# 行政院國家科學委員會專題研究計畫 成果報告

## 結合 Bloom Filter 與 CMAC 作為字串辨識碼多屬性高效能查詢模式
## 研究成果報告(精簡版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 97-2221-E-216-030-
執 行 期 間 ：97 年 08 月 01 日至 98 年 07 月 31 日
執 行 單 位 ：中華大學工業工程與系統管理學系

計 畫 主 持 人：馬恆
共 同 主 持 人：張弘鑫
計畫參與人員：博士班研究生-兼任助理人員：鄭弘裕

處 理 方 式 ：本計畫可公開查詢

中 華 民 國 98 年 11 月 06 日

行政院國家科學委員會補助專題研究計畫 ■ 成 果 報 告
□ 期中進度報告

# 結合Bloom Filter與CMAC作為字串辨識碼多屬性高效能查詢模式

計畫類別：■ 個別型計畫　□ 整合型計畫
計畫編號：NSC 97－2221－E－216－030－
執行期間：　97 年 08 月 01 日至　98 年 07 月 31 日

計畫主持人：馬恆
共同主持人：張弘鑫
計畫參與人員：　鄭弘裕、林子淵、蔡翠芳

成果報告類型(依經費核定清單規定繳交)：■精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：中華大學工業工程與系統管理學系

中　華　民　國　98 年　10 月　01 日

## 摘要

本研究重點著重於具多重屬性之大量資料高效能查詢演算法，透過演算法實現壓縮資料、快速查詢等功能。然而隨著資訊電子化普及導致數位資料的查詢需求快速成長，在過去面對此問題的解決方法大多以演算法搜尋資料庫，然而無論是分散式或集中式系統架構對於龐大的查詢需求也會造成資源負擔。高效能的查詢模式如Bloom filter、CMAC，主要原理是將原始資料轉換為低維度型態以利提高運算效率。然而此類型查詢模式原始的設計重點在於簡單的二元查詢，如存在屬性。對於多重屬性查詢必須設計出更複雜的結構，效能也相對地降低。本研究改良傳統CMAC神經網路，開發多屬性編碼及反饋修正訓練演算法，實現多屬性查詢架構。本研究將以台灣自用車牌做為訓練樣本，實現多屬性編碼、資料壓縮及快速搜尋等功能。以不同的記憶體大小評估資料壓縮效率及誤判率。本研究所發展之演算法不僅具有壓縮資料的效果、多屬性快速查詢的效能，更能感受到誤判率的優秀表現。



**關鍵詞**：多屬性分類,小腦模型控制器, 布隆篩檢



## Abstract

This research focuses on developing a high-speed query algorithm for multi-attribute large-volume data, in which data compression and fast querying can be simultaneously achieved. In the past decade, this type of problem is usually resolved by storing the data in a database, and querying is carried out through database-embedded methods. As the data volume is growing in a tenfold fashion, these methods are encountering difficulties in data manipulation as far as efficiency is concerned. A number of approaches, e.g., the Bloom Filters, have been proposed for querying purposes in large-volume data using a compact structure requiring less memory space. These approaches often limit themselves to single-attribute querying, and the multi counterpart can only be achieved by layered implementations where errors, e.g., the false positives, are prone to increase. We propose an improved CMAC mechanism seeking to achieve fast multi-attribute querying with less memory space and errors. Experimental results show favorable performances for our proposed mechanism.



**Keywords**: Multi-attribute, CMAC, Bloom Filter

## 一、前言

生活中的所有事物因資訊科技影響，將書面化資料全面轉換成電子媒體，透過資料庫技術進行儲存，並且提供查詢、搜尋與運算等服務。受到高度資訊化影響，資料變得更加龐大、複雜與多元化。對於上述的議題與需求，本研究以 CMAC (cerebella model articulation controller) 演算法[1][2]為基礎，應用於解決快速識別與資料搜尋方面的需求，具有快速識別、運算速度快、誤判率低及容易實作的特性。傳統的 CMAC 演算法屬於單屬性輸出，無法提供多屬性分類特性，只能識別是否存在於集合內，儘管效能表現相當顯著但對於應用領域卻有所限制。CMAC 屬於一種模仿人類小腦儲存訊息的類神經網路架構。此網路架構將量化後的輸入狀態資訊分別儲存於索引陣列記憶單元，將記憶單元的實數加總產生輸出，藉由反覆學習的機制完成神經網路學習階段。CMAC 不僅是一種查表的技術，同時也具有快速收斂、計算簡單、函數近似和類比化能力等優點。因此常應用於機器人的控制[3]、階層影像編碼[8]、減少記憶體與硬體實現[9]方面的研究。

