篇論文將原本複數型神經模糊集系統進行改良，將原本使用的複數模糊集 (CFSs)改成球型複數模糊集 (SCFSs)，其中歸屬程度仍然是複數型態，但能呈現在3-D的立體空間中，能擁有更多組複數型態的輸出，意即可以一次對多個目標進行預測。

在本研究中，為了使資料可以有效的被應用，在資料前處理的部分，我們根據將原始資料的30個漲跌值作為特徵，並透過夏農資訊熵 (Shannon Entropy)[29]，去計算他們個別對目標的資訊貢獻量，除此之外，我們透過了多目標特徵選取的概念[23]，算出每個特徵對目標的有效資訊量，以此作為挑選訓練資料的依據。從資料中萃取出最有效的資料，除了降低模型的運算負擔，也能有效的提升預測的效能。最後，機器學習部分，我們個別使用知名的Particle Swarm Optimization (PSO)[8]、Artificial Bee Colony Optimization (ABCO)[17] 兩種演算法和廣為人知的Recursive Least Square Estimator (RLSE)[15]進行參數優化，並將它們分別整合成兩種方法我們稱之為PSO-RLSE [18]、ABCO-RLSE。我們將前鑑部和後鑑部參數藉由不同的演算法訓練，想透過divide-and-conquer原理，降低搜尋的維度，使模型更容易找到最佳解，提高整體的效能。

# Methodology

本章節將說明本研究一系列的研究方法以集模型設計。本研究使用資料驅動方式決定模型結構的大小，模型實作使用球式複數模糊集，且分別利用不同的演算法(PSO、ABC)優化其參數，遞迴式最小平方演算法則最佳化後鑑部參數。在資料進入模型之前，透過多目標特徵挑選[23]，挑選出對所有目標較為有效之特徵資料集合，減少龐大資料對模型的負擔。最後將結果配合投資策略做不同演算法的比較。

## Sphere Complex Fuzzy Sets

1965年提出的模糊集合概念，可以導出元素對集合一對一的歸屬程度。隨後Buckley提出了模糊複述的概念[1-3]，直到2002年，Ramot et al.[25]提出了複數歸屬度型態的模糊集合，可以擁有更豐富的資訊。為了使日後應用更加廣泛，我們希望可以透過一個觀念使得歸屬程度比以往模糊集合更加豐富。

球式複數模糊集合是本論文所提出的原創概念，透過此概念可以將一筆資料轉換成多個複數型態的歸屬程度，以便之後模型可以一次預測多個目標。首先，將原高斯函數得到的歸屬程度置放於半徑為1的球式複數模糊集合內(Fig. 1)，即可得到一組空間向量，其成分表示如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |

其中，為原高斯函數(25)的歸屬程度;;。藉由的拆解，可得到至少四組的複數型態歸屬程度，包含了以下的歸屬程度。

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |

其中，。

1

1. 球式複數模糊集。為高斯函數的歸屬程度，可透過與平面的夾角及計算出在各維度上的投影。一個球式複數模糊及他的歸屬度訊息，是由一個球式空間向量所攜帶，會隨著輸入再求空間裡改變。

## Multi–Target Feature Selection

特徵挑選不僅能刪去負面的資訊來源，更有助於減輕模型的運算負擔，故是資料前處理中重要的一環。面對多個目標時的特徵挑選，更需要謹慎的處理，才能帶來正面的效果。本論文將同時預測多個目標，故使用夏農資訊熵[29]概念，並參考多目標特徵選取方法[23]，實作特徵挑選，最後從挑選後的特徵中取得訓練資料。

在1948年，學者Shannon則提出了資訊熵[29]的概念，熵定義為資訊內容其不確定性的量值，若資訊的隨機性越高，則資訊熵值會越高。對於某一個隨機變數*X*，資訊熵的定義如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，是隨機變數的資訊熵；是事件的發生機率密度；則被視為的資訊混亂度。若大於，則部分會是負數，會影響到整體的期望值，所以我們對公式做了一些更改，更改後的公式如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

其中，為很小的正值。

由於我們特徵的選擇是針對目標，所以我們透過資訊熵的概念，計算每個特徵與目標之間的影響資訊量(Influence information) [23]，公式如下。

|  |  |
| --- | --- |
| , |  |

其中，為隨機變數對隨機變數的影響資訊量為在事件值為正時的隨機變數為在事件值為負時的隨機變數為隨機變數以及隨機變數的互資訊[23]為隨機變數以及隨機變數的互資訊互資訊的定義公式如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

其中，為目標的資訊熵;為事件為正數時所對應的隨機變數的資訊熵;為事件為負數時所對應的隨機變數的資訊熵;條件式資訊熵公式如下。

|  |  |
| --- | --- |
| , |  |
| , |  |
| , |  |
| , |  |

其中，為事件為正的變數;為事件為負的變數;為事件為正數時的機率密度;為事件為正數時所對應的事件*y*的機率密度;為事件為負數時的機率密度;為事件為負數時所對應的事件*y*的機率密度。

透過上述影響資訊量[23]的公式，可以得到每個特徵變數對每個目標的影響資訊量，為了方便計算影響資訊量，我們可以將所有特徵變數以及目標組合成一個資料矩陣 (Data matrix, DM)，如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，為第組候選特徵，，為一個特徵的總筆數，為特徵總數量; 為第個目標變數，，為一個目標的總筆數，為目標的總數量。

我們利用資料矩陣每行的特徵資料，與其他行做影響資訊量的運算，透過特徵與特徵之間以及對於第個目標的影響資訊量整理出第個目標的影響資訊矩陣 (Influence information matrix, IIM)。表示如下。

|  |  |
| --- | --- |
| , |  |

其中，為第個特徵變數;為第個目標變數; 為特徵的總數量;為特徵變數對特徵變數的影響資訊量， 且。

而後可依據影響資訊矩陣裡的影響資訊量做多目標的特徵選取，步驟如下。

Step 1 : 算出第個特徵對第個目標的資訊增益量 (Selection gain)標記為，其中，為第個特徵變數;為第個目標變數。資訊增益量公式如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，為對的影響資訊量;為第個已選特徵池(Selected pool, SP), ; 是第個已選特徵池中，第個元素； 為對中已存在特徵的冗餘資訊量(Redundancy information)。冗餘資訊量公式如下。

|  |  |
| --- | --- |
| , |  |

其中，代表第個已選特徵池內的特徵個數;為對內的第個特徵變數的影響資訊量;為內的第個特徵變數對的影響資訊量。經過上述計算若大於0，則將特徵加入第個已選特徵池 中。

Step 2 :無論重疊與否，將所有已選特徵池中出現過的特徵變數記錄下來，儲存成。其中，為目標變數個數;，是中第個特徵變數。計算每個特徵出現在所有已選特徵池的次數，標記為。

