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.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去做效能測試。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神經模糊集系統 (SCNFS)來對時間序列進行預測。透過一般的複數神經模糊集系統 (CNFS)，我們可得到一組複數型態的輸出，實數部分和虛數部分可針對不同目標做預測。本篇的模型對原本的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，即可同時針對兩個以上目標預測。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規則連結。此外，為了最佳化模型的預測結果，我們前鑑部使用不同的演算法像是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。最後我們將用三個實驗測試在不同的學習演算法下效能的差異性。

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[13]，圖像辨識方面[2, 21]、遊戲方面[14]等。上述這些有一部分屬於時間序列的預測，時間序列的預測是一個很重要的議題，因為他在真實世界中的應用範圍非常廣。過去已經有很多學者提出不同的方法來針對時間序列的預測問題，像是ARIMA[10, 11]、模糊理論、神經網路運算、神經模糊混合系統等。其中，最常被提出的就是神經模糊混合系統[4, 16, 18, 23]。

神經模糊混合系統 (NFSs)是基於神經網路 (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，現今火紅的深度學習即是神經網路的一種，在此領域中已有不少的研究[2, 3, 21]。

而神經網路的延伸，神經模糊混合系統 (NFSs)一直是被廣泛研究的模型，其中，類神經網路系統有所謂的IF-THEN規則，這些規則如同我們人類的經驗法則。通常可以將IF-THEN的規則結合模糊理論，使整體的架構更有彈性，我們稱之為神經模糊混合系統。如上述所提，神經模糊混合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這方面的研究，大多採用類神經網路為模型架構。

本實驗是採用神經模糊混合系統的作法，將整體模型效仿IF-THEN規則，建造成多層神經元架構。在模型實作方面，我們將模糊理論結合類神經網路系統，形成神經模糊混合系統，前鑑部使用高斯型態的球型複數神經模糊集，後鑑部則使用Takagi-Sugeno的線性函式[19]，前鑑部和後鑑部則透過IF-THEN規則結合。透過此模型和機器學習我們預期對時間序列的預測可以更加精準。

關於模糊集，在1965年由Zadeh學者，最先提出模糊集的概念[22]，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)[15]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)[8, 9]，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[8, 9, 11]。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)[17]，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[12]，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)[1]、Artificial Bee Colony Optimization (ABCO)[6] 兩種演算法和廣為人知的Recursive Least Square Estimator (RLSE)[5]進行參數優化，並將它們分別整合成三種方法我們稱之為PSO-RLSE [7]、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

## Sphere Complex Fuzzy Sets

過去的模糊集合概念，可以導出元素對集合一對一的歸屬程度。複數歸屬度型態的模糊集合[20]，可以擁有更豐富的歸屬程度，但為了使應用更廣泛，我們希望可以透過一個觀念使得歸屬程度更加的豐富。

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[39]概念，並參考多目標特徵選取方法[22]，實作特徵挑選，最後從挑選後的特徵中取得訓練資料。

熵一詞最早是由德國物理學家Clausius於 1854 年提出[12]，是一種對物理系統之無秩序或亂度的量度。在1948年，學者Shannon則提出了資訊熵[39]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[39]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。

但若大於，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)[22]，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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[22]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

結構學習是為了將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[11]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。