## 二、研究目的

對於 CMAC 與 Bloom Filter 的架構有許多相似之處，包含了陣列記憶體儲存空間、查表類型、簡單的計算方程式、收斂速度快等多項特性。藉由單純網路結構大幅降低了其運作複雜度提高實作可行性。傳統 CMAC 與 Bloom Filter 也都採用雜湊函數作為資料編碼定址，透過多個 Hash 函數將資料編碼定址，利用定址偏移及微量差異的特性，實現資料壓縮儲存的目的。此資料壓縮儲存方式，對於記憶體空間的規劃也顯得特別重視，由於採用一維的資料結構模式，僅應用於單一屬性識別與查表無法提供多屬性分類應用領域。

然而提到類神經網路之演算法，對於演算效率最重要的評估指標就是 FAR(False Accept Rate)與 FRR(False Reject Rate)。對於傳統的 Bloom Filter 與 CMAC 演算法而言，可藉由記憶體空間、雜湊函數以及相關參數有效控制誤判問題，而誤判率與記憶體空間有深切的關係。然而 CMAC 與 Bloom Filter 之架構與功能具有高度同質性，最大的差異在於資料儲存的方式與反饋機制。本研究改良 CMAC 所儲存的權重陣列為實數資料型態，透過輸出值與目標值誤差作為反饋的依據，進而修正權重陣列數值達到訓練收斂條件。透過實數計算可以控制誤判率、調整訓練學習率等好處。兩者雖同樣為查表架構，CMAC 卻擁有更彈性的應用空間。

許多研究開始發展並改良 Bloom Filter 與 CMAC 演算法，希望能保持高效能之演算特性，提高記憶體空間運用效率，更能提供分類與多屬性的特性，以擴大應用領域範圍。而發展方向主要朝三大方向著手：

(一) 有效運用記憶體空間—對於 CMAC 與 Bloom Filter 而言，記憶體空間的多寡與判別的準確度有密切關係。因此如何善用記憶體空間及擁有高辨識能力將更顯得重要。對於使用 Hash 進行編碼定址的應用而言，若記憶體空間不足容易導致編碼定址的重複性過高，因此 False Negatives 機率會大幅提高。

(二) 發展多屬性分類架構—由於 CMAC 與 Bloom Filter 傳統設計為單一階層(一維陣列)儲存單一屬性架構。本研究提出 CMAC 作為多屬性分類模式，改良式 CMAC 依然保留一維陣列架構，透過實數、反饋修正等特性，實現多屬性分類應用，甚至可延伸為多層級架構模式，

以提高多屬性分類之複雜度。

(三) 降低誤判率—誤判率是關鍵的績效評估指標，無論是 Bloom Filter 或 CMAC 都需透過科學實驗過程獲得最佳參數組合，考量收斂時間、記憶體儲存空間、誤判率等多方目標水準之下方能發揮最佳效益。

## 三、研究方法

本研究結合 CMAC 與 Bloom Filter 特性透過實驗評估及分析，發展適用於文字字串類型的多屬性分類演算法，除了具有查表演算法的優點，更提供多屬性分類功能。本研究提出多屬性分類 CMAC 演算法(Multi-Attribute Classification-CMAC，以下簡稱 MAC-CMAC)如圖 1。以傳統的 CMAC 單層 Association 單輸出架構為基礎，將單屬性輸出轉變成多屬性輸出模式。為了提升編碼效率將多個雜湊單元編碼架構改良成序列式編碼以簡化編碼過程。而輸出門檻值由固定門檻誤差，改良成動態樣本門檻視窗，藉由訓練過程中動態取得多目標屬性專屬的門檻視窗。

