多重資料串流環境序列樣式探勘之應用--以台灣股市為例

趙景明
東吳大學資訊管理學系 教授
chao@csim.scu.edu.tw

楊慧雯
東吳大學資訊管理學系 碩士生
michelle_yang@yahoo.com

摘要

資料變化快速、即時需求提高，資料串流因而興起，其又可分為單資料串流與多重資料串流，多重資料串流可在單一時間處理一整個資料集(Itemset)，提供更即時的分析。因股票資料具有網路公開、數量龐大、更新快速等特點，無疑是具有代表性與實用性的應用[13, 17]，故本研究要建置一個股票探勘系統，運用多重資料串流技術可處理大量資料、即時動態產出分析結果的特色於系統中，將每一檔股票視為一個資料串流，在不同股票間進行多重資料串流序列樣式探勘，從股價的歷史記錄預測未來走勢，週期性探勘不同股票間漲跌的相對順序，幫助投資人掌握即時股市行情，增加獲利機會。

關鍵詞：序列樣式探勘、資料串流、多重資料串流、股市分析

1. 前言

資訊科技迅速發展，現今的交易資料，都以數位化的方式來儲存，大量的數位資料無法提供決策者直接有效的參考。由於資料探勘(Data Mining)能從大量資料中取出隱含、未知且有用的資訊[11]，因此得到愈來愈廣泛的應用[19, 21, 23]。資料探勘的技術主要有以下四種：關聯規則探勘(Association Rule Mining)、分類(Classification)、分群(Clustering)以及序列樣式探勘(Sequential Pattern Mining)；序列樣式探勘的概念是要找出隨著時間或是特定順序而經常發生的序列樣式，例如：在網頁存取上，探勘網頁使用者的瀏覽習慣，來預測下一個可能存取的網頁，以便預先擷取，增快其網頁讀取速度。

由於資料變化快速、即時需求提高，資料串流(Data Stream)因而興起，其具有：(1)輸入的資料是無窮 (2)資料持續快速到達 (3)主記憶體有限 (4)資料無法永久儲存，而且只能作一次處理 (5)當使用者有需求時，其分析結果能立即產生 (6)分析結果的偏差，必須在使用者能容許的範圍等六項特質[15]。資料串流又可分為單資料串流與多重資料串流(Multiple Data Streams)，單一時間處理單一資料(Item)已不足以因應多重資料串流環境的變化；在多重資料串流中，單一時間可處理一整個資料集(Itemset)，提供更即時、更準確的分析。單資料串流與多重資料串流差異如下：(1)同一時間的串流數不同 (2)多資料串流的資料量較大 (3)多資料串流可視為不同地點，具有資料同步特性 (4)多資料串流可產生串流序列(Stream Sequences)。

在多重資料串流探勘領域中，關聯規則、分類、分群等技術上已有不少研究，但在序列樣式探勘技術上鮮少找到相關研究；因此，本研究針對多重資料串流序列樣式探勘之應用進行研究，希望能夠實際應用到現實世界中。

股票資料具有網路公開、數量龐大、更新快速等特點，所以無疑是具有代表性與實用性的應用[13, 17]。由於影響股價的因素眾多，使

得股票市場難以預測，有些分析因加入過多複雜及無法客觀量化的資訊後，造成過度擬合(Over Fitting)，不見得能有效分析，而投資人買賣股票皆希望能夠維持高報酬並降低風險[25]。ICspan (Incremental Mining of Closed Sequential Patterns in Multiple Data Streams)演算法是以漸進式的方法不斷的探勘多重資料串流下的封閉序列樣式[6]；而 IAspam (Incremental Across - streams Sequential Patterns Mining)演算法則是以位元映射法尋找跨串流間的序列樣式[7]。本研究將運用這兩個演算法，建置一個多重資料串流序列樣式探勘系統(Multiple Data Streams Sequential Pattern Mining System)。將每一檔股票視為一個資料串流，在不同股票間進行多重資料串流序列樣式探勘，從股價的歷史記錄預測未來走勢，週期性探勘不同股票間漲跌的相對順序，幫助投資人掌握即時股市行情，增加獲利機會。

本研究主要由以下幾個章節章節所組成：第一章主要說明本研究的背景、動機、目的及論文架構；第二章將探討相關文獻，包括序列樣式探勘及資料串流環境下的序列樣式探勘；第三章則是說明研究方法，包括系統架構及所運用的演算法；第四章說明系統實作，包括系統環境、樣本取樣、方法流程；最後第五章作結論。

2. 文獻探討

本研究主要在多重資料串流環境中，建置一個序列樣式探勘系統，用以探勘股票資料。因此本章將針對相關文獻進行探討，首先第一節介紹序列樣式探勘演算法，瞭解序列樣式探勘後；接下來第二節將介紹在串流環境中，單串流及多串流的序列樣式探的勘技術與發展；最後介紹本研究所使用的 ICspan[6]及 IAspam[7]演算法。

2.1 序列樣式探勘

序列樣式(Sequential Patterns)探勘的概念是由 Agrawal and Srikant 在 1995 年所提出[1]，如：在網頁上，可先預測使用者下一個可能存取的網頁，預先擷取，以加快網頁讀取速度。針對此問題提出演算法，主要做法是先由序列資料庫中找出所有長度大小為 1 的頻繁序列，然後再以此頻繁列為基礎，藉由結合的方式產生新的候選序列，並再次進行資料庫掃描，如此反覆進行結合、掃描直到找出所有序列樣式為止，圖 1 是 AprioriAll 的演算法及其候選序列產生程序。AprioriAll 演算法雖然簡單易懂，但是在效率上卻不好，因此，在隔年 Srikant and Agrawal 提出 GSP (Generalized Sequential Patterns)一般化的序列樣式探勘演算法[26]，來改善 1995 年 AprioriAll 的缺點，在產生候選序列(Candidate Sequences)的步驟上改善執行效率，並在資料序列數量的線性延伸上有較好的延展性。接著 Zaki 於 2001 年提出的 SPADE (Sequential Pattern Discovery using Equivalence Classes)演算法[28]，它是屬於 Apriori 家族之一，SPADE 演算法是運用 ID-list 做交集運算，利用 lattice 階層式理論將原有的問題分解(Divide)成較小的子問題，獨立進行探勘，能處理大量資料僅需使用較少的記憶體，以垂直資料庫(Vertical Databases)的資料排序形式取代 AprioriAll 及 GSP 二方法[1, 26]的平行資料庫(Horizontal Database)，使其在執行時間效能上更優於 GSP 演算法。

