

運用資料探勘技術於製程品質分析改善-以 LCD 驅動 IC 封裝產業為例

Application Data Mining Technique to improve Manufacturing Quality –A Case Study of LCD Driver IC Packaging Industry

摘要(abstract)

近年來由於專業的分工，以 LCD 驅動 IC 封裝廠來說，為提升產品良率及加速了解產品不良原因及找出對策，建置一產品品質問題分析模式，有效率，更快取得所需的資訊，並提高客戶對資訊系統的滿意度。

本研究以資料倉儲的星狀資料綱要作為線上分析處理的基礎，用資料探勘的決策樹方法及類神經網路方法來將封裝廠製造過程不良原因，建構一品質分析系統，提供問題分析的界面，能夠迅速掌握問題原因作出判斷，達到縮短整個品質問題解決的時間。研究的結果，使用決策樹(decision tree)方法及類神經網路(neural network)方法在降低內引腳(inner lead)不良顆數、樹脂(resin)問題不良顆數及晶片問題不良顆數方面均獲得改善，and 決策樹方法較類神經網路方法更適用於 LCD 驅動 IC 封裝產業之品質問題分類分析。結果亦可用於電腦組裝業、電子零件製造、半導體封裝製造業品質問題分析。

關鍵詞：客戶關係管理、資料探勘、資料倉儲、決策樹

1. 簡介(Introduction)

在半導體製造業講求專業分工製造，上下游產業供應鏈明顯，如何在產業中競爭，提昇客戶滿意度以及維持客戶忠誠度是企業競爭的關鍵因素。從 Burgess[9]的研究結果得知，製造業競爭策略的優先順序有「品質」(Quality)、「成本」(Cost)、「時間」(Time)以及「彈性」(Flexibility)等四項關鍵因素，因此企業如何提昇產品良率、提高產品的品質、降低生產製造的成本、滿足客戶交貨期限以及彈性製造需求等，這些都是刻不容緩的議題。

而在知識經濟的時代中，企業體認到顧客關係管理的重要以及利用資訊科技來提昇競爭優勢，近年來更有許多新的技術及觀念應用出

來，如人工智慧、資料倉儲、資料探勘等，藉由電腦化系統的運作，輔助企業進行資料處理、資訊分析，並整合各來源客戶資訊並快速回應與解決問題，推進到主動行銷與服務、協助企業作出各種合理的反應處理或決策動作[5, 10, 11]，亦即電腦化的境界可由最基礎資料處理為目的的思考，提昇並發展對資訊的管理，進而形成決策輔助的目的，而運用知識規則達到智慧決策最佳的目的[2]。

目前對於製造品質問題研究有針對半導體製程機台組合用分類式資料探勘，以避免產生品質不良的機台組合，提高半導體製程的良率[14]，以及利用決策樹方法，針對半導體晶圓允收測試及相關製程資料，發展製程事故診斷資料探勘[15]，及針對半導體生產製造資料，利用 SOM 以及決策樹方法，找出生產績效指標間的特徵與規則，以做為半導體廠製造決策之依據[16]。

而針對半導體驅動 IC 封裝產業中，目前尚無相關研究，因此充分整合及運用資訊及系統功能，建構一個運用資料倉儲及資料探勘的技術在分析製程異常的產生作為決策依據。透過基本及歷史資料取得建立，分析過去產品品質的問題資料，能夠快速提供找出產品問題原因及降低對問題處理因應時間，以提高客戶的滿意度。

2. 資料探勘

2.1. 知識發掘

資料探勘可視為資料庫知識發掘(Knowledge Discovery in Database ; KDD)的一部份，資料庫知識發覺是一連串可創造知識的程序，資料探勘只是其中的一個步驟[4, 13]。Brachman 等人[6]認為：「知識探索中所有的活動及過程，是為了從資料中找到有用的樣型，藉由運用資料探勘的演算法，以及對於知識的事後處理或再處理，以找到解決問題的關鍵原因。資料探勘實為整個知識發現過程中最重要的核心步驟，而對於整個資料探勘的過程，其實是不間斷地重複確認問題、分

析資料、採取行動、評估結果四個步驟[7]。

Pyle [8] 指出，利用資料探勘技術來發覺知識皆有其相似的地方。例如首要對欲解決的問題與需求做明確的定義，其次選擇目標資料庫，將對雜訊資料進行排除，並且對資料進行編碼來轉化資料顯示格式，以增加處理上的效能。