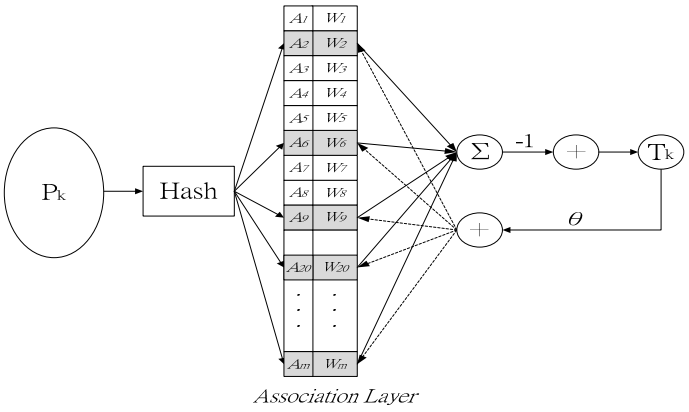


圖 1. MAC-CMAC 網路架構

本研究發展之演算法可應用於字串型態、連續及非連續數值的類型，因此樣本可包含符號、字元等串列形式組合而成，例如:URL、車牌號碼等。編碼方式使用樣本字元所對應之 ASCII Code 進行組合式編碼。本研究所使用之樣本為車牌號碼，車牌由 26 個英文字母、10 個阿拉伯數字以及特殊符號所組合的字串，為了提高其組合複雜度，採用組合式編碼，使其降低樣本編碼重複率。

CMAC 演算法關鍵階段就屬雜湊定址階段，傳統 CMAC 採用多組雜湊演算函數作為定址組合，MAC-CMAC 採用單一單元雜湊定址，改善傳統固定長度、固定雜湊定址數量等限制，對於動態長度的樣本能提供更適當的解決方案。單一單元雜湊定址就是使用單一雜湊函數對樣本進行序列式定址編碼，產生之定址編碼組合，作為權重記憶體定址索引指標，也就是該樣本所對應之索引陣列定址的集合。本研究利用樣本的特徵值作為編碼種子，包含 ASCII 特徵值、權重記憶體大小$(m)$、雜湊因子$(\gamma)$以及串列編碼累加值 $(\partial)$ 如式 1。本研究所發展的雜湊函數適用於動態的樣本大小與連續字元串列組合字串等類型。依據樣本的長度$(l(p))$產生雜湊編碼數量，其定址輸出結構如同傳統 CMAC 使用多個雜湊函數產生多組定址。

本研究主要是發展多屬性分類，因此每個樣本需設定單一屬性分類，並且以實數方式表示。本研究以目標值$(T(p))$作為樣本的分類屬性。神經網路訓練階段在每個分類屬性集合$(S_T)$，以

輸出權重向量的最大值與最小值作為門檻視窗，在批次訓練階段必須記錄該分類屬性之門檻視窗範圍，除了作為回想階段門檻值判斷依據，亦作為訓練停止條件。因此將取得各分類屬性的動態門檻視窗(Dynamic Thread-hold Window)。由於權重向量為實數運算，因此可能具有正負值。為了提高反饋補償效率，設計每個分類屬性具有正負值門檻視窗如式 2，這種作法可加快訓練收斂速度並且提升多屬性函數逼近能力。

訓練階段由雜湊函數計算產生的索引陣列定址$\big(h(p)\big)$如式 3，並累加索引陣列定址集合的實數而產生權重輸出向量$(\omega(p))$ 如式 4，將累加定址集合的實數與目標值$(T_k(p))$進行加法運算，取得誤差值作為反饋補償的誤差總向量$(\varepsilon)$ 如式 5。將誤差總向量平均累加至索引陣列定址集合所對應之權重儲存陣列，透過學習率$(\theta)$穩定且漸進的調整反饋補償速度，漸進修正權重網路如式 6，透過此訓練階段達成神經網路學習穩態。

$$\partial = \left(\sum_{j=0}^{l(p)} Ascii(p_j)\right) \tag{1}$$

$$T_k(p) = \begin{cases} +T_k & \omega(p)>0 \\ -T_k & otherwise \end{cases} \tag{2}$$

$$h\big(p(i)\big) = \sum_{j=0}^{i}\Big(\big(Ascii\big(p(j)\big)\times\gamma\times\partial\big) mod\ m\Big) \qquad i = 1\sim l(p) \tag{3}$$

$$\omega(p) = \left(\sum_{j=0}^{l(p)} Wei\left(h\big(p(j)\big)\right)\right) \tag{4}$$