AprioriAll 演算法

1. LS_1=大型 1-序列所成的集合;
2. **For (**k=2; $LS_{k-1} \neq \phi$; k++) **do begin**
3. CS_k=Candidate_seq_gen(LS_{k-1});
4. **For each** 在資料庫中的顧客序列 C **do**
5. 對於 CS_k 中的每一個候選序列 S，若 C 包含 S，則將 S 的支持個數增加 1;
6. LS_k=在 CS_k 中滿足最小支持個數的候選序列所成的集合;
7. **End**
8. **return** 所有大型序列中的最大序列

圖 1 AprioriAll 演算法

Lin and Lee 在 1998 年提出的 FASTUP (Fast Sequential Pattern Update Algorithm)演算法[18]，為了避免重新探勘(re-mining)整個資料庫，是以漸進式探勘交易資料庫，減少了對舊資料的重覆掃描，將新進資料庫(Increment Database)所探勘的頻繁序列樣式新增至原始資料庫(Original Database)所探勘的頻繁序列樣式中，以更新序列樣式與支持度，節省重新探勘的時間。雖然前面提到的方法提高不少效率，大量產生候選序列(Candidate Sequences)會降低探勘效能，尤其是遇到序列資料庫相當大、探勘出的序列樣式相當多或相當長時就會遇到瓶頸，因此，Han and Pei 在 2000 年提出 FreeSpan (Frequent pattern-projected Sequential pattern mining)演算法[14]，以頻繁的序列樣式和投射序列樣式資料庫來搜尋與產生新的序列樣式片段，限制搜尋頻繁序列次數和子序列的成長，用以減少一些候選序列樣式的產生。Pei and Han 在 2001 年提出 PrefixSpan (Prefix-projected Sequential Pattern Mining)演算法[22]，它主要是在序列樣式探勘中探勘出前序投射(Prefix Projection)，序列資料庫被遞迴地投射成較小的投射資料庫(Projected Databases)集合，並且在各個投射資料庫中，序列樣式的增長(grow)都只透過探勘出的局部(Local)片段；PrefixSpan 演算法可探勘出完整的樣式並且大大減少候選序列的產生，前序投射可減少投射資料庫的大小，使得探勘效能變的更有效率。

雖然 PrefixSpan 可有效減少候選序列的產生，但是當序列資料庫龐大、支持度較低或樣式長度較長時，探勘效率便會大幅下降，因此 Yan 等學者在 2003 年提出 CloSpan (Closed Sequential Pattern Mining)演算法[27]來解決此問題，其為引用 PrefixSpan 的概念以投射方式來探勘封閉式的序列樣式，提出搜尋空間刪減(Search Space Pruning)法來減少多餘重覆的序列樣式，並在投射的過程中以雜湊表(Hash Table)來記錄探勘出的封閉(Closed)序列樣式，以增快其搜尋序列辭彙樹(Lexicographic Sequence Tree)的速度。

Ayres 等學者在 2002 年提出 SPAM (Sequential Pattern Mining) [2]演算法，是以位元映射表示法探勘序列樣式，其延伸候選序列可分為兩個步驟：項目延伸步驟(Itemset-extension Step, I-step)和序列延伸步驟(Sequence-extension, S-step)。在這兩個延伸步驟上，使用位元映射表示法(Bitmap Representation)將序列資料庫中的項目集合轉換成 1 和 0，並直接以邏輯運算 AND 的方式快速執行 I-step 和 S-step，這兩個步驟所產生的結果可直接對應至序列資料庫找到序列所在，依此方法可在探勘大量資料集上可節省探勘時間。PrefixSpan 與 SPAM 演算法雖不會產生候選序列，但會有部分子集合重疊的情形，就如同重複掃描相同資料，Chang 等學者在 2007 年提出 IMCS (Incremental Mining of Closed Sequential Patterns)演算法[5]，以漸進式探勘封閉式序列樣式，探勘過程中是由封閉式序列樹(Closed Sequence Tree)來保留封閉式序列樣式，以不同類別的節點儲存不同狀態的序列樣式，並將新的探勘結果更新至封閉式序列

樹中，再改變節點狀態(Node State)。

2.2 資料串流環境之序列樣式探勘

現今隨著資料串流環境的興起，傳統的資料庫探勘演算法已經無法適用於資料串流環境中，因此便趨向於在資料串流環境中作資料探勘，不僅是在關聯規則、分類、分群等方面都相繼提出資料串流環境的探勘技術，就是序列樣式探勘也不例外。

首先談到 Oates and Cohen 在 1996 年提出的 MSDD (Multi - Stream Dependency Detection)演算法[20]，應用在多重資料串流架構中尋找資料串流間的依賴性規則，即特定事件發生於固定時間範圍內的規則。Golab and Ozsu 在 2003 年提出資料串流管理系統[13]，重點放在應用需求，是探討資料串流的特性與模型以及資料串流的連續查詢語義(Query Semantics)，並說明在現實環境中資料串流的應用，由於資料串流環境是一種新型態的資料環境，因此有許多改良的資料探勘演算法被提出以適用於資料串流環境。基於漸進式探勘會保留先前的探勘結果，導致探勘結果若長久時間未被更新時，有資料過期的問題發生，舊的探勘結果長久被儲存在記憶體，造成記憶體的浪費，為解決前述問題，許多學者提出資料串流環境探勘的遞減機制，是利用遞減機制減少記憶體的使用量；一般的刪減策略會在滑動視窗往後移時會刪減掉最早的交易資料，Chang and Lee 在 2004 年提出一個 estDec 演算法[3]是藉著遞減舊交易資料的權重，在資料串流環境中探勘頻繁項目集的遞減機制(Decay Mechanism)，依資料串流環境的特性，舊的探勘資料長久未在資料串流中出現時，必須遞減其支持度，以減少處理時間以及記憶體的耗用並確保探勘結果的正確性。

Chen 等學者在 2005 年提出 MILE 演算法[8]來改良[9, 20]兩篇論文的缺點，其主要是建立在 PrefixSPAN 演算法的前序投射概念上，是以一次性(one-time fashion)的探勘方式來對一段期間內的多重資料串流進行探勘，尋找多串流資料的前序與後序序列，以減少候選序列的產生與合併相同 prefix 的序列樣式，當有新的串流資料輸入時，它只能針對新的串流資料進行探勘，而無法將新舊探勘結果整合以得到更精確的探勘結果，因此會花較多時間在重新探勘。隨著資料探勘在各個領域研究發展，Chang and Lee 在 2005 年提出的 eISeq 演算法[4]，該方法可靈活的取捨記憶體使用和探勘準確性，可以在短時間內抓住(Catch)序列資料串流的最近變化情形；並透過一個遞減機制(Decaying Mechanism)緩慢地(Gracefully)丟棄(Discarding)可能不再有用的(Useful)舊資訊。Ho 等學者在 2006 年提出的 IncSPAM (Incremental Mining of Sequential Patterns using SPAM)演算法[15]，為單一資料串流環境的序列樣式探勘演算法，使用移動視窗來擷取串流資料，再轉換成顧客位元向量陣列(Customer Bit-vector Array with Sliding Window, CBSW)，它是利用漸進式的方式探勘資料串流中的序列樣式，此外，為了處理過期(Out-of-date)資料所導致的誤測(False Positive)問題，也改良 Chang and Lee [3] 所提出的遞減率(Decay-rate)概念，另外在探勘過程中，以 I-step 和 S-step 產生候選序列，再以位元映射方式迅速地計算支持度並且找到頻繁序列樣式。