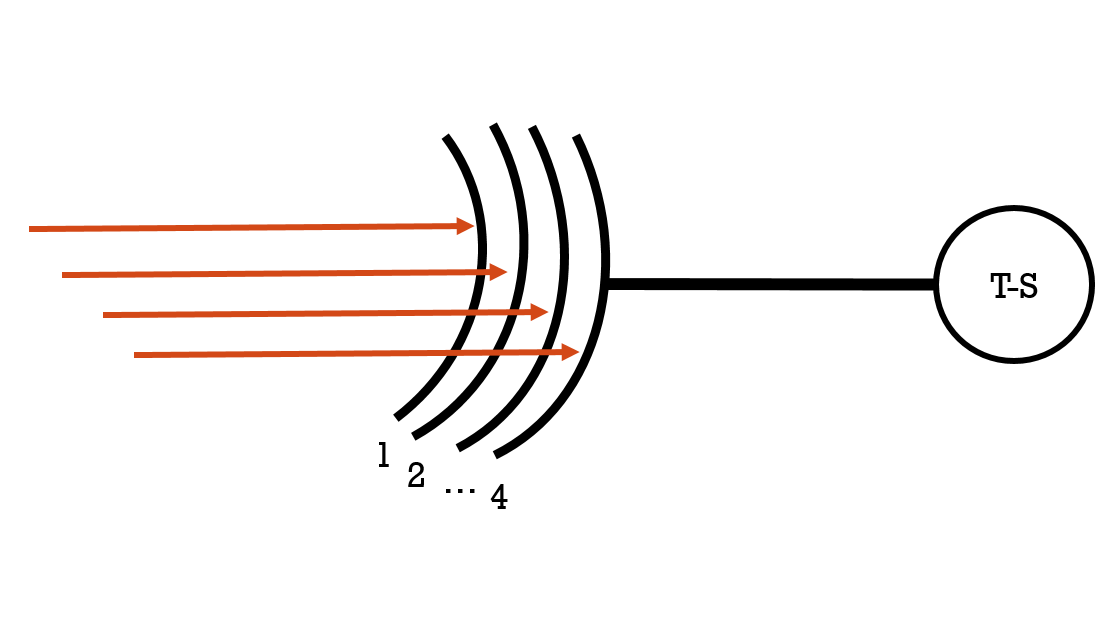
Step 3:查看每個區塊，若，則將累加。設定上下界，標記為和，透過上下界找出，表示最後選取的區塊數目。本研究中所有實驗皆設定為15，皆設定為4。若介於上下界間，則將設定成;若小於下界，則將設定成;若大於上界則將設定成。

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。

第3層的神經元數目將由訓練資料集合進行減數分群[11]決定，決定神經元數量後，將透過Fuzzy C–Mean分群方法[36]決定第3層神經元–箭靶神經元的相關數值，步驟如下:

Step 1:將輸入資料集合進行減數分群[11]，決定箭靶神經元個數為，即為第3層神經元的數目。

Step 2: 決定群數之後，透過Fuzzy C–Mean對目標集合進行分群，可以得到個群中心以及標準差。



1. 多層式箭靶。箭靶可一次接收多個輸入，在每個箭靶之後與T-S神經元相接。

Step 3: 由於本研究採用球式複數模糊集合，箭靶又會承接上一層的輸入，故每個箭靶都會有許多層，以接收輸入向量中的每個值，如圖(Fig. 4.)所示。其中，第層的第群中心標記為}，第層的第群標準差標記為{, }。用得到的群中心以及標準差製作箭靶，箭靶的製作以及詳細公式將會在下一小節討論。每個箭靶神經元後面連接著一個T–S神經元，T–S神經元為T–S function構成，T–S function公式如下。

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

其中，{}是第個T–S神經元的參數，是第個輸入。

在結構學習後，我們可以透過分群確認第1層的神經元個數，藉由區塊挑選得到個第2層神經元，最後透過分群可得到個第3層神經元和第4層神經元，藉此創建模型，模型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。

**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:** 箭靶是用以承接上一層的輸出，為向量的型態，本層的輸出亦是向量的型態，如下。

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

其中，為第個神經元射在第個箭靶第層的值，。本研究中使用的是球式複數模糊集，因此輸入會是複數型態，故輸出也要在複數單位圓盤中，因此箭靶需進行轉換，以確保箭靶層輸出也是複數型態，如下。

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

其中，。為第個箭靶轉換後的中心，為第個箭靶轉換後的靶寬，公式如下。

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

其中，使用作為輸入;使用作為輸入;為第個目標數據的平均;為第個目標數據的標準差。

**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**此層為輸出層，將上一層得到的個模型輸出加總，即為我們的模型輸出。

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

## 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。對於第1層模糊集的參數優化，我們使用知名的粒子群演算法(Particle swarm optimization, PSO)演算法[19]學習，其原理為模擬鳥群在尋找食物，每回合透過自身的最佳位置和全群最佳位置調節速度，特性為收斂快速，演算法公式如下。

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

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，粒子的位置代表模糊集中的參數，其中包含了每個維度的分群中心、標準差以及相位頻率函數、。