Step 3 :透過即可計算覆蓋率，公式如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

計算的平均，標記為。

Step 4 :累加每個已選特整池裡，特徵的資訊增益量。

|  |  |
| --- | --- |
| , |  |

計算的平均標記為。

Step 5 :根據累加後的資訊增益量和覆蓋率，計算出特徵的有效貢獻量。

|  |  |  |
| --- | --- | --- |
|  | , |  |

Step 6 :測試中所有的特徵變數，若，則將累加。

Step 7 :設定上下界，標記為和，透過上下界找出，表示最後選取的特徵數目。若介於上下界間，則將設定成;若小於下界，則將設定成;若大於上界則將設定成。

Step 8 :將排序，並選取前個特徵變數加入最後特徵池(Final pool, FP)中，當作多目標的特徵挑選結果。

## Structure Learning

透過資料驅動的概念，模型的結構會透過資料自動產生。本論文藉由機器學習，使訓練資料可以更有邏輯的應用到模型建造中，此外結構學習中的結果，也會成為之後參數學習的一部分。在本研究中，會將這些不同輸入維度的訓練資料，透過減數分群演算法[5]進行分群。並將分群後的群中心配合每個維度的標準差形成模糊集，各個維度的模糊集個數總和即為第1層神經元的數量，本研究採用高斯型態的模糊集合，高斯函數的公式如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，為輸入變數，和為群中心和標準差的參數。基於各個輸入維度的模糊集，我們可以組成個區塊。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，為第個輸入維度的分群個數，亦即模糊集個數。以第個區塊的組成為例，如下。

|  |  |
| --- | --- |
| , |  |

，其中，為第個輸入的語意變數;為第個輸入變數，;為第個區塊中第個輸入語意變數的模糊集合，使用高斯函數建構，如(25)。



1. 模糊集合輸入空間(2維)。兩個輸入維度，各分出3群則會形成共9塊區域，其中z軸為該區的資料密度。



1. 資料密度總和。可看出部分區塊資料密度總和較高，代表用於建造模型較有利。

為了模型的運算效率，以及降低模型運算負擔，我們將會篩選出若干個較重要的區塊，成為第2層中的神經元數量。若以兩個輸入維度為例子可得到如圖(Fig. 2)的區塊，而後將透過資料密度的概念，將資料灑入區塊中，並計算資料密度量，累加資料密度量後可得到如圖(Fig. 3)，從中可挑選資料密度較高的區塊，作為第2層神經元。詳細步驟如下:

Step 1:從各個輸入維度與模糊集可以得到每個區塊的資料密度量，以第個區塊為例，資料密度量公式如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，為第個輸入維度的第筆資料;為第個區塊中第個輸入維度的模糊集。

Step 2:將每個區塊的資料密度量累加標記為，公式如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，為資料總筆數。計算平均值標記為，標準差標記為。

Step 3:查看每個區塊，若，則將累加。設定上下界，標記為和，透過上下界找出，表示最後選取的區塊數目。本研究中所有實驗皆設定為15，皆設定為4。若介於上下界間，則將設定成;若小於下界，則將設定成;若大於上界則將設定成。

Step 4:將排序，並保留前個區塊，當作之後模型第2層的神經元。

本研究採用傳統的if-then 規則建構模型，因此前鑑部的神經元個數會與後鑑部個數一樣，透過上述區塊挑選的方法，可以得出個第2層神經元，最後，會以相同個數建構出個第4層神經元。後鑑部神經元為T-S神經元，由T-S 函式構成，T-S公式如下。

|  |  |
| --- | --- |
| , |  |

其中，{}是第個T–S神經元的參數，是第個輸入。

透過結構學習的過程，本論文以客觀的方法建構模型架構，模型運算過程以及詳細說明將在下個小節探討。

## Model Structure and I/O Relationship

本研究的模型為一個六層的類神經網路。訓練資料集合標記為，為資料總筆數，是*-*by*-*1的輸入向量，為輸入維度數量;為-by-1的目標向量，為複數型態目標的數量。透過模型可以得到輸出。

模型是透過條T-S模糊規則組成，每條規則是由前鑑部(If part)與後鑑部(Then part)所結合，其規則形式如下：

|  |  |
| --- | --- |
| , |  |

規則數；為模糊系統的輸入變數；為第條規則的複數模糊集；則是輸入的語意變數。複數模糊集的參數即是前鑑部(If part)參數，為後鑑部(Then part)參數。此複數類神經糢糊模型可轉為一個六層架構的類神經網路模型。

**Layer 0:** 此層為輸入層，是原始資料透過多目標特徵選取後，將最後挑出的特徵當作訓練資料，我們將時間序列第個點的輸入向量標記如下。

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Layer 1:** 此層為SCFS神經層，透過前面結構學習的分群，可在不同維度上建構數個模糊集，每個不同維度的輸入都可經由模糊集得到歸屬程度。透過球式複數模糊集合可得到多組複數型態的歸屬程度，不同的歸屬程度可以給不同的模型輸出做應用，以達到多目標預測的效果，透過球式複數模糊集的公式(1)-(7)，從中可得到歸屬度向量，如下。

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Layer 2:** 此層為前提式神經層，經過前面的結構學習，我們可以篩選出個對模型較有用的區塊並建構神經元，由於神經元的輸入為上一層的歸屬程度，且輸出為每個輸入維度的歸屬程度相乘結果，故稱之為神經元，每個神經元的輸出為該神經元的啟動強度，由於本研究採用球式複數模糊集合，因此每個神經元的輸入會是向量型態，輸出亦然如此。