Raissi 等學者在 2006 年提出 SPEED (Sequential Patterns Efficient Extraction in Data Streams)演算法[24]，以獨創性的探勘方法，用一種新的資料架構，來保留頻繁序列樣式，加上快速剪枝策略，令使用者可以在任何時間尋找任意(Arbitrary)時間區段(Time Interval)的最大序列樣式，且保證近似結果誤差不會超過使用者所定義的門檻值。Ezeife and Monwar 在 2007 年所提出的 SSM 演算法[10]亦為在資料

串流中漸進式探勘頻繁序列樣式，可以用於分析電子商務的資料，其中的主要資料來源是點擊流；在探勘過程中使用 D-list 結構有效地儲存和維護所有項目的支持度，並建置 PLWAP 樹有效地批次探勘序列樣式，再將其儲存至 FSP-tree 中並維護此批次性頻繁序列樣式，隨著批次處理資料串流，FSP-tree 會漸進式更新探勘結果。Li and Chen 在 2008 年所提出 DSOSW 演算法[17]，使用新的刪減策略來減少運算成本，在框架移動時，利用項目集辨識度去辨識出新的項目集合與舊的項目集合使否大於使用者所設的最低門檻值，執行刪減策略，不僅提升探勘效率，又不失探勘精確度。

Chao and Chen 在 2010 年所提出 ICspan 演算法[6]，以漸進的方式不斷探勘封閉序列樣式，來解決因無法保留先前的序列樣式探勘結果，導致探勘的結果不夠精確的問題。Chao and Lin 在 2010 年所提出 IAspam 演算法[7]，與一般多重資料串流序列樣式探勘演算法的差異在於，它是探勘介於串流之間的跨串流序列樣式，是一種新類型的序列樣式。

2.3 ICspan 演算法

之前關於多重資料串流序列樣式探勘的研究，如：MILE 探勘演算法，無法保留先前的序列樣式探勘結果，導致探勘的結果不夠精確[8]；SPAMDAS (Sequential Pattern Mining in Multiple Data Streams) [6]以漸進的方式不斷探勘序列樣式來解決此問題，主要分為資料取樣與漸進式探勘兩個階段。這個概念在序列樣式越多的清況下越能減少記憶體空間的耗費，讓 ICspan 演算法可以提供更精確的序列樣式結果給使用者參考與決策，而實驗結果也證明 ICspan 演算法能夠有效減少序列樣式的記錄進而降低記憶體使用量，以及在資料不斷輸入的情況下能夠維持良好的探勘效率，圖 3 為 SPAMDAS 的整體流程。

分別說明如下：

- 資料取樣

採用時間導向的移動框架(Time-sensitive Sliding Window)對輸入的串流資料進行不重複(Non-overlapping)取樣；另外，為了探勘出一定週期(例如：1 分鐘、1 小時、1 天等)的序列樣式，我們限制頻繁序列樣式的時間跨度不大於固定時間區段，因此我們將移動框架內的資料依固定時間區段大小切割成若干個相同長度的框架(這些框架稱為基礎框架)，接著就能透過比對這些基礎框架內的序列資料來找出一定週期的頻繁序列樣式。

- 漸進式探勘

在漸進式探勘階段中，先對移動框架裡的不同基礎框架資料進行比對以尋找封閉序列樣式，接著將新舊序列樣式進行組合與更新以產生精確的序列樣式探勘結果。

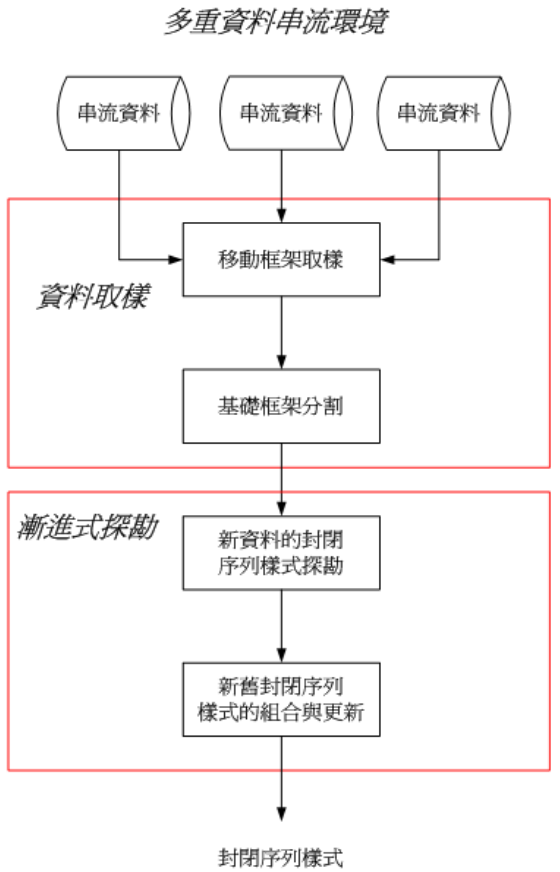

圖 2 SPAMDAS 的整體流程

Algorithm: ICspan

Input: SW', SW, min_sup, r, F, SF

Output: F and SF

1. F' $\leftarrow \phi$; SF' $\leftarrow \phi$;
2. Scan SW' to find 1-item sequential patterns and add into F' or SF';
3. **For each** 1-item sequential pattern in F' or SF' **do**
4. transform into closed sequential patterns and add into F' or SF';
5. **For each** sequential pattern p in F or SF **do**
6. **If** $sup_{sw}(p) + sup_{sw'}(p) \geqq$ min_sup **then**
7. **If** p is a closed sequential pattern **then** add p into F';
8. **If** $sup_{append}(p) \geqq (1-r)*$min_sup **then**
9. transform into closed sequential patterns and add into F' or SF';
10. **If** $(sup_{sw}(p) + sup_{sw'}(p) \leqq$ min_sup) **and** $(sup_{sw}(p) + sup_{sw'}(p) \geqq r*$min_sup) **then**
11. **If** p is a closed sequential pattern **then** add p into SF';
12. F←F'; SF←SF';
13. **return;**