$$\varepsilon = -\omega(p) + T_k(p) \tag{5}$$

$$Wei_j = Wei_j + \left(\theta \times {}^{\varepsilon}/_{l(p)}\right) \tag{6}$$

訓練階段(如圖 2)過程中要判斷神經網路訓練完成與否，取決於訓練停止條件，因此設計適合的訓練停止條件是神經網路判斷是否具有穩定性關鍵的要素。過於寬鬆停止條件容易造成誤判率的發生；過於嚴格的停止條件不僅會造成訓練時間的延長，更會造成過度訓練，而提高誤判率。本研究以誤差總向量的穩定程度作為停止條件，若每世代的誤差總向量差距低於收斂誤差參數$(err)$表示以達到穩定狀態，亦表示訓練結束，反之則繼續進行下批次訓練階段。

Algorithm Training()

1: $Wei \leftarrow CreateCMAC(m)$
2: $P \leftarrow Load\_Pattern()$
3: $Threadhold = CreateThreadhold()$
4: $ThRecord = CreateThRecord()$
5: ***while***
6: ***for*** $i = 1$ ***to*** $P.Count$ ***do***
7: $Addr \leftarrow Hash(P_i)$
8: $\gamma \leftarrow Sum(Addr)$
9: $\varepsilon \leftarrow -\gamma + P_i.Target$
10: $ThRecord \leftarrow Record(\gamma, \varepsilon, P_i, ThRecord)$
11: $Wei[Addr] \leftarrow Wei[Addr] + (\theta \times \frac{\varepsilon}{P_i.length})$
12: ***if*** $Steady(ThRecord, ThRecord', err)$ ***then Exit_while***
$ThRecord' \leftarrow ThRecord$

圖 2. 訓練階段

為了達到本研究的多目標屬性識別，MAC-CMAC 神經網路透過回想階段(如圖 3)來判斷樣本的分類屬性結果，藉此達到多屬性判別之目的。回想階段架構與訓練架構類似，主要差別於回想階段是透過訓練階段產生的動態門檻視窗來判斷其樣本所歸屬之分類屬性。由於回想演

算法結構單純與計算簡易等特性，大幅降低實作的複雜度。首先載入 MAC-CMAC 神經網路，包含權重節點向量、雜湊函數、分類屬性之動態門檻視窗等資料。回想階段與訓練階段相同需進行樣本編碼、雜湊定址，累加定址之權重節點產生輸出值(β)如式 7，並以訓練階段的動態門檻視窗($T_k$)作為判別分類屬性之指標，判斷若ε落在某一個動態門檻視窗內即表示為該分類屬性，若滿足多個門檻視窗或無任何吻合，將回應無法識別。

| Algorithm Recall(p) | |
|---|---|
| 1: | $T_k = \{T_1, T_2, \dots, T_j\}$ |
| 2: | $Wei \leftarrow LoadCMAC()$ |
| 3: | $\beta \leftarrow Sum(Wei[Hash(p)])$ |
| 4: | $\boldsymbol{for}\ i\ = 0\ to\ T_k.count\ \boldsymbol{do}$ |
| 5: | $\boldsymbol{if}\ (T_i.lowerbound \le \beta \le T_i.upbound\ )$ |
| 6: | $\boldsymbol{Return}\ i$ |

圖 3. 回想階段

$$\beta = \left(\sum_{j=0}^{l(p)} Wei\left(h(p(j))\right)\right) \tag{7}$$

## 四、實驗結果

研究實驗內容將針對多目標屬性數量、儲存空間及樣本數量來進行實驗分析，並以 FAR(False Accept Rate)、FRR(False Reject Rate)作為衡量效能的指標，而誤判率影響的因子包含樣本數量、記憶體大小等因素。影響記憶體大小的因素包含：權重網路陣列大小、權重網路陣列儲存資料型態(解析度)。在相同的記憶體大小限制下，當權重網路陣列越大，解析度相對的降低。以車牌號碼作為實驗樣本，車牌號碼為六位英數字及分隔符號所組成，並隨機將車牌歸類成不同的分類屬性，做為樣本之目標值。

1. 記憶體空間與解析度對於誤判率的影響

本階段實驗以相同的樣本數量進行實驗，以不同規模的記憶體大小來分析其誤判率效能。記憶體大小將由解析度與權重陣列大小所構成，而解析度與權重陣列將成反比。以權重節點所儲存的實數範圍稱為解析度(γ)，本研究將解析度分成 16-bit, 24-bit, 32-bit 三種層級，實數由正負號、整數及小數所構成。實驗階段所使用之解析度所對應之實數範圍及收斂誤差參數(如表 1)。