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

，為的乘積，為第個神經元中第個維度歸屬度向量的第項歸屬程度，。

**Layer 3:** 此層為正規化層，會將個輸入向量中的各個元素正規化，之後將結果以向量型態輸出，輸出如下。

|  |  |
| --- | --- |
| , |  |
|  |  |

其中，為第個神經元中第個元素正規化後的值，。本研究中使用的是複數模糊集，因此輸入是複數型態，故輸出也是複數型態。

**Layer 4:** 此層為T–S層，經過此層的運算可以得到個模型輸出，公式如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，為第個輸出；{}是第個T–S神經元的參數。

**Layer 5:** 此層為輸出層，將上一層得到的個模型輸出加總，即為模型的輸出。

|  |  |  |
| --- | --- | --- |
|  |  |  |

## Parameter Learning

根據分治法(Divide-and-conquer)的概念，我們將使用不同的機器學習演算法，對各層的參數優化，以便更容易找到最佳解。前鑑部的參數優化會使用PSO以及ABCO，後鑑部的參數優化會使用RLSE，期待透過混合型演算法PSO-RLSE以及ABCO-RLSE帶來優異的效能表現。

1. Particle Swarm Optimization

粒子群演算法是由J. Kennedy et al. [8] 1995年所提出的優化演算法。其原理類似鳥群尋找食物，除了粒子自身提供的資訊，慣性以及群體智慧中全群最佳位置也會被使用在演算法中，用以調節粒子的速度，此演算法特性為收斂快速，公式如下。

|  |  |
| --- | --- |
| , |  |
| , |  |

其中，為第回合時第個粒子的位置，為第回合時第個粒子的速度，為第回合時第個粒子的最好位置，為第回合時全部粒子中最好的位置，為PSO的參數，、為介於0到1的隨機數。在本研究中，粒子的位置代表前鑑部的參數，其中包含了每個維度的分群中心、標準差以及相位頻率參數。

1. Artificial Bee Colony Optimization

人工蜂群演算法為Karaboga學者所提出 [17]，原理類似蜜蜂尋找食物來源的概念，其特性包含群體智慧與隨機性，如蜜蜂傳達食物位置時會彼此透過搖擺舞溝通，而搖擺舞所指示食物位置會有偏差帶有隨機性。此演算法中，總共有三種蜜蜂，包括工蜂 (Employed bee)、觀察蜂 (Onlooker bee)以及偵查蜂 (Scout bee)。其中，工蜂負責尋找食物源的位置，然後以跳舞傳達食物源的收益訊息，每次在傳達時會帶有隨機性，代表著整體蜜蜂大維度的搜尋；觀察蜂負責在其中一個食物源附近搜尋，首先會觀察各個食物收益度，並以輪盤法選擇一個食物源，到食物源的鄰近周遭進行搜尋，代表著食物源附近小維度的搜尋；偵查蜂作用為當食物源經過一定的開發次數後，若食物源的收益度沒有改善時，則會派出偵查蜂取代食物源，意味著隨機的探索一個全新的食物源，可以避免演算法陷入區域最佳解中。其步驟與真實的蜜蜂找尋食物不盡相同，演算法步驟如下:

1. 隨機尋找其中一隻工蜂，並透過公式形成新位置，此位置為該次迭代的食物源位置，公式如下。

|  |  |
| --- | --- |
| , |  |

其中，為第個食物源的第個維度;為第隻工蜂的第個維度;為其他隨機工蜂的第個維度的值。

1. 觀察蜂使用輪盤法 (Roulette method)挑選一食物源，其中食物源收益越好越容易被選中，輪盤機率公式如下。

|  |  |
| --- | --- |
|  |  |

其中，為第個食物源被選中的機率;為收益程度，本研究將成本函數的倒數視為收益程度;為食物源的總數目。

1. 派出每隻觀察蜂在剛剛被選中的食物源位置附近搜尋，公式如下。

|  |  |
| --- | --- |
|  |  |

其中，為第隻觀察蜂的第個維度;為被選中的食物源第個維度;為被其他隨機的食物源第個維度的值，若觀察蜂位置比被選中食物源位置好，則取代食物源位置。

1. 判斷每隻工蜂是否已經達到限制回合都未更新，若為真則派出偵查蜂取代，偵查蜂位置產生公式如下。

|  |  |
| --- | --- |
|  |  |

其中，為第個食物源的第個維度;為所有食物源中第個維度的最大值;為所有食物源中第個維度的最小值。

1. 重複Steps 2~4，直到反覆運算結束。
2. Recursive Lest squares estimation

在本論文使用遞迴式最小平方演算法 (Recursive least squares estimation, RLSE)[15]更新T–S神經元參數，RLSE方法在更新參數時是利用每筆資料，不斷的更新比起一次接收所有資料的LSE方法更加有效，一般來說LSE問題可以被視為一個線性的問題，如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，y是目標;u是模型的輸出;{}是u已知的方程式;{, =1,2,…,m}是我們估計的未知參數，則是整個模型的誤差LSE的問題也可以被寫成矩陣的方式表達，如下。

|  |  |  |
| --- | --- | --- |
|  | , |  |

其中，

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |
|  | , |  |
|  | , |  |

是輸入的矩陣，是我們估計的未知參數矩陣，是目標矩陣，是誤差的向量。要最佳化，可透過RLSE的等式運算。

|  |  |  |
| --- | --- | --- |
|  | , |  |
|  | , |  |

其中，是遞迴次數，{}，為資料總筆數，是的第行，再開始RLSE演算法時，會設定為0，則設定為，為一極大整數，為單位矩陣。

混合型演算法中各部分參數優化會交由不同演算法負責，本研究使用到PSO-RLSE以及ABCO-RLSE兩種混合型演算法，PSO和ABCO負責前鑑部的參數學習，RLSE則負責線性的T-S function 參數學習。混合演算法與模型計算的流程如下:

1. 準備訓練資料及測試資料。
2. 以前鑑部演算法(PSO或ABCO)粒子位置作為模糊集參數，將訓練資料帶入模型，並計算每個神經元啟動強度。
3. 用RLSE更新T-S神經元的參數，RLSE算式中的和向量如下。

|  |  |
| --- | --- |
| , |  |
| , |  |
| , |  |
| , |  |

其中。因為多目標預測中，正規化後的啟動強度為一向量，使為一矩陣，因此在原本的公式(52)中，利用單位矩陣取代原本的常數項1，改良後公式如下。

|  |  |
| --- | --- |
| , |  |

1. 更新完所有參數後，計算出模型的輸出。
2. 計算成本，更新前鑑部演算法粒子的位置及相關數據。
3. 重複Steps 2~5，直到迭代結束。

## Methodology for financial application

為了評估模型是否對投資有實質上的幫助，使用成本函數是不夠的，因為從中無法看出是否有利潤，只能瞭解模型的配適率，而配適率高並不代表投資效益高，因此本研究將預測出來的收盤價配合投資策略[35]，進一步決定要買進或是賣出，買進與賣出公式如下。

|  |  |
| --- | --- |
| 買進: if  , |  |
| 賣出: if  , |  |

其中，為門檻參數，同時也代表股票的漲跌；為模型的輸出，意即預測日的收盤價格;為日實際的收盤價格。若預測明天的收盤價高於今天實際收盤價，代表必須買進;若預測明天的收盤價低於今天實際收盤價，代表模型預測明天會跌，所以要儘快賣出。