圖 3 漸進式封閉序列樣式探勘演算法

ICspan 演算法的步驟分為兩個部份。第一部分是新序列樣式的尋找(圖 4 的 line 1-4)，這個部分是要找出新資料裡的封閉序列樣式，所以對長度為 1 的序列樣式進行投射，以找出新的頻繁封閉序列樣式；第二部分則是新舊序列樣式的整合(圖 4 的 line 5-13)，這個部分是對第一部分找到的新樣式與先前記錄之樣式進行投射，以尋找是否會有新舊項目組合而成的序列樣式。

2.4 IAspam 演算法

基於商業因素考量，業者可能需要找到不同串流之間的關聯性以提供使用者更完善的服務，先前的研究只能處理單一時間單一資料，並不足以因應多重資料串流環境的變化，所探勘出來的序列樣式可能會存在於不同的串流中但被視為同一個串流之序列樣式。IAspam 演算法與一般多重資料串流序列樣式探勘演算法的差異在於，它是探勘介於串流之間的跨串流序列樣式，而跨串流序列樣式是一種新類型的序列樣式，圖 5 為 ASPAMDAS 的整體流程[7]，ASPAMDAS (Across - streams Sequential Pattern Mining in Multiple Data Streams)主要分為資料取樣與漸進式探勘兩個階段：

- 資料取樣

在資料串流環境中，由於即時性知識需求提高和資料串流特性的限制，相對地，越新的資料便顯得越重要，因此我們使用移動框架來擷取新的串流資料，並將處理完的舊串流資料丟棄，移動框架只會保留最新 N 筆的交易資料，N 為移動框架的框架大小(Window Size)，每次處理完框架內的交易資料後，框架便會往前移動一個時間點，以擷取最新的串流資料做處理。

- 漸進式探勘

在漸進式探勘階段中，將移動框架中的顧客交易資料轉換為位元映射來尋找跨串流序列樣式，以提升探勘效率，解決在處理大量資料時可能會造成探勘效率不佳的問題；接著再針對新舊跨串流序列樣式進行更新或刪除以產生精確的跨串流序列樣式探勘結果。

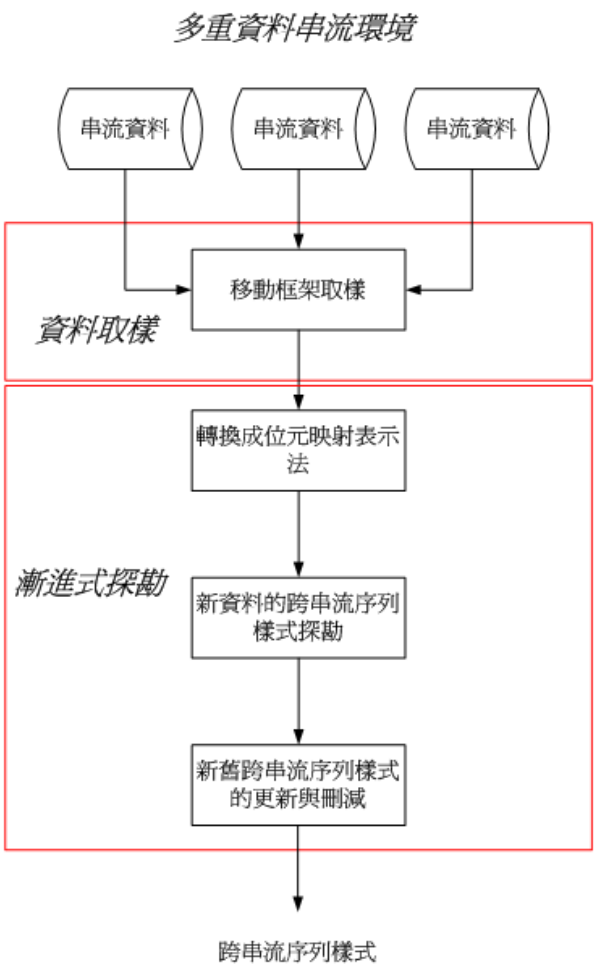

圖 4 ASPAMDAS 的整體流程

Algorithm: IAspam

Input：SW', SW, min_sup, CID, S, I, k-item, CS, L-tree

Output：FASP, StP

1. **For each** transaction data of SW' **do**
   store into CS as <S, I> and order by CID;
2. **For each** I of CS **do**
   turn into bit-vector matrix;
   update time point column of k-item;
3. **For each** 1-item **do**
   add 1-item and sup_{SD}(1-item) into L-tree ;
   If sup_{SD}(1-item) ≧ min_sup **then**
   generate CP with I-step and S-step and map to bit-vector matrix;
4. **For each** CP **do**
   If $sup_{sw'}$(CP) ≧ min_sup **then**
   If CP exists in L-tree **then**
   update sup_{sw}(CP);
   else add FASP and StP into L-tree;
5. **For each** FASP in L-tree **do**
   If sup(FASP) not updated **then**
   decay sup(FASP) in L-tree;
   If sup(FASP) ≦ min_sup **then**
   eliminate FASP;
6. **return**;

圖 5 漸進式跨串流序列樣式探勘演算法

IAspam 演算法的步驟分為三個部份。第一部份是轉換成位元映射表示法(圖 6 的 Step 1-2)，將顧客交易資料轉換成位元映射矩陣；第二部份是找出新資料裡的跨串流序列樣式(圖 6 的 Step 3)，對 1-item 進行 I-step 與 S-step 以產生候選序列並對映至位元映射矩陣中；第三部份是新舊序列樣式的整合(圖 6 的 Step 4-5)，對第二部分找到的新樣式與先前記錄之樣式進行比對，以更新序列樣式，並且遞減未被更新的序列樣式支持度。

3.研究方法

隨著資料串流環境的改變，單一時間處理單一資料並不足以因應多重資料串流環境的變化，本研究將建置一個多重資料串流序列樣式探勘系統，用以探勘股票資料。首先第一節我們先說明系統架構；第二節介紹系統環境建置；第三節介紹樣本選取的方法與邏輯；第四節介紹方法流程。

3.1 系統架構

本節將介紹本研究的系統架構，如圖 2 所示。首先先收集所需的資料，並進行前置處理

後；再將資料執行資料清潔與轉換後，成為我們所需的資料串流，接著執行多重資料串流序列樣式探勘，以 IAspam 和 ICspan 兩個演算法，進行序列樣式探勘，產出封閉序列樣式及跨串流序列樣式，以此來預測股價，最後針對所完成的系統，進行系統評估。