表 1.解析度的儲存實數範圍對照表

| 解析度 | 實數範圍 | 收斂誤差參數($err$) |
|---|---|---|
| 16-bit | +/- 1.9999 | 0.001 |
| 24-bit | +/- 7.999999 | 0.00001 |
| 32-bit | +/- 15.99999999 | 0.00000001 |

由於解析度與權重陣列大小息息相關，若採用高解析度(32-bit)將減少權重陣列數量，導致陣列激發重複性高，造成收斂不易；若著重於權重陣列數量，會導致解析度太低造成門檻視窗過大以致誤判率偏高。因此如何權衡這些因素，讓演算架構達到最佳效率將是本階段實驗的主要目的。本階段實驗採用訓練樣本及測試樣本各 100,000 組，隨機將樣本目標屬性分成 5 種類型(1.0~5.0)，以不同的記憶體空間分別測試 16-bit、24-bit、32-bit 三種不同等級解析度的誤判率表現。由此階段實驗(如表 2)可得知誤判率與解析度關係密切，然而解析

度越高所需的訓練時間相對較長。因此適用於離線(Off-line)或非即時性(Non-real-time)的環境下。

表 2. 記憶體大小與解析度實驗結果

| 記憶體(KB) / 解析度 | 500 | 550 | 600 | 650 | 700 | 750 | 800 | 850 | 900 | 950 |
|---|---|---|---|---|---|---|---|---|---|---|
| 16-bit | 11998 | 9615 | 10052 | 9363 | 9454 | 7637 | 9999 | 9459 | 6001 | 8391 |
| 24-bit | 113 | 139 | 92 | 85 | 82 | 85 | 41 | 94 | 65 | 37 |
| 32-bit | 2 | 2 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

※結果為誤判次數

2. 不同分類屬性數量對於誤判率的影響

本研究所發展出 MAC-CMAC 多目標屬性演算法中，目標屬性的數量也將影響其誤判率的表現，本階段實驗採用訓練樣本及測試樣本各 100,000 組，隨機將樣本的分類屬性並分成 5、10、15 類型(目標值：1.0~15.0)的分類屬性，以 16-bit、24-bit、32-bit 三種解析度為實驗環境。藉由下列實驗可得知屬性數量與解析度對於誤判率的表現。(如表 3、表 4、表 5 所示)

表 3. 16-bit 解析度實驗結果

| 分類屬性 / 記憶體(KB) | 5 | | 10 | | 15 | |
|---|---|---|---|---|---|---|
| | FAR | FRR | FAR | FRR | FAR | FRR |
| 400 | 147100 | 0 | 7070 | 83097 | 107 | 97142 |
| 500 | 8893 | 0 | 26020 | 63365 | 237 | 97056 |
| 600 | 6230 | 0 | 73509 | 12188 | 719 | 96714 |
| 700 | 5171 | 0 | 64311 | 7 | 1752 | 95809 |
| 800 | 5090 | 1 | 52274 | 8808 | 2613 | 95162 |
| 900 | 4627 | 1 | 51912 | 14129 | 2021 | 95044 |
| 1,000 | 4340 | 0 | 67754 | 2376 | 1270 | 95703 |

※結果為誤判次數

表 4. 24-bit 解析度實驗結果

| 分類屬性 / 記憶體(KB) | 5 | | 10 | | 15 | |
|---|---|---|---|---|---|---|
| | FAR | FRR | FAR | FRR | FAR | FRR |
| 400 | 349 | 0 | 279 | 0 | 533 | 0 |
| 500 | 129 | 0 | 107 | 0 | 108 | 0 |
| 600 | 108 | 0 | 60 | 0 | 81 | 0 |
| 700 | 96 | 0 | 50 | 0 | 49 | 0 |
| 800 | 56 | 0 | 60 | 0 | 35 | 0 |
| 900 | 61 | 0 | 36 | 0 | 29 | 0 |
| 1,000 | 39 | 0 | 41 | 0 | 33 | 0 |