為了使得買賣更加謹慎，本研究提出另一買賣策略，策略步驟如下。

1. 使用公式(43)-(44)判定是否買或賣，接著進入第二階段
2. 計算第天的過去30天漲跌平均值，做為第二階段判斷標準。經過多次測試以過去30天平均最佳。
3. 若，且則買進；若，且，則賣出。
4. 所有交易日需通過兩階段評判，若其中一階段未通過則不操作。

計算利潤的方式，則透過今天實際的收盤價與隔天實際的收盤價做運算，公式如下。

|  |  |
| --- | --- |
|  |  |
|  |

其中，為利潤，為策略為買的總天數;為策略為賣的總天數;代表第天的真實收盤價。

透過上述投資策略以及利潤公式，我們可以計算出整個模型所帶來的利潤值，並大致模擬出此模型運用到真實世界的效果，本研究將會在每個實驗中秀出上述兩種策略的利潤值與其他參數與現有文獻做比較。

# Expeimentation

本研究總共有3個實驗，實驗一為雙目標的預測，使用的目標為2000年台灣股票加權指數 (The Taiwan stock exchange capitalization weighted stock index, TAIEX)以及恆生指數 (Hang Seng index, HSI)；實驗二為四目標的預測，檢驗模型透過複數型態的歸屬程度可行性，使用複數輸出預測四個目標，第一組輸出實數部分負責第一個目標，虛數部分負責第二個目標；第二組輸出實數部分負責第三個目標，虛數部分負責第四個目標，同時預測TAIEX道瓊工業指數 (Dow Jones industrial average index, DJI)、納斯達克 (National association of securities dealers automated quotation, NASDAQ)以及標準普爾500 (Standard and Poor’s 500, S&P 500)。實驗三也是四個目標的預測，使用到兩組複數型態的輸出，其中包含了蘋果、IBM、DELL和微軟的股票收盤價。第三個實驗為四個有名的科技公司，除此之外，其餘實驗為非常著名的股票指標，其中，TAIEX為台灣上市的股票中經過加權計算出的指標，代表著台灣上市股票的波動；HSI是以反映香港股市行情的重要指標，指數由五十隻恆指成份股的市值所計算；DJI涵蓋著財務等9大產業，為一股價加權指標；NASDAQ為超過三千檔股票所組合成的市值加權指標，大多以科技產業為例；S&P500為美國前500大公司的市值加權，當中包含IT等11個產業。這些指標代表著一個國家甚至全球股票的趨勢，故能精準預測可為投資者帶來不少的幫助。

為了與其他文獻方法比較，我們將透過誤差指標對模型評估以及計算模擬投資後的利潤。成本函數 (Cost function)與評估指標皆使用均方根誤差(Root mean square error, RMSE)，公式如下。

|  |  |
| --- | --- |
| , |  |
| , |  |

，為資料總筆數；為模型第筆資料的誤差向量;為第筆目標向量；為第筆模型輸出向量；為埃爾米特共軛(Hermitian transpose)，意即轉置矩陣後，並對矩陣元素做共軛運算。

在結構學習部分，本研究中所有實驗的規則數，意即第2層神經元數目上限皆設定為15，下限皆設定為4，故模型第2層神經元個數會介於4~15之間。

在模擬投資方面，所有實驗的買賣策略門檻參數會介於0至0.1之間，因為門檻參數代表著股票的波動，而台灣股票的漲跌幅為10%，因此會從此區間中挑選出最佳的，從0開始每次以0.001增加，直至0.1，所有實驗會透過訓練資料找出利潤最佳的門檻參數，並透過測試資料計算利潤，若訓練階段利潤為0，則測試資料利潤設為0，亦即不參與投資。

## Example 1—Doublee Time Series of Daily Taiwan Stock Exchange Capitalization Weighted Stock Index

在這個範例中，我們也是對現實世界的時間序列資料來驗證模型的效能，四個目標使用的資料為TAIEX和HSI，每年的收盤價。本論文對2002年的收盤價預測並與其他文獻方法比較效能，資料總共248筆，每年的前十個月的資料當作訓練資料，剩餘的當作測試資料。訓練資料會萃取出各30個特徵，共60個特徵，每個特徵資料為205筆，特徵排序為TAIEX、HSI。本篇論文所提出的模型可以一次有多個複數型態的輸出，故可以預測多個目標，我們在範例中使用的第一個複數目標實數部分為TAIEX的收盤價，需數部分則為HSI的收盤價。結構學習後的參數，如表I所示，前鑑部透過區塊挑選，從625個降低至15個，大量地減少模型大小。機器學習設定，如表II所示。為了檢驗模型的穩定度，本論文跑了10次模型，效能結果如表III所示，投資利潤表如表IV所示。本次範例的結果將與其他論文所提的方法做比較，效能比較結果如表V所示，投資利潤比較結果如表VI所示。學習曲線如Fig. 4.所示。目標與模型輸出的結果，如Fig. 5.所示。

實驗一模型設定

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | {3, 3, 3, 3} |
| Number of outputs (complex-valued)\* | 2 |
| Type of premises | SCFS |
| Number of premises (after selection) | 15 |
| Number of premise parameters | 48 |
| Number of consequences | 15 |
| Number of consequence parameters | 75 |

\* 每一個複數目標的實部與虛部，分別包含兩個實數目標

機器學習參數設定

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | 藉由第二章所提SC演算法決定 |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 75x1 zero vector |
|  | **I** |
| **I** | 75x75 identity matrix |

十次測試效能表 (實驗一, RMSE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |
| TAIEX | HSI | TAIEX | HSI |
| 1 | 96.96 | 83.05 | **85.25** | **80.62** |
| 2 | 90.68 | 86.59 | 87.82 | 86.38 |
| 3 | 104.11 | 186.15 | 36748 | 32343 |
| 4 | 109.61 | 112.37 | 4352.6 | 2340.3 |
| 5 | 87.71 | 93.62 | 88.69 | 84.88 |
| 6 | 211.39 | 313.15 | 84.30 | 85.56 |
| 7 | 94.29 | 89.77 | 2093.3 | 1365.1 |
| 8 | **89.77** | **85.72** | 90.01 | 85.14 |
| 9 | 183.06 | 221.61 | 90.07 | 85.68 |
| 10 | 83.27 | 103.40 | 91.43 | 89.47 |