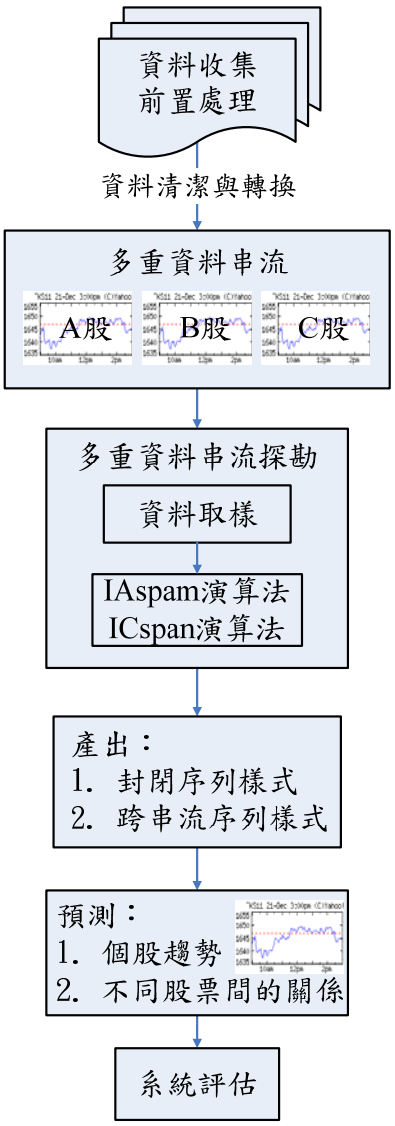

圖 6 系統架構圖

資料來源：本研究整理

3.2 系統環境

本研究使用以下環境來執行多重資料串流序列樣式探勘：

(1) 實驗平台：

(a) 處理器：Intel® Pentium® 4 CPU 2.80GHz

(b) 記憶體：1G bytes

(c) 作業系統： Microsoft Windows XP Professional

(2) 實驗資料庫：Microsoft SQL Server 2005

(3) 資料來源：TEJ 財經資料庫[29]

3.3 樣本選取

本研究以台灣股市為研究對象，所選取的股票為台灣證券交易所裡台灣 50 指數的 50 檔股票[30]，台灣 50 指數涵蓋臺灣證券市場中市值排名前 50 大的台灣龍頭上市公司做代表，如表 1, 2，其與大盤的關連性高達九成，代表藍籌股之績效表現，同時也是臺灣證券市場第一支交易型指數，本研究將選取從 2008 年 1 月 1 日到 2008 年 12 月 31 日，共 249 天的資料作為樣本，總計有 12,450 筆資料。

表 1 台灣 50 指數介紹

指數名稱	臺灣 50 指數
編製機構	臺灣證券交易所編製
成分股選擇之標準	選取在臺灣證交所市值前 50 大之上市公司為成份股。
成分股調整	調整時間：1, 4, 7, 10 月第 3 個星期五收盤納入成分股市值排名。

資料來源：寶來投信[31]

表 2 研究對象

序號	代號	名稱	平均市值(百萬元)	序號	代號	名稱	平均市值(百萬元)
1.	2330	台積電	1,459,073	26.	2352	佳世達	35,379
2.	2317	鴻海	958,909	27.	2015	豐興	33,999
3.	2002	中鋼	492,131	28.	2315	神達	31,582
4.	1303	南亞	482,192	29.	1504	東元	28,358
5.	1301	台塑	423,171	30.	2204	中華	28,313
6.	1326	台化	364,494	31.	2344	華邦電	26,335
7.	2303	聯電	207,117	32.	2014	中鴻	26,253
8.	1402	遠東新	176,510	33.	2323	中環	25,848
9.	2308	台達電	175,602	34.	2103	台橡	25,590
10.	2325	矽品	138,651	35.	2332	友訊	23,630
11.	1216	統一	138,090	36.	1717	長興	22,528
12.	2311	日月光	135,160	37.	1314	中石化	22,131
13.	1101	台泥	129,518	38.	2328	廣宇	22,106
14.	1102	亞泥	116,657	39.	2101	南港	21,256
15.	2324	仁寶	112,244	40.	1440	南紡	19,509
16.	1722	台肥	99,005	41.	1907	永豐餘	18,672
17.	2301	光寶科	76,084	42.	1704	榮化	18,358
18.	2347	聯強	75,809	43.	2023	燁輝	18,069
19.	2105	正新	63,034	44.	1723	中碳	17,328
20.	1802	台玻	53,790	45.	1503	士電	17,094
21.	1434	福懋	46,960	46.	2349	錸德	16,050
22.	2201	裕隆	46,861	47.	1210	大成	15,198
23.	2006	東鋼	41,197	48.	2106	建大	10,785
24.	2337	旺宏	40,545	49.	1319	東陽	10,015
25.	1605	華新	39,856	50.	1201	味全	9,482