在本論文使用遞迴式最小平方演算法(Recursive least square estimation, RLSE)[17]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[17]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。

PSO–RLSE混合演算法的流程如下:

Step 1: 準備訓練資料及測試資料。

Step 2: 以PSO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。

Step 3: 用RLSE更新神經元的參數，RLSE算式中的和向量如下。

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

其中。因為多目標預測中，啟動強度為一向量，且為一矩陣，因此在原本的公式(62)中，利用單位矩陣取代原本的常數項1，改良後公式如下。

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

Step 4: 更新完所有參數後，計算出模型的輸出。

Step 5: 計算成本，更新PSO粒子自身的最佳位置和全群最佳位置。本研究成本函數(Cost function)使用均方根誤差(Root mean square error, RMSE)，定義如下。

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

其中，為模型第筆資料的誤差;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。

Step 6: 對所有PSO粒子重複Step 2~Step 5，直到PSO迭代結束。

## Parameter Learning

根據divide-and-conquer的概念，我們將使用不同的機器學習演算法，對各層的參數作優化，以便更容易找到最佳解。本實驗中我們使用了粒子群演算法、螞蟻演算法以及人工蜂群演算法個別當作前鑑部的參數學習演算法，後鑑部參數則使用了RLSE做學習，演算法內容如下:

1. 粒子群演算法 (PSO)

對於前鑑部的參數優化，我們使用知名的PSO演算法[18]學習，其原理類似鳥群在尋找食物，每回合透過自身的最佳位置和全群最佳位置調節速度，特性為收斂快速，演算法公式如下:





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。在本實驗中，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、。

1. 人工蜂群演算法 (ABCO)

此演算法總共有三種蜜蜂，包括工蜂、觀察蜂、偵查蜂，但與真實的蜜蜂找尋食物無直接相關，步驟如下:

Step 1: 隨機形成工蜂位置，並更新工蜂位置，公式如下:



其中，為第只工蜂在移動後的第個維度;為第只工蜂移動前的第個維度;為其他隨機蜜蜂的第個維度的值。

Step 2: 使用輪盤法選一個位置，輪盤機率公式如下:



其中，為機率;為適應值，本實驗將成本的倒數視為適應值;為蜜蜂的總數目

Step 3: 所有的觀察蜂在剛剛被選中的位置附近搜尋，公式如下:



其中，為第只工蜂在移動後的第個維度;為第只工蜂移動前的第個維度;為被選中蜜蜂的第個維度的值。

Step 4: 若該工蜂已經達到限制回合，都未更新則派出偵查蜂取代，偵查蜂位置產生公式如下:



其中，為第只工蜂的第個維度;為所有工蜂第個維度的最大值;為所有工蜂第個維度的最小值。

Step 5: 重複step 2~ step 4，直到反覆運算結束。

1. Recursive Least Square Estimator (RLSE)

在本實驗中RLSE[19]是用來更新後鑒部參數，一般來說LSE問題可以被指定成一個線性的模型，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:

a

其中:

b

c

d

e

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。要優化，可透過RLSE的等式[19]運算:

a

b

其中，是的第行，再開始RLSE演算法時，我們會設定為0，則設定為，為單位矩陣。我們cost function使用RMSE，定義如下:



其中，()’為複數的共軛運算;，其中為目標向量，為模型輸出向量。整個機器學習的的運作流程如下:

Step 1: 準備訓練資料。

Step 2: 用前鑒部演算法的粒子位置計算啟動強度。

Step 3: 用RLSE更新後鑒部參數，RLSE算式中的和向量如下:

a

b

c

其中。

Step 4: 更新完所有參數後，計算出模型的輸出。

Step 5: 計算RMSE。

Step 6: 對所有粒子重複Step2~Step5，直到回合結束。

## Methodology for financial application

為了評估模型是否賺錢，使用RMSE是不夠的，因為從RMSE中無法看出是否有賺錢，只能從中瞭解模型的配市率，而配市率高並不代表利潤高，因此本實驗將預測出來的值配合策略，進一步決定要買進或是賣出，進而算出模型所能賺的利潤，買進與賣出公式如下:

買進: a

賣出: b

其中，為模型的輸出，意即預測日的收盤價格;為日實際的開盤價格。若預測的收盤價高於實際開盤價，代表必須買進，等今日結束後才會賺錢;若預測的收盤價低於實際開盤價，代表模型預測今天會跌，所以要儘快賣出。

計算利潤的方式，則透過實際的收盤價與開盤價去做運算，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;代表第天的收盤價。

透過上述公式，我們可以獲得整個模型的所帶來的利潤值，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利潤值與其他參數。

# EXPERIMENTATION

## Example 1— Quadruple Time Series of Daily Dow Jones Industrial Average Index

在這個範例中，我們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The Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX)、Dow Jones Industrial Average Index (DJI) 、National Association of Securities Dealers Automated Quotation (NASDAQ)、Standard & Poor's 500 (S&P500)，每年的收盤價。為了和其他論文比較效能，我們對2001年的收盤價做預測，2001年的前十個月的資料當作訓練資料，剩餘的當作測試資料，訓練資料總共為181筆，測試資料為66筆。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當年的收盤價，複數部分則為DJI當年的收盤價，第二個目標實數部分為NASDAQ當年的收盤價，複數部分為S&P500當年的收盤價。

範例一模型設定

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Feature variable as model input | {} |
| Number of fuzzy sets of each input | 3、3、3、3 |
| Number of targets (real-valued) | 4 |
| Number of outputs (complex-valued) | 2 |
| Number of premises (before selection) | 81 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises (after selection) | 9 |
| Number of premise parameters | 48 |
| Type of consequences | Takagi-sugeno |
| Number of consequences | 9 |
| Number of consequence parameters | 45 |

結構學習部分，則將每個被挑選的特徵，用MATLAB提供的subclust()函式分群，係數為0.2。並透過前鑒部篩選機制，從原本的81個前鑒部篩選到剩下9個前鑒部。整體模型在結構學習後的參數，如表I所示。機器學習參數設定，如表II所示。

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 | Given by the subtractive clustering algorithm | |
| Initial velocity | 0 | |
| **ABCO** | | |
| Swarm size | 64 | |
| Iterations | 100 | |
| Limit | 20 | |
| **RLSE** | | |
|  |  | |
|  | 45x1 zero vector | |
|  | | **I** |
| **I** | | 45x45 identify matrix |

本次範例的結果除了本身三個演算法比較也與其他論文所提的方法做比較，像是ANFIS[21]，CNFS-ARIMA[6]，RBF network[22]和SVR[23][24]。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，結果如表III所示。模型的機器學習曲線，如圖(2)所示;目標與模型輸出的結果，如圖(3)所示。而模擬投資所賺的利潤，以及買賣次數如表V所示。



(a)



(b)

1. 機器學期曲線 (a) PSO-RLSE (b) ABCO-RLSE

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) [23] | - | - |  | 162.46 | 101.44 |
| ANFIS (two models, each with single output) [21] | - | - |  | 147.36 | 105.56 |
| ANFIS (one model with two outputs) [21] | - | - |  | 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) [6] | - | - |  | 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** |



(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 (實驗一)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Triers | 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 (實驗一)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Triers | PSO | | ABCO | |
| Profit | Sell/buy | Profit | Sell/buy |
| 1 | **2486** | 123/123 | 1808 | 127/119 |
| 2 | 1950 | 125/121 | 1706 | 125/121 |
| 3 | 2112 | 126/120 | 1539 | 126/120 |
| 4 | 1787 | 128/118 | 1650 | 128/118 |
| 5 | 2276 | 132/114 | 1912 | 124/122 |
| 6 | 2078 | 125/121 | 1519 | 125/121 |
| 7 | 1539 | 126/120 | 1637 | 125/121 |
| 8 | 2040 | 128/118 | **2298** | 125/121 |
| 9 | 2077 | 127/119 | 1954 | 127/119 |
| 10 | 2058 | 128/118 | 1709 | 126/120 |

## Example 2—Quadruple 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The Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX)、Hang Seng Index (HSI) 、Nikkei 225、Shanghai Stock Exchange Index (SSEI)，每年的收盤價。為了和其他論文比較效能，我們對2002年的收盤價做預測，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，訓練資料為205筆，測試資料為43筆。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當年的收盤價，複數部分則為HSI當年的收盤價，第二個目標實數部分為Nikkei 225當年的收盤價，複數部分為SSEI當年的收盤價。