2.2. 資料探勘

在資料探勘的過程中，由於不同的問題定義會產生不同的探勘結果類型，以提供作為不同決策時使用。目前常見的問題類型如下：

1. 預測(Prediction):Berry 與 Linoff[7,12] 認為預測會根據某些未來的行為來分類，或推估某變數未來可能的值以及趨勢。
2. 關連規則(Association Rule): Berson[12]等認為關連規則是從資料庫探勘出滿足一定條件的依賴性關係。
3. 聚類(Clustering/Segmentation): Berson[12]等認為聚類可以根據一組特性，將資料庫區隔成幾個子集合。
4. 分類(Classification):分類所作的是為每一種類別找到一個合理的描述或模型，然後再利用這些分類的描述或模型，來對未知的新資料進行分類[7]。

3. IC 封裝製程品質問題

3.1. LCD 驅動 IC 封裝製程

隨著 IC 製程的快速微細化，傳統打線的封裝技術已無法應付高電性需求，而直接以晶片之凸塊(Bump)與基板連結的覆晶封裝技術卻能因應電子產品小型化與晶片性能高速化之趨勢，使得滿足輕、薄、短、小特性的晶片尺寸封裝(CSP)蔚為風潮。由於以 LCD 驅動 IC 的封裝為主要對象，LCD 驅動 IC 係製造液晶顯示器(LCD)零件之一，主要應用於筆記型電腦及液晶監視器大尺寸 TFT-LCD 面板上或行動電話、PDA、翻譯機等小尺寸 STN 面板上。

LCD 驅動 IC 封裝製程大致可以區分成以下幾道流程，其流程架構如圖 2 所示：

建構 LCD 驅動 IC 封裝製程產品品質問題分析系統，以品質的觀點而言，必須找出每個

工作站中可能會影響生產品質的因素，如表 1 所示主要為製造工作站中所包含的製程作業以及可能發生的缺陷。

3.2. 品質問題分析

品質分析系統主要是由資料庫、線上分析處理、資料探勘等元件組成，如圖 3 所示，主要分幾個層級：

在資料準備階段將進行資料彙整與篩選、資料清理與新資料產生與轉換等 3 個資料準備的動作。因此建立問題資料格式的主要欄位必須包括當初製造製令編號、屬性與問題。在資料屬性的部分，大致定義成 4 個屬性，分別為人員(man)、設備(machine)、材料(material)與方法(method)，一般俗稱 4M，其定義如下：

1. 人員因素：如未按照流程方法進行、漏檢、參數未設定等。
2. 設備因素：乃有關設備運作的問題如壓力過大、Tool 頭過高、Sensor 無感應等。
3. 材料因素：為相關於材料本身的問題如膠過濃、內引腳過細等。
4. 方法因素：如程式設計不良、指示不明確、未使用無塵布等。

將屬性變數及品質問題類別予以編號，以編號 0, 1, 2, 3, ... 的方式，屬性變數若為「無錯誤發生」則計為 0，依序再按編號排列之。同樣的，品質問題類別也以編號編排，即 1, 2, 3, ...，製作屬性變數及品質問題編號對照表，如表 2 所示：

個案公司之品質問題資料蒐集自 2002 年 01 月至 2003 年 12 月。資料總數為 9752 筆。問題類型總計 15 個類型，而資料探勘 4 個屬性下的個別變數個數分別為：人員 17 個變數，設備 13 個變數，材料 10 個變數，方法 12 個變數。

4. 品質問題探勘系統設計

(1). 決策樹演算法

決策樹是以樹狀資料結構為基礎的分類分析方法。其詳細的演算法如下所示：

Step1:設定根節點為目前節點 C，此時所有物件

都屬於 C 的物件集合。

Step2: 如 C 中所有的物件都屬於同一個類別，則設定節點 C 為此類別，然後停止，否則繼續執行 Step3。

Step3: 計算其熵值 $E(C)$ 。若集合 C 分屬於 j 個不同類別， $E(C)$ 為： $E(C) = -\sum p_j \log_2(p_j)$ 其中 $p_j = (\text{屬於類別 } j \text{ 的總物件數}) / (C \text{ 的總物件數})$ 。

Step4: 針對所有尚未出現在從根節點到目前節點之路徑的屬性 A_i (稱之為候選屬性)，以 A_i 分別對 C 的物件集合進行分割，並計算所導致之部分決策樹的熵值 $E(A_i)$ 與資訊獲利 $G(A_i) = E(C) - E(A_i)$ 。選擇某一屬性 A_i 為決策樹節點，意謂著在此節點下建立 m 個子節點。因此，以屬性 A_i 為決策樹節點所產生的子決策樹之熵值計算如下：

$$E(A_i) = \sum (n_k/n) * E(C_k)$$

其中 C_k 為物件集合 C 中其 A_i 屬性值相同的物件子集合 k

$E(C_k)$ 為物件子集合 C_k 之熵值

n 為物件集合 C 的總物件數

n_k 物件子集合 C_k 的物件數

資訊獲利 (Information Gain) 是指因選擇屬性 A_i 為決策樹節點而帶來熵值的改變程度，也就是原來物件集合的熵值與以 A_i 為決策樹節點的子決策樹之熵值間的差距，計算如下：