3.4 方法流程

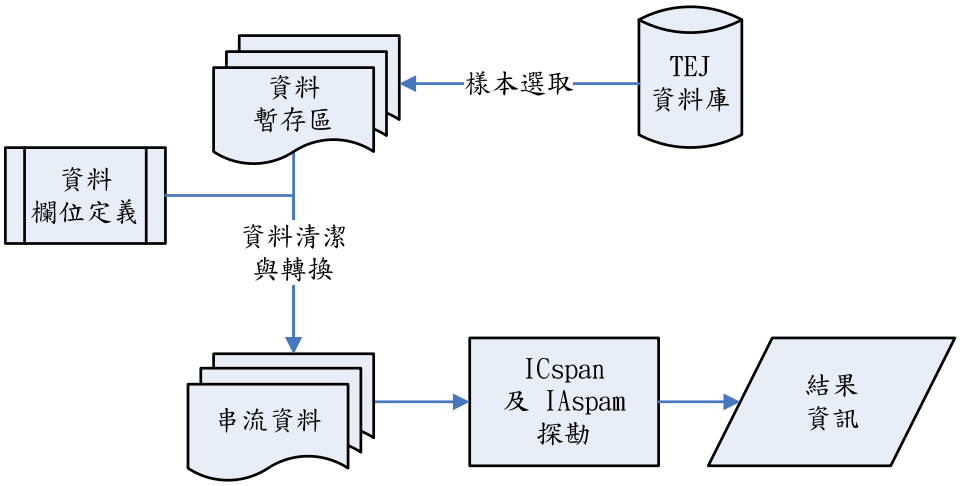

圖 7 方法流程圖

資料來源：本研究整理

本研究的方法流程圖如圖 7 所示，分別說明如下：

首先由 TEJ 財經資料庫篩選並匯出台灣 50 指數從 2008 年 1 月 1 日到 2008 年 12 月 31 日的股價資料，如圖 8。

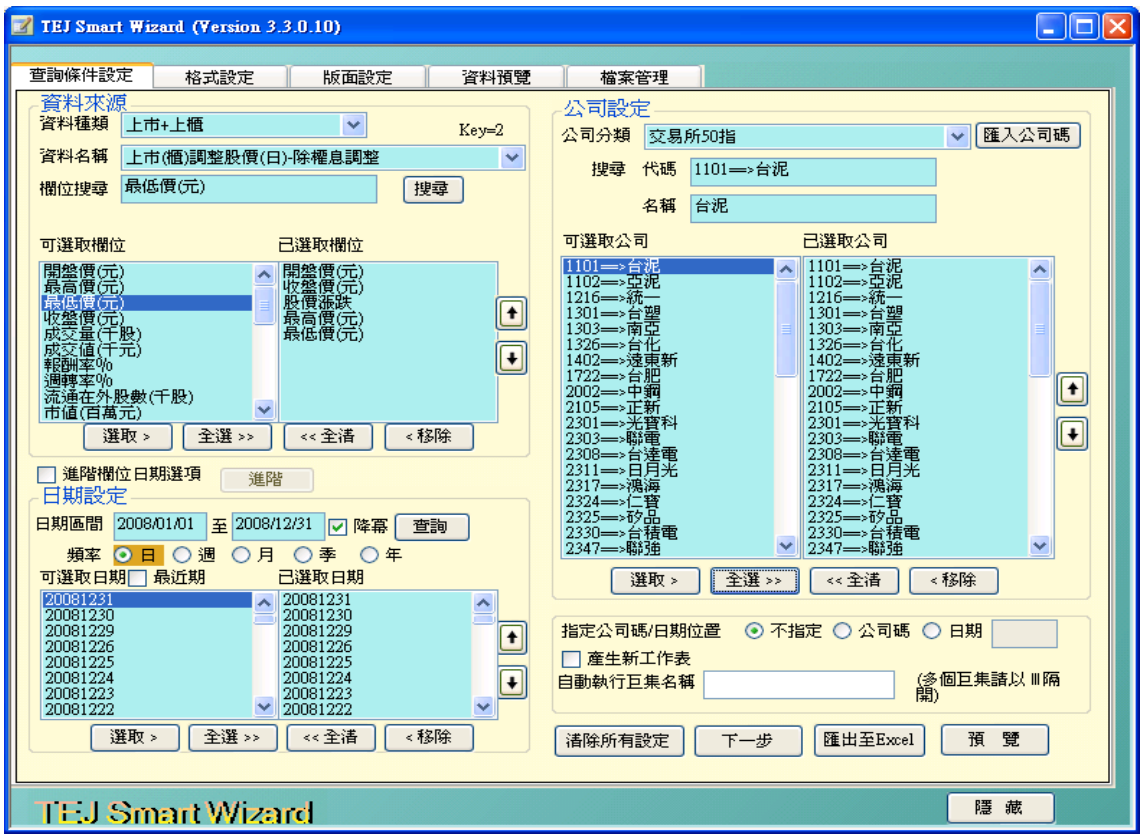

圖 8 TEJ 財經資料庫之 TEJ Smart Wizard 操作畫面示意圖

再來將所匯出的資料，匯入 SQL Server 所建的 SP_Mining 資料庫中，如表 3 資料表名稱為歷史股價，50 檔股票 249 天的股價記錄，總計有 12,450 筆資料，如圖 9。

表 3 歷史股價資料表說明

資料表名稱	歷史股價暫存檔 stock_history	
說明	存放股價的歷史資料	
序號	欄位名稱	資料型態
1	證券代碼	nvarchar(255)
2	年月日	datetime
3	開盤價(元)	float
4	收盤價(元)	float
5	股價漲跌	float
6	最高價(元)	float
7	最低價(元)	float