結構學習部分，則將每個被挑選的特徵，用MATLAB提供的subclust()函式分群，係數皆為0.05。整體模型在結構學習後的參數，如表VI所示。機器學習參數設定，如表VII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 targets (real-valued) | 4 |
| Number of outputs (complex-valued) | 2 |
| Number of premises (before selection) | 625 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises (after selection) | 15 |
| Number of premise parameters | 48 |
| Type of consequences | Takagi-sugeno |
| Number of consequences | 15 |
| Number of consequence parameters | 75 |

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 | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 75x1 zero vector |
|  | **I** |
| **I** | 75x75 identify matrix |

本次範例的結果將與其他論文所提的方法做比較。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，結果如表VIII和表X所示。學習曲線如圖(4)所示。目標與模型輸出的結果，如圖(5)所示。而模擬投資所賺的利潤以及買賣次數，如表IX所示。

十次測試RMSE效能表 (實驗二)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Triers | 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 (實驗二)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Triers | PSO | | ABCO | |
| Profit | Sell/buy | Profit | Sell/buy |
| 1 | 1428 | 127/120 | 1544 | 128/119 |
| 2 | 903 | 126/121 | 1246 | 128/119 |
| 3 | 1428 | 124/123 | 1308 | 127/120 |
| 4 | 1467 | 127/120 | **2013** | 126/121 |
| 5 | **1727** | 127/120 | 1721 | 129/118 |
| 6 | 1092 | 128/119 | 1426 | 129/118 |
| 7 | 1699 | 126/121 | 1427 | 129/118 |
| 8 | 1718 | 129/118 | 1361 | 131/116 |
| 9 | 966 | 129/118 | 1999 | 128/119 |
| 10 | 1423 | 127/120 | 1701 | 130/117 |



(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)

多目標預測效能比較表(RMSE)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **RMSE** | | | | |
| Training phase | |  | Testing phase | |
| TAIEX | HSI |  | TAIEX | HSI |
| Chen [31] | - | - |  | 75 | 187 |
| Yu [32] | - | - |  | 101 | 170 |
| AR(1) [33] | - | - |  | 66 | 105 |
| SVR [23] | - | - |  | 66 | 107 |
| ANFIS [21] | - | - |  | 65 | 106 |
| ANFIS (EMD) [34] | - | - |  | 52 | 97 |
| PSO-RLSE (proposed) | **99.16** | **98.19** |  | **89.77** | **85.72** |
| ABCO-RLSE (proposed) | **98.77** | **97.12** |  | **85.25** | **80.62** |

## Example 3—Quadruple 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，這次預測的目標為APPLE Computer Inc.、International Business Machines Corporation (IBM)、Dell Inc.、Microsoft Inc.，四個股票的收盤價，期間為2003年2月10號至2005年1月21號，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，共92筆。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，我們這個範例中使用的第一個複數目標實數部分為IBM的收盤價，複數部分則為APPLE的收盤價，第二個目標實數部分為Dell的收盤價，複數部分為Microsoft的收盤價。

結構學習部分，則將每個被挑選的特徵，用MATLAB提供的subclust()函式分群，係數皆為0.15。並透過前鑒部篩選機制，從原本的81個前鑒部篩選到剩下8個前鑒部。整體模型在結構學習後的參數，如表XI所示。機器學習參數設定，如表XII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 targets (real-valued) | 4 |
| Number of outputs (complex-valued) | 2 |
| Number of premises (before selection) | 81 |
| Type of premises | Sphere complex fuzzy set |
| Number of premises (after selection) | 8 |
| Number of premise parameters | 44 |
| Type of consequences | Takagi-sugeno |
| Number of consequences | 8 |
| Number of consequence parameters | 40 |

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 | Given by the subtractive clustering algorithm |
| Initial velocity | 0 |
| **ABCO** | |
| Swarm size | 64 |
| Iterations | 100 |
| Limit | 20 |
| **RLSE** | |
|  |  |
|  | 40x1 zero vector |
|  | **I** |
| **I** | 40x40 identify matrix |