$$G(A_i) = E(C) - E(A_i)$$

Step5: 計算獲利比 (Gain-Ratio) 做為屬性選擇衡量方法，其計算如下：

$GR(A_i) = G(A_i) / IV(A_i)$ 其中 $G(A_i)$ 為 A_i 屬性對 C 的物件集合進行分割所得的資訊獲利 $IV(A_i)$ 為 A_i 屬性的資訊值 $= -\sum p_m \log_2 p_m$ ($m=1..k$) 其中 $p_m = (C \text{ 集合中屬於 } A_i \text{ 屬性值為 } a_m \text{ 的總物件數}) / (C \text{ 的總物件數})$

Step6: 選擇獲利比最大的候選屬性，將之當成節點 C 的分類屬性。

Step7: 在節點 C 下建立子節點 C_1, C_2, \dots, C_m (假設所選定的分類屬性有 m 個屬性值)，依據分類屬性值，將 C 中的所有物件分派至適當的子節點中。

Step8: 對每個子節點 C_i 當成目前節點 C，並分別由 Step2 繼續執行。

C4.5 修剪決策樹的標準是以預估錯誤率 (Predicted Error Rate) 的值作為判斷條件。錯誤比率來評估其他非學習資料的錯誤比率。假設被分類於某一子樹的所有學習資料個數為

N，其中有 E 個學習資料分類錯誤，而錯誤預估率就是由 E/N 來評估當有新的資料被測試時，利用此一條件判斷結果為不正常的機率值 [1]。

表 3 為 LCD 驅動 IC 封裝產業品質問題資料，設備、材料、方法及人員為主要預測變數：

一般在決策樹的每一個節點上使用信息增益度量來選擇測試屬鍵，假設 S 是 s 個資料樣本的集合。假設類別標籤屬性具有 m 個不同值，定義 m 個不同類

$C_i (i=1, \dots, m)$ 。設 S_i 是類 C_i 的樣本數。對於一個給定的樣本分類所需的期望資訊由下列式子算出：

(a). 計算預測變數的期望資訊與資訊獲利：

$$I(s_1, s_2, \dots, s_m) = -\sum p_i \log_2(p_i) \quad i=1, \dots, m$$

(1). 計算材料變數期望資訊與資訊獲利：

對於材料=" 膠本身過濃"

$$S_{11}=2 \quad S_{21}=3 \quad I(S_{11}, S_{21})=0.971$$

對於材料=" 膠本身氣泡"

$$S_{12}=4 \quad S_{22}=0 \quad I(S_{12}, S_{22})=0$$

對於材料=" Tape 本身內縮"

$$S_{13}=3 \quad S_{23}=2 \quad I(S_{13}, S_{23})=0.971$$

$$I(S_1, S_2) = I(9, 5) = 0.940$$

期望資訊 $E(\text{材料}) = (5/14) * I(S_{11}, S_{21}) + (4/14) * I(S_{12}, S_{22}) + (5/14) * I(S_{13}, S_{23}) = 0.694$

$$I(S_{11}, S_{21}) + (4/14) * I(S_{12}, S_{22}) + (5/14) * I(S_{13}, S_{23}) = 0.694$$

資訊獲利 $\text{Gain}(\text{材料}) = I(S_1, S_2) - E(\text{材料}) = 0.246$

$$-E(\text{材料}) = 0.246$$

(2). 計算設備變數期望資訊與資訊獲利：

對於設備=" 無"

$$S_{11}=3 \quad S_{21}=1 \quad I(S_{11}, S_{21})=0.811$$

對於設備=" Marking 壓力有異"

$$S_{12}=2 \quad S_{22}=2 \quad I(S_{12}, S_{22})=1$$

對於設備=" Curing 溫度有異"

$$S_{13}=4 \quad S_{23}=2 \quad I(S_{13}, S_{23})=0.918$$

期望資訊 $E(\text{設備}) = (4/14) * I(S_{11}, S_{21}) + (6/14) * I(S_{12}, S_{22}) + (4/14) * I(S_{13}, S_{23}) = 0.911$

$$I(S_{11}, S_{21}) + (4/14) * I(S_{12}, S_{22}) + (6/14) * I(S_{13}, S_{23}) = 0.911$$

資訊獲利 $\text{Gain}(\text{設備}) = I(S_1, S_2) - E(\text{設備}) = 0.940 - 0.911 = 0.029$

$$-E(\text{設備}) = 0.940 - 0.911 = 0.029$$

(3). 計算方法變數期望資訊與資訊獲利：

對於方法=" 溫度有異"

$$S_{11}=3 \quad S_{21}=4 \quad I(S_{11}, S_{21})=0.985$$

對於方法=" 無"

$$S_{12}=6 \quad S_{22}=1 \quad I(S_{12}, S_{22})=0.592$$

$$\begin{aligned} \text{期望資訊} E(\text{方法}) &= (7/14) * I(S_{11}, S_{21}) \\ &+ (7/14) * I(S_{12}, S_{22}) = 0.789 \\ \text{資訊獲利 Gain}(\text{方法}) &= I(S_1, S_2) \\ &- E(\text{方法}) = 0.151 \end{aligned}$$