模擬投資利潤表 (實驗一)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| 利潤 | | 利潤 | | 利潤 | | 利潤 | |
| Best | 0.07 | | 0.05 | | 0.047 | | 0.052 | |
| 1 | -1786.5 |  | 643.46 |  | 129.68 |  | 25.08 |  |
| 2 | 88.62 |  | 806.72 |  | 682.75 |  | 157.62 |  |
| 3 | -463.39 |  | -36.57 |  | 475.26 |  | 202.11 |  |
| 4 | -914.08 |  | 1664.9 |  | 64.19 |  | 82.35 |  |
| 5 | -936.58 |  | -355.81 |  | 173.98 |  | **961.18** |  |
| 6 | -173.21 |  | -689.96 |  | 14.83 |  | -3.451 |  |
| 7 | -1070.4 |  | -911.59 |  | **767.09** |  | 59.93 |  |
| 8 | **2524** |  | -1640.9 |  | 64.19 |  | -30.80 |  |
| 9 | -376.84 |  | 1620.8 |  | 119.86 |  | 514.65 |  |
| 10 | 124.96 |  | **2469.8** |  | 364.09 |  | 478.03 |  |
| 平均 | -298.34 | | 357.09 | | 350.48 | | 244.67 | |
| 標準差 | 1151.83 | | 1308.77 | | 345.16 | | 315.57 | |
| 最大值 | 2524 | | 2469.8 | | 767.09 | | 961.18 | |
| 最小值 | -1786.5 | | -1640.9 | | 14.83 | | -30.8 | |

\*為本研究提出投資策略方法的結果

多目標預測效能比較表 (RMSE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| TAIEX | HSI |  | TAIEX | HSI |
| Chen [35] | - | - |  | 75 | 187 |
| Yu [35] | - | - |  | 101 | 170 |
| AR(1) [35] | - | - |  | 66 | 105 |
| SVR [35] | - | - |  | 66 | 107 |
| ANFIS [35] | - | - |  | 65 | 106 |
| ANFIS (EMD) [35] | - | - |  | 52 | 97 |
| PSO-RLSE (proposed) | **99.16** | **98.19** |  | **89.77** | **85.72** |
| ABCO-RLSE (proposed) | **98.77** | **97.12** |  | **85.25** | **80.62** |

模擬投資利潤比較表(實驗一)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Chen [4] | Yu [4] | SR+ANFIS [4] | SR+SVR [4] | Elman [4] | Cheng et al. [4] | PSO-RLSE | ABCO-RLSE |
| Best | 0.012 | 0.012 | 0.012 | 0.012 | 0.012 | 0.012 | 0.07 | 0.05 |
| 利潤(TAIEX) | 0 | 0 | 0 | 0 | 0 | -231.02 | **1284.2** | 847.8 |
| 利潤(HSI) | -1471 | -1368 | -602.94 | 190.71 | **2342** | 1793.12 | 1239.7 | 1622.0 |



(a)



(b)

1. 機器學期曲線 (a) PSO-RLSE (b) ABCO-RLSE



(a)



(b)



(c)



(d)

1. 實際數值和模型輸出(a) PSO-RLSE (TAIEX) (b) PSO-RLSE (HSI) (c) ABCO-RLSE (TAIEX) (d) ABCO-RLSE (HSI)

## Example 2— Quadruple Time Series of Daily Dow Jones Industrial Average Index

在這個範例中，四個目標使用的資料為TAIEX、DJI、NASDAQ以及S&P500。我們預測四個目標2001年的收盤價，並與其他文獻方法比較效能，每年前十個月的資料當作訓練資料，剩餘的當作測試資料，訓練資料總共為181筆，測試資料為66筆。本論文將訓練資料萃取出各30個特徵，4個目標共120個特徵，特徵排序依序為TAIEX、DJI、NASDAQ和S&P500。本論文所提出的模型可以一次有多個複數型態的輸出，在此範例中，共有兩個複數型態目標，第一個目標實數部分為TAIEX收盤價，虛數部分為DJI收盤價，第二個目標實數部分為NASDAQ收盤價，虛數部分為S&P500收盤價。結構學習部分，透過區塊挑選，前鑑部從81個降低至9個，明顯縮減模型大小，降低模型參數。模型參數如表VII所示。機器學習設定，如表VIII所示。本次範例的結果除了PSO-RLSE與ABCO-RLSE的比較也與其他論文所提的方法做比較，像是ANFIS[22]，CNFS-ARIMA[22]，RBF network[22]和SVR[22]。其中除了SVR之外的模型，皆可以同時預測兩個實數目標。所以我們將使用模型的第一組複數型態的輸出與其他論文做比較，效能比較結果如表IX所示，投資利潤比較如表X所示。為了檢驗模型的穩定度，本論文跑了10次模型，效能如表XI所示，投資利潤如表XII所示。模型的機器學習曲線，如Fig. 6.所示; 目標與模型輸出的結果，如Fig. 7.所示。

範例二模型設定

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | {3, 3, 3, 3} |
| Number of outputs (complex-valued)\* | 2 |
| Number of premises (before selection) | 81 |
| Type of premises | SCFS |
| Number of premises (after selection) | 9 |
| Number of premise parameters | 48 |
| Number of consequences | 9 |
| Number of consequence parameters | 45 |

\* 每一個複數目標的實部與虛部，分別包含兩個實數目標

機器學習參數設定

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | 藉由第三章所提SC演算法決定 |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 45x1 zero vector |
|  | **I** |
| **I** | 45x45 identity matrix |