本次範例的結果將與其他論文所提的方法做比較，像是HiMMI[35]、ANN-GA-HMM-Interpolation[35]、ANN-GA-HMM-WA[35]、ARIMA[5]。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。模型的機器學習曲線，如圖(5)所示;目標與模型輸出的結果，如圖(6)所示。而模擬投資所賺的利潤以及買賣次數，如表XIV所示。



(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)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Triers | 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)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Triers | PSO | | ABCO | |
| Profit | Sell/buy | Profit | Sell/buy |
| 1 | **0.762** | 265/226 | 0.700 | 246/245 |
| 2 | 0.596 | 266/225 | 0.570 | 255/236 |
| 3 | 0.671 | 264/227 | **1.041** | 235/256 |
| 4 | 0.67 | 264/227 | 0.504 | 263/228 |
| 5 | 0.396 | 270/221 | 0.617 | 257/234 |
| 6 | 0.369 | 267/224 | 0.147 | 257/234 |
| 7 | 0.760 | 265/226 | 0.354 | 252/239 |
| 8 | 0.281 | 269/222 | 0.190 | 257/234 |
| 9 | 0.477 | 268/223 | 0.410 | 255/236 |
| 10 | 0.494 | 267/224 | 0.722 | 252/239 |



(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 [35] | - | - |  | 2.8373 | 1.2186 |
| ANN-GA-HMM-Interpolation [35] | - | - |  | 2.1649 | 1.0555 |
| ANN-GA-HMM-WA [35] | - | - |  | 1.9247 | 0.8487 |
| Bayesian ANN [36] | - | - |  | 1.9688 | 0.7441 |
| ARIMA [5] | - | - |  | 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** |

# CONCLUSION

透過三個實驗可以發現，本論文提出的模型SCFNS確實有多目標預測的能力。而透過多目標特徵挑選，可以根據不同的資料去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。結構化學習的部分，可以將輸入的資料自動地根據資料做調整，在面臨不同的資料可以自己生成不同的結構。從實驗中可以發現透過一次對四個目標做時間序列的預測，各個目標的效果不亞於其他論文所提出的方法，甚至更好。證明了，不同資料都能在此模型中，被有效的預測。也代表此研究中的三種演算法配合RLSE的混合方法，有著一定的水準。目前可以發現模型對多目標的時間序列預測是可以運作的。

機器學習的部分可能會受限於演算法本身的特性，像是粒子群演算法(PSO)收斂快速、較容易掉入區域最佳解，PSO演算法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。而螞蟻演算法(ACO)則是每回和每只螞蟻都有可能會在不同的地方搜尋，且具有淘汰機制，會將較差的淘汰掉，整體來說模型較穩定。透過三個實驗可以發現，ABCO十次表現中的最佳表現是優於ACO以及PSO的，但可以在第二個實驗中發現，ABCO的穩定度較不足，原因在於可能在初始值就已經有很大的誤差，在後續更新搜尋位置時，無法順利的進入最佳解位置。在未來，可以透過不同的機器學習演算法結合SCFNS，像是Random Search[25]等，以克服目前演算法所遇到的問題，亦可以透過CPSO[26]等演算法，使用divide-conquer概念去尋找最佳解。

本篇論文所使用的IF-THEN規則類神經網路，前鑒部(IF-part)和後鑒部(THEN-part)各數相同，未來可使用不同個數，如同神經元之間的連接，增加模型的彈性，使其可以增添更多的隱藏層。最後，在模擬投資方面，我們可以從實驗中瞭解到，每個演算法皆能賺取利潤，且我們發現兩件有趣的事，第一，我們可以觀察到買與賣的次數和利潤毫無相關，第二，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。透過虛擬投資，可以得知投資任何專案皆有變數，因此需謹慎的操作每次的投資，未來可透過不同的投資組合，降低失利的風險。

##### ACKNOWLEDGMENT

##### REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] A. J. Patton, "A review of copula models for economic time series," *Journal of Multivariate Analysis,* vol. 110, pp. 4-18, 2012.

[14] D. Perez-Liebana *et al.*, "The 2014 general video game playing competition," *IEEE Transactions on Computational Intelligence and AI in Games,* vol. 8, no. 3, pp. 229-243, 2016.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.