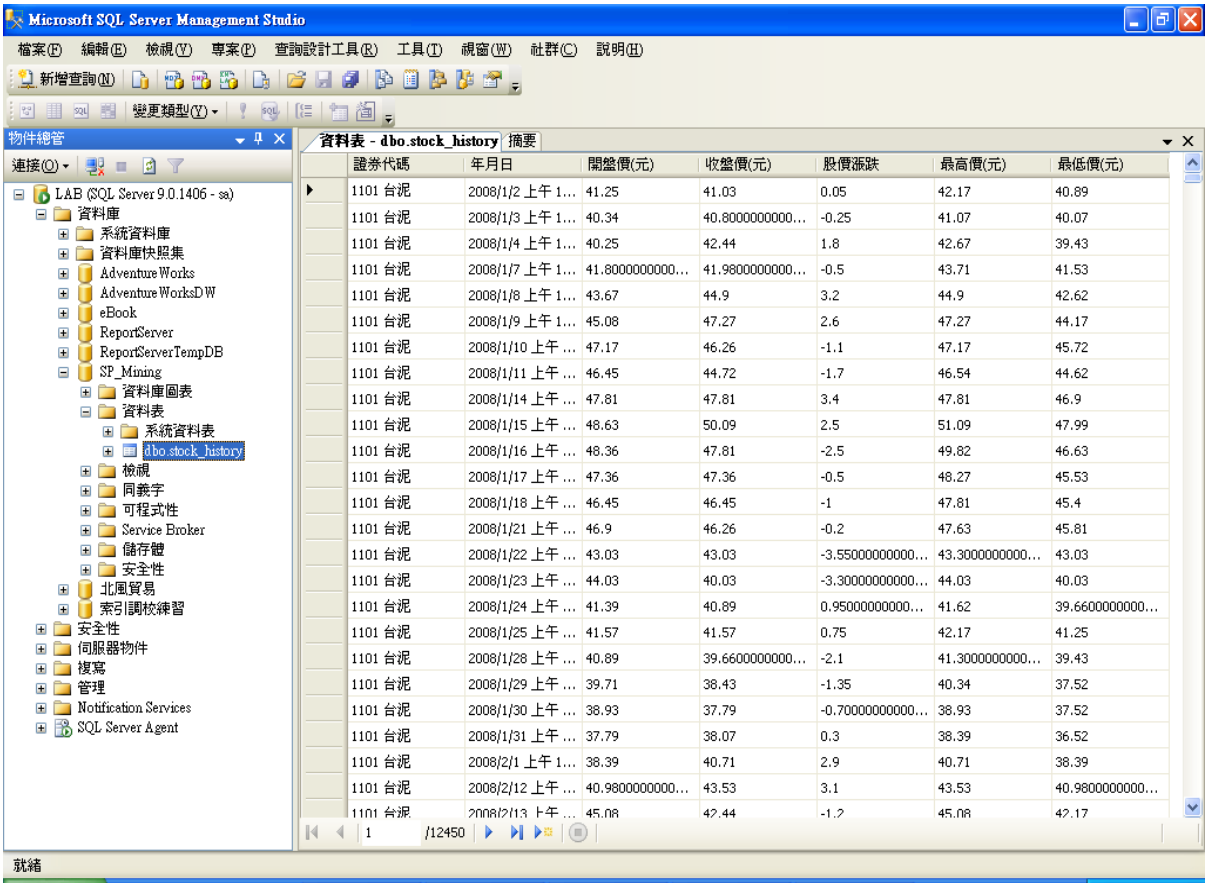

圖 9 歷史股價記錄暫存檔

接著執行資料清潔與轉換，我們將每一檔股票視為一個資料串流，其中狀態欄位用以判斷其當日漲跌幅，定義如表 4，將歷史股價資料，轉換為依證券代碼為資料表名稱的資料格式如表 5，執行資料清潔與轉換後，結果如圖 10，左邊每一個資料表即代表一個串流，右邊為其資料內容。

運用 ICspan 可從股價的歷史記錄，預估出

股價未來走勢；IAspam 可探勘出不同股票間，漲跌的相對順序，協助投資人投資決策時參考。

表 4 狀態欄位說明

值	意 義	說 明
+2	大漲	漲幅超過 3%
+1	小漲	漲幅 3% 以內
0	平盤	今日收盤價同前一日收盤價
-1	小跌	跌幅 -3% 以內
-2	大跌	跌幅超過 -3%

表 5 轉換後的資料串流

名 稱	同證券代碼	
說 明	每一檔股票即為一個資料串流	
序號	欄 位 名 稱	資 料 型 態
1	證券代碼	nvarchar(255)
2	年月日	datetime
3	昨收(元)	float
4	今收(元)	float
5	股價漲跌	float
6	百分比	float
7	狀態	float

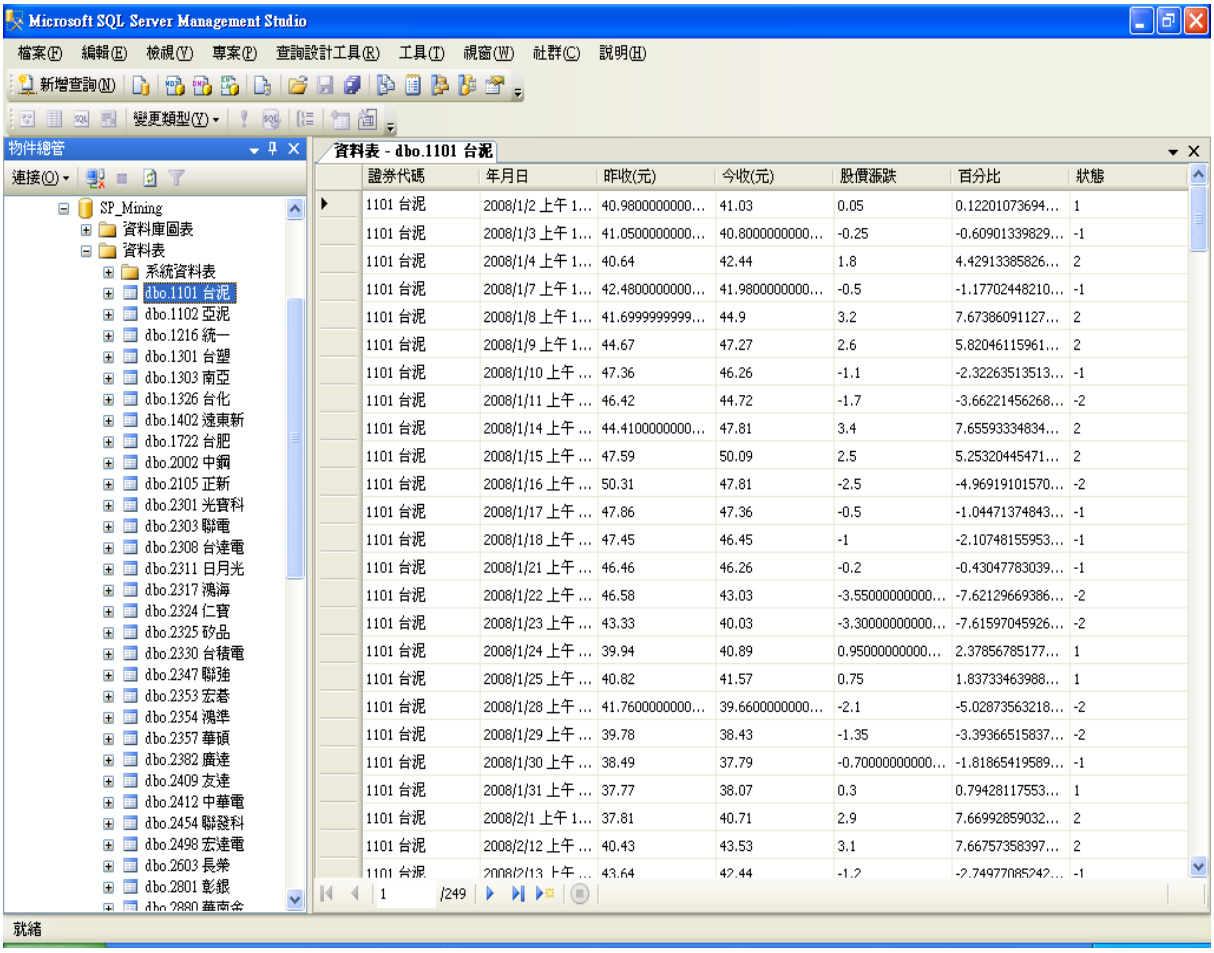

圖 10 轉換後的資料串流示意圖

4. 結論

本研究建置一個股票探勘系統，運用多重資料串流技術可處理大量資料、即時動態產出分析結果的特色於系統中，將每一檔股票視為一個資料串流，在不同股票間進行多重資料串流序列樣式探勘，從股價的歷史記錄預測未來走勢，週期性探勘不同股票間漲跌的相對順序，幫助投資人掌握即時股市行情，增加獲利機會。

參考文獻

[1] Agrawal, R. and Srikant, R., "Mining Sequential Pattems," in ***Proceedings of the 11th International Conference on Data Engineering***, pp. 3-14, Taipei, Taiwan, ROC, March 1995.