多目標預測效能比較表 (RMSE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| TAIEX | DJI |  | TAIEX | DJI |
| SVR (two models, each with single output) [22] | - | - |  | 162.46 | 101.44 |
| ANFIS (two models, each with single output) [22] | - | - |  | 147.36 | 105.56 |
| ANFIS (one model with two outputs) [22] | - | - |  | 151.62 | 128.20 |
| RBF (two models, each with single output) [22] | - | - |  | 134.32 | 106.33 |
| RBF (one model with two outputs) [22] | - | - |  | 137.58 | 181.79 |
| CNFS(4)-ARIMA (one model with two outputs) [22] | - | - |  | 115.82 | 103.06 |
| PSO-RLSE (proposed) | **92.92** | **91.50** |  | **96.51** | **92.18** |
| ABCO-RLSE (proposed) | **93.69** | **91.06** |  | **87.92** | **91.95** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 投資利潤比較表 | | | | | | | | |
|  | Chen [35] | Yu [35] | AR(1) [35] | SVR [35] | ANFIS [35] | Wei [35] | PSO-RLSE | ABCO-RLSE |
| Best | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.047 | 0.012 |
| 利潤(TAIEX) | -92 | -73 | 671 | 202 | 686 | **795** | 745.1 | 458.48 |



(a)



(b)

1. 機器學期曲線 (a) PSO-RLSE (b) ABCO-RLSE



(a)



(b)



(c)



(d)

1. 實際數值和模型輸出 (a) PSO-RLSE (TAIEX) (b) PSO-RLSE (DJI) (c) ABCO-RLSE (TAIEX) (d) ABCO-RLSE (DJI)

十次測試RMSE效能表 (實驗二)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |
| TAIEX | DJI | TAIEX | DJI |
| 1 | 98.95 | 108.39 | **87.92** | **91.95** |
| 2 | 94.15 | 94.49 | 89.21 | 94.14 |
| 3 | 93.24 | 97.72 | 88.69 | 93.54 |
| 4 | 335.68 | 464.85 | 102.07 | 92.31 |
| 5 | 114.71 | 122.97 | 86.22 | 94.15 |
| 6 | **96.51** | **92.18** | 88.19 | 94.27 |
| 7 | 87.59 | 102.31 | 88.96 | 93.62 |
| 8 | 90.24 | 95.65 | 87.44 | 93.98 |
| 9 | 151.31 | 148.51 | 88.70 | 93.91 |
| 10 | 96.29 | 121.39 | 88.47 | 93.48 |

模擬投資利潤表 (實驗二)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| 利潤 | | 利潤 | | 利潤 | | 利潤 | |
| Best | 0.047 | | 0.012 | | 0.061 | | 0.012 | |
| 1 | -264.81 |  | -274.69 |  | 0 |  | 0 |  |
| 2 | -168.32 |  | 359.55 |  | 77.76 |  | 0 |  |
| 3 | 105.9 |  | 916.80 |  | -21.34 |  | 298.98 |  |
| 4 | -1636.8 |  | -862.40 |  | **363.20** |  | 82.62 |  |
| 5 | -378.64 |  | -274.69 |  | 277.37 |  | 0 |  |
| 6 | 532.59 |  | -274.69 |  | -34.84 |  | 0 |  |
| 7 | **987.72** |  | 359.55 |  | 10.77 |  | 0 |  |
| 8 | -235.02 |  | **916.80** |  | -10.67 |  | **298.98** |  |
| 9 | 412.35 |  | -274.69 |  | 0 |  | 0 |  |
| 10 | 199.98 |  | 359.55 |  | 114.98 |  | 0 |  |
| 平均 | -44.50 | | 95.11 | | 77.72 | | 68.06 | |
| 標準差 | 702.12 | | 579.65 | | 137.28 | | 124.40 | |
| 最大值 | 987.72 | | 916.80 | | 363.20 | | 298.98 | |
| 最小值 | -1636.8 | | -862.40 | | -34.84 | | 0 | |

\*為本研究提出投資策略方法的結果

## Example 3—Quadruple Time Series of Apple, IBM, Dell and Microsoft

在這個範例中，預測的目標為APPLE、IBM、Dell以及Microsoft，為期間2003年2月10號至2005年1月21號的股票收盤價，總共為492筆。為了和其他論文比較效能，我們用2003年2月10號至2004年9月10號的資料當作訓練資料，共400筆，剩餘的當作測試資料。本論文從訓練資料中個別萃取出30個特徵，4個目標共120個特徵，特徵排序依序為APPLE、IBM、Dell以及Microsoft。本篇論文所提出的模型可以一次有多個複數型態的輸出，故可以預測多個目標，此範例中使用的第一個複數目標實數部分為IBM收盤價，虛數部分則為APPLE收盤價，第二個目標實數部分為Dell收盤價，虛數部分為Microsoft收盤價。結構學習部分，透過區塊挑選，可以從原本的81個區塊挑選出8個區塊，降低模型中的參數數量，其餘模型設定如表XIII所示。機器學習設定，如表XIV所示。本次範例的結果將與其他論文所提的方法做比較，像是HiMMI [32]、ANN-GA-HMM-Interpolation [32]、ANN-GA-HMM-WA [32]、ARIMA [32]。所以我們將使用模型的第一組輸出的實數及複數部分做比較，結果如表XV所示。模擬投資利潤如表XVI所示。模型的機器學習曲線，如Fig. 8.所示;目標與模型輸出的結果，如Fig. 9.所示。

範例三模型設定

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | {2, 3, 3, 3} |
| Number of outputs (complex-valued)\* | 2 |
| Type of premises | SCFS |
| Number of premises | 8 |
| Number of premise parameters | 44 |
| Number of consequences | 8 |
| Number of consequence parameters | 40 |

\* 每一個複數目標的實部與虛部，分別包含兩個實數目標

機器學習參數設定

|  |  |
| --- | --- |
| **PSO** | |
| Swarm size | 64 |
| Iterations | 100 |
|  | 0.8 2.0 2.0 |
|  | Random in [0,1] |
| Initial position | 藉由第三章所提SC演算法決定 |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 40x1 zero vector |
|  | **I** |
| **I** | 40x40 identity matrix |



(a)



(b)

1. 機器學期曲線 (a) PSO-RLSE (b) ABCO-RLSE

十次測試MAPE效能表 (實驗三)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trials | PSO | | ABCO | |
| APPLE | IBM | APPLE | IBM |
| 1 | 1.7948 | 0.8043 | 1.7625 | 0.8032 |
| 2 | 1.7972 | 0.8039 | 1.7835 | 0.8042 |
| 3 | 1.7963 | 0.8033 | 1.8273 | 0.8002 |
| 4 | 1.7954 | 0.8043 | 1.7818 | 0.8015 |
| 5 | **1.7776** | **0.8033** | 1.7922 | 0.8033 |
| 6 | 1.7921 | 0.8044 | 2.0309 | 0.8002 |
| 7 | 1.7942 | 0.8064 | **1.7602** | **0.8036** |
| 8 | 1.7967 | 0.8043 | 1.7946 | 0.8031 |
| 9 | 1.7920 | 0.8042 | 1.8567 | 0.8046 |
| 10 | 1.7927 | 0.8041 | 1.7758 | 0.8027 |