※結果為誤判次數

表 5. 32-bit 解析度實驗結果

| 分類屬性 / 記憶體(KB) | 5 | | 10 | | 15 | |
|---|---|---|---|---|---|---|
| | FAR | FRR | FAR | FRR | FAR | FRR |
| 400 | 7256 | 16576 | 897 | 34038 | 2015 | 44327 |
| 500 | 8 | 0 | 3 | 0 | 3 | 0 |
| 600 | 0 | 0 | 1 | 0 | 1 | 0 |
| 700 | 1 | 0 | 1 | 0 | 1 | 0 |
| 800 | 1 | 0 | 2 | 0 | 0 | 0 |
| 900 | 2 | 0 | 2 | 0 | 0 | 0 |
| 1,000 | 3 | 0 | 0 | 0 | 0 | 0 |

※結果為誤判次數

由實驗結果得知，當相同解析度環境下，目標屬性將會影響門檻視窗的收斂效果，當目標屬性越多越容易導致不同的目標屬性門檻視窗重疊，而產生多個目標屬性輸出，並且無法分辨樣本的正確目標屬性。然而實驗結果顯示高解析度的環境下可有效降低誤判率，但也大幅減少權重陣列的數量，產生樣本雜湊編碼重複機會高，造成不同目標屬性門檻視窗重疊，發生多目標屬性輸出。因此由本階段實驗得知，本研究所發展之多目標屬性演算架構對於目標屬性數量的多寡與誤判率並無顯著影響，但目標屬性過多將導致不易收斂，而影響收斂時間。

本研究所採用之樣本目標$T_k$為實數，隨著目標屬性而遞增。而本演算架構之目標輸出值$\omega(p)$由樣本雜湊編碼所激發的權重陣列累加產生，因此樣本目標值的訂定與權重陣列的解析度關係密切。本研究建議目標值$(T_k)$的設計必須依據解析度的範圍及樣本長度$(l(p))$而訂(如式 8)，而目標值集合建議以等距方式設計，避免造成權重值失衡影響收斂的效果。

$$\sum^{l(p)} \gamma_{min} < T_k < \sum^{l(p)} \gamma_{max} \tag{8}$$

3. MAC-CMAC 與 Bloom Filter 對於多目標屬性效能比較

本階段將以 Bloom Filter 演算法作為對照實驗，傳統 Bloom Filter 與 CMAC 相同，為單屬性輸出。因此為了達到多屬性輸出。Guo[6]提出降低 BF 所浪費的空間與效能，使用多層級 BF，依據樣本數量動態增減其 BF 的層級數量，使其達到動態改變目的，然而雖然為多層級架構但仍為單屬性輸出。本研究將傳統的 Bloom Filter 以多層次架構實現，將每個分類屬性樣本儲存至個別 BF 層級，藉此達到多屬性輸出之目的。然而此作法雖可達到多目標屬性輸出，會容易造成層級誤判的現象，產生 FAR 的誤判現象。

本研究將設計多層級 Bloom Filter 網路架構(如圖 4)實現多屬性分類，由於傳統 Bloom Filter 是採用多個雜湊函數作為輸入定址數量，為了實現字串長度動態定址編碼，因此將 Bloom Filter 改成單一雜湊函數序列式定址編碼，並且與 MAC-CMAC 使用相同之雜湊函數模型，避免雜湊定址效果的差異而影響實驗結果。

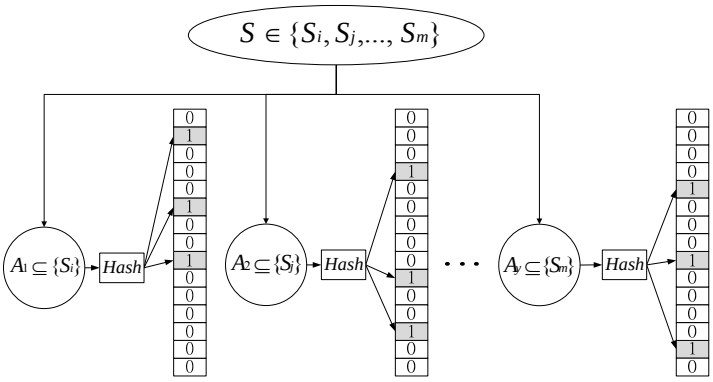


圖 4. 多層級 Bloom Filter 網路架構

本階段以多層級 Bloom Filter 及 MAC-CMAC 演算法進行實驗比較，雖然 CMAC 與 Bloom Filter 具有相同的雜湊編碼、陣列對應類型等特徵，但最大的差異在於 CMAC 沒有 False Positive Probability(Type I error)的錯誤型態，也就是會產生多組目標值輸出現象。因此本階段實驗不僅使用測試樣本進行誤判率測試，也使用訓練樣本來進行誤判率比對。本階段實驗環境如下：