[2] Ayres, J., Gehrke, J., Yiu, T. and Flannick, J., "Sequential Pattern Mining using A Bitmap Representation," in ***Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining***, pp.429-435, Edmonton, Alberta, Canada, July 2002.

[3] Chang, J. and Lee, W., "Decaying Obsolete Information in Finding Recent Frequent Itemsets over Data Stream," ***IEICE Transaction on Information and Systems***, Vol. E87-D, No. 6, June, 2004.

[4] Chang, J.H. and Lee, W.S., "Efficient Mining Method for Retrieving Sequential Patterns over Online Data Streams," ***Journal of Information Science***, Vol. 31, Issue 5, pp. 420-432, October 2005.

[5] Chang, L., Yang, D., Wang, T. and Tang, S., "IMCS: Incremental Mining of Closed Sequential Patterns," in ***Proceedings of APWeb/WAIM 2007, LNCS 4505***, pp. 50–61, Huang Shan, China, June 2007.

[6] Chao, C. M. and Chen, W. T., "Incremental Mining of Closed Sequential Patterns in Multiple Data Streams," in ***Proceedings of 2010 International Conference on e-Commence, e-Administration, e-Society, e-Education, and e-Technology***, Macau, China, January 2010

[7] Chao, C. M. and Lin, Y. T., "Incremental Mining of Across-streams Sequential Patterns in Multiple Data Streams," in ***Proceedings of the 25th International Conference on Computers and Their Applications***, Honolulu, Hawaii, U.S.A., March 2010.

[8] Chen, G., Wu, X. and Zhu, X., "Sequential Pattern Mining in Multiple Streams," in ***Proceedings of 5th IEEE International Conference on Data Mining***, pp. 27-30, Washington, USA, November 2005.

[9] Das, G., Lin, K.-I., Mannila, H., Renganathan, G. and Smyth, P., "Rule Discovery from Time Series," in ***Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining***, pp. 16-22, New York, USA, August 1998.

[10] Ezeife, C.I. and Monwar, M., "SSM: A frequent Sequential Data Stream Patterns Miner," in ***Proceedings of IEEE Symposium on Computational Intelligence and Data Mining***, pp. 120-126, Honolulu, USA, March 2007.

[11] Frawley, W. J., Piatetsky-Shapiro, G. and Matheus, C. J., "Knowledge Discovery in Databases:An Overview," ***AAAI/MIT press***, 1991.

[12] Gavrilov, M., Anguelov, D., Indyk, P. and Motwani, R., "Mining the stock market: which measure is best?," in: ***Proceedings of the 6th ACM Int'l Conference on Knowledge Discovery and Data Mining***, pp 487-496, Boston, MA, August 2000.

[13] Golab, L. and Ozsu, M. T., "Issues in Data Stream Management," ***ACM SIGMOD*** Record Vol. 32, No. 2, pp. 5-14, June 2003.

[14] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U. and Hsu, M-C., "Freespan : Frequent pattern-projected sequential pattern mining," in ***Proceedings of the 6th ACM SIGKDD International Conference on Knowledge discovery and data mining,*** pp. 355–359, Boston, USA, August 2000.

[15] Ho, C.C., Li, H.F., Kuo, F.F. and Lee, S.Y., "Incremental Mining of Sequential Patterns over a Stream Sliding Window," in ***Proceedings of 6th IEEE International Conference on Data Mining***, pp. 677-681, Hong Kong, China, December 2006.

[16] Kargupta, H., Park, B., Pittie, S., Liu, L., Kushraj, D. and Sarkar, K., "MobiMine: Monitoring the Stock Market from a PDA," ***ACM SIGKDD Explorations***, Vol. 3, No. 2, pp. 37–46, January 2002.

[17] Li, H. and Chen, H., "DSOSW: A Deleting Strategy in Mining Frequent Itemsets over Sliding Window of Stream," in ***Proceedings of International Symposiums on Information Processing***, pp. 135-138, Moscow, Russia, May 2008.

[18] Lin, M.Y. and Lee, S.Y., "Incremental update on sequential patterns in large databases," in ***Proceedings of 10th IEEE International Conference on Tools with Artificial Intelligence***, pp. 24-31, Taipei, Taiwan, ROC, November 1998.

[19] Muthukrishnan, S., "Data Stream: Algorithms and Applications," ***Now Publishers***, March, 2005.

[20] Oates, T. and Cohen, P. R., "Searching for structure in multiple streams of data," in ***Proceedings of the 13th International Conference on Machine Learning***, pp. 346–354, Bari, Italy, July 1996.

[21] Parthasarathy, S., Zaki, M. J., Ogihara, M. and Dwarkadas, S., "Incremental and Interactive Sequence Mining," in ***Proceedings of the 8th International Conference on Information and Knowledge Management***, pp. 251-258,, Kansas City, MO Nov. 1999.

[22] Pei, J., Han, J., Mortazavi-Asl, B. and Pinto, H., "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," in ***Proceedings of the 17th Internatioanl Conference on Data Engineering***, pp.215-224, Heidelberg, Germany, April 2001.

[23] Peng, W. C. and Chen, M. S., "Mining User Moving Patterns for Personal Data Allocation in a Mobile Computing System," in ***Proceedings of 29th International Conference on Parallel Processing (ICPP2000)***, August 2000.

[24] Raissi, C., Poncelet, P. and Teisseire, M., "SPEED: Mining Maximal Sequential Patterns over Data Streams," in ***Proceedings of the 3rd International IEEE Conference Intelligent Systems***, pp. 546-552, Varna, Bulgaria, September 2006.

[25] Sharpe, W., "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk," ***Journal of Finance,***

1964, Vol.19, No.2, pp.425-442.

[26] Srikant, R. and Agrawal, R., "Mining Sequential Patterns: Generalizations and Performance Improvements," in ***Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology***, pp. 3-17, London, UK, March 1996.

[27] Yan, X., Han, J. and Afshar, R., "CloSpan: Mining Closed Sequential Patterns in Large Datasets," in ***Proceedings of 2003 SIAM Int. Conf. Data Mining on Data Mining***, pp. 438-457, San Francisco, USA, May 2003.

[28] Zaki, M. J., "SPADE : An Efficient Algorithm for Mining Frequent Sequences," in ***Proceeding of Machine Learning Journal, special issue on Unsupervised Learning***, Vol. 42, pp.31-60, 2001.

[29] TEJ 財經資料庫，http://www.tej.com.tw/twsite

[30] 台灣證券交易所，http://www.twse.com.tw/ch/

[31] 寶來投信，http://www.p-shares.com/_big5_index/petf3a.asp