模擬投資利潤表 (實驗三)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trials | PSO-RLSE | | ABCO-RLSE | | PSO-RLSE\* | | ABCO-RLSE\* | |
| 利潤 | | 利潤 | | 利潤 | | 利潤 | |
| Best | 0.011 | | 0.012 | | 0.021 | | 0.021 | |
| 1 | -4.5715 |  | -10.08 |  | 0 |  | -0.359 |  |
| 2 | 0.94 |  | 1.361 |  | 0 |  | 0 |  |
| 3 | -14.06 |  | -8.678 |  | **1.022** |  | -2.764 |  |
| 4 | -3.37 |  | **2.061** |  | 0 |  | -1.441 |  |
| 5 | **12.22** |  | -8.678 |  | -1.163 |  | **-2.764** |  |
| 6 | 8.72 |  | 2.061 |  | 0.176 |  | -1.441 |  |
| 7 | -10.18 |  | -0.3700 |  | -0.334 |  | 0 |  |
| 8 | 0.94 |  | -8.678 |  | 0 |  | -2.764 |  |
| 9 | -14.06 |  | 2.061 |  | 1.022 |  | -1.441 |  |
| 10 | -3.37 |  | -0.37 |  | 0 |  | 0 |  |
| 平均 | -2.6794 | | -2.9308 | | 0.072 | | -1.2974 | |
| 標準差 | 8.8075 | | 5.3385 | | 0.626 | | 1.1758 | |
| 最大值 | 12.22 | | 2.061 | | 1.022 | | 0 | |
| 最小值 | -14.06 | | -8.678 | | -1.163 | | -2.764 | |

\*為本研究提出投資策略方法的結果



(a)



(b)



(c)



(d)

1. 實際數值和模型輸出(a) PSO-RLSE (APPLE) (b) PSO-RLSE (IBM) (c) ACO-RLSE (APPLE) (d) ACO-RLSE (IBM) (e) ABCO-RLSE (APPLE) (f) ABCO-RLSE (IBM)

多目標預測效能比較表 (MAPE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **MAPE** | | | | |
| Training phase | |  | Testing phase | |
| APPLE | IBM |  | APPLE | IBM |
| HiMMI [32] | - | - |  | 2.8373 | 1.2186 |
| ANN-GA-HMM-Interpolation [32] | - | - |  | 2.1649 | 1.0555 |
| ANN-GA-HMM-WA [32] | - | - |  | 1.9247 | 0.8487 |
| Bayesian ANN [32] | - | - |  | 1.9688 | 0.7441 |
| ARIMA [32] | - | - |  | 1.8009 | 0.9723 |
| PSO-RLSE (proposed) | **1.7840** | **1.1960** |  | **1.7776** | **0.8033** |
| ABCO-RLSE (proposed) | **1.7654** | **1.1958** |  | **1.7602** | **0.8036** |

# Discussion

透過三組實驗，我們可以發現本論文所提出的一系列方法有著多目標的預測能力。在預測的表現上，從實驗一可發現雙目標的預測效能勝過其他文獻所提出的方法，代表本論文模型即使一次預測兩個目標也能有著優異的表現，透過實驗二及實驗三一次預測四個目標，更再次的證明了模型的預測能力。透過多目標特徵挑選，可從超過100個特徵中，萃取出對所有目標都有益的特徵，有效率的應用訓練資料。在結構學習部分，第一個實驗透過區塊篩選，從625個區塊中挑選出有用的15個區塊做為模型前鑑部神經元，第二個與第三個實驗則從81個區塊中作區塊的縮減，透過客觀的方式降低模型的參數量，有效地控制模型大小，降低模型運算的負擔，是此模型的優點之一。在參數學習部分，可從效能表現中看出，混合式的演算法在搜尋參數時，有著較良好的表現，配適率比其他文獻方法更佳，當中，ABCO-RLSE又比PSO-RLSE表現更良好，顯現出ABCO演算法的優點，透過觀察蜂的局部搜索以及偵查蜂的跳脫機制，使得整體表現優於PSO。最後，模擬投資部分，本論文提出的模型，在虛擬投資中利潤在部分實驗勝過過去文獻，而輸給其他文獻的實驗相差也不大，且可從10次效能比較表與投資利潤表中發現，效能與投資利潤並無正相關。本論文所提出的投資策略，提高投資時決策的標準，更加謹慎的判斷每日交易，從結果中可發現，投資策略所賺取的利潤有所提高，且各實驗中標準差有降低的趨勢，意即投資時更佳的穩定，可以此提供給投資者做決策參考。

# Conclusion

本論文提出混合型演算法ABCO-RLSE，相對於PSO-RLSE有著更佳的效能表現，兩者皆擁有多目標的預測能力，從實驗中可看見雙目標與四目標皆能同時預測，透過球型複數模糊集合概念，甚至可以一次預測更多目標。多目標特徵挑選可以根據不同的資料萃取原始資料中有用的資料，並且控制進入模型的資料大小，對於大量資料是很重要的過程。結構化學習的部分，使用資料驅動的概念，將輸入的資料自動地根據資料做調整，在面臨不同的資料時模型能適應性的生成不同結構。從3個實驗中可以發現不同資料都能在此模型中，被有效的預測。也代表此研究中的兩種演算法配合RLSE的混合方法，有著一定的水準。最後，本論文所提出的投資策略，有助於降低投資時的風險，可為投資者提供一客觀參考數據。

機器學習的部分可能會受限於演算法本身的特性，例如粒子群演算法(PSO)收斂快速、較容易掉入區域最佳解，意即在前幾個回合就已經接近最後一回合的RMSE，故在資料量龐大的情況下，所需搜尋維度增多，PSO效能就會較差，可能會限制住模型的整體發揮。而像是人工蜜蜂演算法(ABCO)，如果陷入區域最佳解，會在在限制(limit)回合數到達後重新搜尋，相對來說，跳脫區域最佳解機率較高，但因為重新搜尋的公式是透過最大最小值，所以所有蜜蜂的初始值很重要，會深深地影響到模型的訓練，較不穩定。在未來，可以透過不同的機器學習演算法結合SCFNS，像是Random Search [37]、蟻群演算法[6]等，以克服目前演算法所遇到的問題，亦可以透過CPSO[33]等演算法概念尋找最佳解。模擬投資方面，我們可以從實驗中瞭解到，每個演算法皆能賺取利潤，但並不是模型的誤差越小，所賺取的利潤越多，兩者並無正面相關，在策略當中又以門檻參數最為重要，故未來可以將門檻參數加入機器學習中，增加模型帶來的利潤。

##### ACKNOWLEDGMENT

##### REFERENCES

[1] J. Buckley, "Fuzzy complex numbers," *Fuzzy Sets and Systems,* vol. 33, no. 3, pp. 333-345, 1989.