- 樣本類型：7 位英數及符號所組成的車牌號碼。
- 訓練樣本：1,000,000 組
- 測試樣本：100,000 組
- 目標屬性：5 組〔目標值：1.0~5.0〕

表 6. MAC-CMAC 與 Bloom Filter 比較實驗結果

| | CMAC | | | Bloom Filter | | |
|---|---|---|---|---|---|---|
| | FAR | | | 訓練樣本 | 測試樣本 | |
| Memory(KB) | 16-bit | 24-bit | 32-bit | FRR | FRR | FAR |
| 12600 | 36886 | 1071 | 0 | 880 | 2 | 131 |
| 11900 | 36183 | 207 | 0 | 1043 | 0 | 129 |
| 11200 | 24157 | 229 | 3 | 1231 | 3 | 143 |
| 10500 | 27288 | 214 | 0 | 1284 | 1 | 166 |
| 9800 | 33701 | 269 | 2 | 1473 | 5 | 177 |
| 9100 | 42189 | 251 | 1 | 1783 | 2 | 235 |
| 8400 | 37512 | 237 | 4 | 1995 | 3 | 237 |
| 7700 | 42681 | 395 | 2 | 2426 | 6 | 271 |
| 7000 | 34719 | 334 | 1 | 2986 | 5 | 345 |
| 6300 | 29130 | 237 | 14 | 3594 | 9 | 451 |

※結果為誤判次數

本階段使用數量龐大的樣本來進行實驗測試，由實驗結果得知幾項結論。首先，儲存空間的解析度與誤判率具有密切的相關性。MAC-CMAC 演算法對於樣本資料量與誤判率表現並無顯著影響。對於樣本數量多寡具有高抗雜訊能力以及較穩定的識別效能。其次，與 Bloom Filter 相較之下，MAC-CMAC 不具有 Type I errors，因此適合應用於已存在的資料搜尋。然

而 Bloom Filter 應用於大量資料且多目標屬性環境下，需要更大的記憶體儲存空間。若儲存空間不足將容易導致 FRR 現象。

## 五、結論

本研究提出一種應用於字串快速搜尋與屬性識別的改良式 MAC-CMAC 演算法，不僅可應用於大量資料字串的識別，更可應用於分散式資料庫查詢服務中，透過分散式快速搜尋識別進行前處理，過濾不存在資料庫內的資訊，以減輕後端資料庫服務的負擔。本研究將由先前 CMAC 單屬性研究[13]延伸發展，為了解決多屬性輸出需求，並保留快速搜尋的優點，透過多屬性的架構提供更多元應用。

本研究透過三項實驗階段得知，記憶體空間、儲存陣列解析度兩者參數對於誤判率有密切的關係，高解析度有助於控制誤判率的現象，但需要花費更長的收斂時間，因此適合於離線模式的應用環境。本研究以搜尋應用最普及的 Bloom Filter 演算法為基礎，改良為多屬性輸出架構作為實驗對照組。實驗結果發現 MAC-CMAC 演算法對於龐大資料量的應用較為適合，與 Bloom Filter 演算法最大的優勢在於不具有 FRR(Type I error)的問題，因此對於已存在的資料搜尋應用更為適合。實驗結果得知在有限的記憶體空間環境下 MAC-CMAC 的誤判率相對的穩定度。在未來的研究方向將朝向多元、長度不一致的字串樣本進行多屬性識別的研究，將 MAC-CMAC 演算法應用於更寬廣的領域。

## 六、參考文獻


[1] Albus, J. S. (1975). “A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC).” ASME J. Dynamic Systems, Measurement, Control, pp.220-227.

[2] Albus, J. S. (1975). “Data Storage in the Cerebellar Model Articulation Controller (CMAC).” ASME J. Dynamic Systems, Measurement, Control, pp.228-233.

[3] Albus, J. S. (1979). “A Model of the Brain for Robot Control, Part2: A Neurological Model.” BYTE, vol.4, no.7, pp.54-95.

[4] Cohen, S., and Matias, Y. (2003). “Spectral bloom filters.” In Proc. 22nd ACM International Conference on Management of Data (SIGMOD), pages 241.

[5] Fan, L., Cao, P., Almeida, J., and Broder, A. (2000). “Summary cache: A scalable wide-area web cache sharing protocol.” IEEE/ACM Trans. Networking, 8(3):281rans.

[6] Guo, D., Wu, J., Chen, H., and Luo, X. (2009). “The Dynamic Bloom Filters.” IEEE Transaction on Knowledge and Data Engineering.

[7] Hao, F., Kodialam, M., and Lakshman, T. V. (2007). “Building high accuracy bloom filters using partitioned hashing.” In Proc. SIGMETRICS/ Performance, pages 277TRICS.

[8] Iiguni, Y. (1996). “Hierarchical Image Coding via Cerebellar Model Arithmetic Computers.” IEEE Transaction on Image Processing, vol.5, no.10, pp.1393-1401.

[9] Ker, J. S., Kuo, Y. H., Wen, R. C., and Liu, B. D. (1997). “Hardware Implementation of CMAC Neural Network with Reduced Storage Requirement.” IEEE Transaction on Neural Networks, vol.8, no.6, pp.1545-1556.

[10] Kirsch, A., and Mitzenmacher, M. (2006). “Distance-sensitive bloom filters.” In Proc. the 8th Workshop on Algorithm Engineering and Experiments (ALENEX06), Miami, Florida, USA, January.

[11] Kumar, A., Xu, J., Wang, J., Spatschek, O., and Li, L. (2004). “Space-code bloom filter for efficient per-flow traffic measurement.” In Proc. 23rd IEEE INFOCOM, pages 1762EEE INFOCOM for efficient.

[12] Laufer, R. P., Velloso, P. B., and Duarte, O. C. M. B. (2005). “Generalized bloom filters.” Technical Report Research Report GTA-05-43, University of California, Los Angeles (UCLA).

[13] Liu, C. C., and Ma, H. (2006). “String Filtering of Large String Collection on Mobile Device using a Neural Network.” Proceedings of the 9th Joint Conference on Information Sciences.

[14] Ma, H. (2008). “Fast blocking of undesirable web pages on client PC by discriminating URL using neural networks.” Expert Systems with Applications, vol.34, pp.1533-1540.

[15] Ma, H., amd Tsai, T.F. (2009). “Fast Determining Existence of Text Strings in a Large Dataset Using CMAC Mapping.” International Journal of Information and Management Sciences, vol. , No. , pp. -.

[16] Mitzenmacher, M. (2002). “Compressed bloom filters.” IEEE/ACM Trans. Networking, 10(5):604ans.

## 七、計畫成果自評

近年來無線感測網路(Wireless Sensor Network)、普及計算(Pervasive Computing/Ubiquitous Computing)概念相當熱門，此概念強調與環境合而為一，並且融入日常生活中的每件事物與活動。然而無線感測網路與普及計算概念的影響下，低效能且低成本處理運算設備以分散式架構嵌入於各許多電器設備(如:手機、智慧型冰箱、家用感測裝置...等)，而需大量運算處理的應用需求也面臨著考驗。面對處理運算發展的趨勢，必須提高處理運算效率、降低硬體需求規格。本研究因應此趨勢，發展出具有快速搜尋識別的演算法，除了具備高效能搜尋效能外，對於硬體資源需求低，適用於低階處理運算環境。

本研究以低資源需求的環境為背景，考量其應用環境、特性與需求，以警方查緝贓車的需求作為實驗環境，符合其原計畫之構想。然而目前警方使用之查緝贓車的手持裝置屬於固定用途，因此無法在該裝置平台開發應用系統。本研究所發展之多目標屬性演算法不僅可應用於車牌查緝，對於資料量龐大的字串查詢亦可發揮其效率，不僅可查詢是否存在於資料集合，更提供查詢目標的分類屬性，應用範圍包含:資料庫前處理查詢、病毒碼識別、非法網站(URL)過濾、網路封包過濾、生物特徵辨識、物流貨物分類...等。本研究團隊將持續提升其處理運算效率及開發新應用領域，並且發表相關學術期刊與論文研究報告。