[2] J. Buckley and Y. Qu, "Fuzzy complex analysis I: differentiation," *Fuzzy Sets and Systems,* vol. 41, no. 3, pp. 269-284, 1991.

[3] J. J. Buckley, "Fuzzy complex analysis II: integration," *Fuzzy Sets and Systems,* vol. 49, no. 2, pp. 171-179, 1992.

[4] C.-H. Cheng and J.-H. Yang, "Fuzzy time-series model based on rough set rule induction for forecasting stock price," *Neurocomputing,* vol. 302, pp. 33-45, 2018.

[5] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & fuzzy systems,* vol. 2, no. 3, pp. 267-278, 1994.

[6] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the first European conference on artificial life*, 1991, vol. 142, pp. 134-142: Paris, France.

[7] P. du Jardin, "Dynamics of firm financial evolution and bankruptcy prediction," *Expert Systems with Applications,* vol. 75, pp. 25-43, 2017.

[8] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, 1995, pp. 39-43: IEEE.

[9] Z. Feng, L. Jin, D. Tao, and S. Huang, "DLANet: a manifold-learning-based discriminative feature learning network for scene classification," *Neurocomputing,* vol. 157, pp. 11-21, 2015.

[10] N. Fumo and M. R. Biswas, "Regression analysis for prediction of residential energy consumption," *Renewable and Sustainable Energy Reviews,* vol. 47, pp. 332-343, 2015.

[11] H. Greenspan, B. Van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging,* vol. 35, no. 5, pp. 1153-1159, 2016.

[12] J. Heaton, N. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry,* vol. 33, no. 1, pp. 3-12, 2017.

[13] L. J. Herrera *et al.*, "Multigrid-based fuzzy systems for time series prediction: CATS competition," *Neurocomputing,* vol. 70, no. 13-15, pp. 2410-2425, 2007.

[14] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861,* 2017.

[15] C. T. S. a. E. M. J. S. R. Jang, *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence prentice hall*. Upper Saddle River, 1997.

[16] R. Ž. Jovanović, A. A. Sretenović, and B. D. Živković, "Ensemble of various neural networks for prediction of heating energy consumption," *Energy and Buildings,* vol. 94, pp. 189-199, 2015.

[17] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization,* vol. 39, no. 3, pp. 459-471, 2007.

[18] C. Li and T.-W. Chiang, "Complex neuro-fuzzy self-learning approach to function approximation," in *Asian Conference on Intelligent Information and Database Systems*, 2010, pp. 289-299: Springer.

[19] C. Li, T.-W. Chiang, J.-W. Hu, and T. Wu, "Complex neuro-fuzzy intelligent approach to function approximation," in *Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on*, 2010, pp. 151-156: IEEE.

[20] C. Li and T.-W. Chiang, "Complex fuzzy computing to time series prediction—A multi-swarm PSO learning approach," in *Asian Conference on Intelligent Information and Database Systems*, 2011, pp. 242-251: Springer.

[21] C. Li and J.-W. Hu, "A new ARIMA-based neuro-fuzzy approach and swarm intelligence for time series forecasting," *Engineering Applications of Artificial Intelligence,* vol. 25, no. 2, pp. 295-308, 2012.

[22] C. Li and T.-W. Chiang, "Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets," *IEEE Transactions on Fuzzy Systems,* vol. 21, no. 3, pp. 567-584, 2013.

[23] C. Li, "Multi-target feature selection," *unpublished,* 2017.

[24] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics,* vol. 5, no. 4, pp. 115-133, 1943.

[25] D. Ramot, R. Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *IEEE Transactions on Fuzzy Systems,* vol. 10, no. 2, pp. 171-186, 2002.

[26] V. Ravi, D. Pradeepkumar, and K. Deb, "Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms," *Swarm and Evolutionary Computation,* vol. 36, pp. 136-149, 2017.

[27] N. Rochester, J. Holland, L. Haibt, and W. Duda, "Tests on a cell assembly theory of the action of the brain, using a large digital computer," *IRE Transactions on information Theory,* vol. 2, no. 3, pp. 80-93, 1956.

[28] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, "Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction," *Fuzzy sets and systems,* vol. 157, no. 9, pp. 1260-1275, 2006.

[29] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review,* vol. 5, no. 1, pp. 3-55, 2001.

[30] I. Sugiarto and S. Natarajan, "Parameter estimation using least square method for MIMO Takagi-Sugeno neuro-fuzzy in time series forecasting," *Jurnal Teknik Elektro,* vol. 7, no. 2, pp. 82-87, 2008.

[31] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE transactions on systems, man, and cybernetics,* no. 1, pp. 116-132, 1985.

[32] J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications,* vol. 40, no. 14, pp. 5501-5506, 2013.

[33] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE transactions on evolutionary computation,* vol. 8, no. 3, pp. 225-239, 2004.

[34] J. Vieira, F. M. Dias, and A. Mota, "Neuro-fuzzy systems: a survey," in *5th WSEAS NNA international conference on neural networks and applications, Udine, Italia*, 2004.

[35] L.-Y. Wei, "A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting," *Applied Soft Computing,* vol. 42, pp. 368-376, 2016.

[36] P. Werbos, "Beyond regression: new fools for prediction and analysis in the behavioral sciences," *PhD thesis, Harvard University,* 1974.

[37] Z. B. Zabinsky, "Random search algorithms," *Wiley Encyclopedia of Operations Research and Management Science,* 2010.

[38] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research,* vol. 17, no. 1-32, p. 2, 2016.

[39] L. A. Zedeh, "Fuzzy sets," *Information and control,* vol. 8, no. 3, pp. 338-353, 1965.

[40] M. Zounemat-Kermani and M. Teshnehlab, "Using adaptive neuro-fuzzy inference system for hydrological time series prediction," *Applied Soft Computing,* vol. 8, no. 2, pp. 928-936, 2008.