(4). 計算人員變數期望資訊與資訊獲利：

$$\begin{aligned} \text{對於人員} = \text{無} \\ S_{11} = 6 \quad S_{21} = 2 \quad I(S_{11}, S_{21}) &= 0.811 \\ \text{對於人員} = \text{Marking 壓力有異} \\ S_{12} = 3 \quad S_{22} = 3 \quad I(S_{12}, S_{22}) &= 1 \\ \text{期望資訊} E(\text{人員}) &= (8/14) * I(S_{11}, S_{21}) \\ &+ (6/14) * I(S_{12}, S_{22}) = 0.892 \\ \text{資訊獲利 Gain}(\text{人員}) &= I(S_1, S_2) \\ &- E(\text{人員}) = 0.048 \end{aligned}$$

(b). 計算資訊獲利比 Gain-Rate(GR)：

$$\begin{aligned} \text{GR}(\text{材料}) \\ &= 0.246 / (- (5/14) * \log_2(5/14) - (4/14) * (\log_2(4/14) - 5/14 * \log_2(5/14))) = 0.156 \\ \text{GR}(\text{設備}) \\ &= 0.029 / (- (4/14) * \log_2(4/14) - (4/14) * (\log_2(4/14) - 6/14 * \log_2(6/14))) = 0.019 \\ \text{GR}(\text{方法}) &= 0.151 / (- (7/14) * \log_2(7/14) - (7/14) * (\log_2(7/14))) = 0.151 \\ \text{GR}(\text{人員}) &= 0.048 / (- (8/14) * \log_2(8/14) - (6/14) * (\log_2(6/14))) = 0.048 / 0.986 = 0.049 \end{aligned}$$

由於材料在屬性中有最高資訊獲利比，它被選作測試屬性，因而以此建立節點，其他則配合修剪演算或限制深度等作法達到決策樹分類的目的。

(2). 類神經網路演算法

類神經網路是由一組神經元節點(node)與具有權重的節點間連結(link)所構成的網路。神經網路的多層結構可分為輸入層(input layer)、隱藏層(hidden layer)以及輸出層(output layer)，而層與層間的神經元則以完全連結(fully connected)的方式向前連接著(forward connected)，其演算法如下所示[5]：

假設類神經網路為單一隱藏層，輸入層有*i*個單元，隱藏層有*j*個單元，則連接權重以 W_{ij} 表示，容忍誤差以 θ_j 表示

Step1: 設定網路參數，包括各層神經元數目，學習率1、訓練次數、容忍誤差 θ_j 。

Step2: 設定網路的初始權值 W_{ij} ；乃隨機產生之亂數值。

Step3: 將輸入向量及輸出向量依神經元轉換函數之值域，分別將其正規化(normalize)。神經元轉換函數為 $f(x) = 1 / (1 + e^{-x})$

Step4: 計算隱藏層及輸出層之輸出。

$$\text{第} j \text{單元輸入向量為 } I_j = \sum W_{ij} * O_i + \theta_j$$

$$\text{第} j \text{單元輸出向量 } O_j = 1 / (1 + e^{-I_j})$$

Step5: 計算輸出層及隱藏層之誤差。

$$\text{輸出層誤差為 } Err_j = (O_j(1-O_j)(T_j-O_j))$$

$$\text{隱藏層誤差為 } Err_j = (O_j(1-O_j) \sum Err_k * W_{kj})$$

其中 W_{kj} 是由下一較高層中單元*k*到單元*j*的連接權重，而 Err_k 而是單元*k*的誤差，而是*j*基於給定的已知類別標籤的真正輸出。

Step6: 計算網路權值修正量，並修正之。

$$\text{網路權值修正量 } (W_{ij} = W_{ij} + 1 * Err_j * O_i)$$

$$\text{誤差修正量 } (\theta_j = \theta_j + 1 * Err_j)$$

Step7: 跳回 Step3，重複計算直到誤差量小於容忍誤差或指定的訓練次數為止。

圖 4 為一多層前饋類神經網路範例，假設學習率為 0.9。

(a). 該網路的初始權值和偏差值以及第一個訓練樣本 $X = (1, 0, 1)$ ，如表 4 所示。即學習率 1、容忍誤差 θ_j ($j = 4, 5, 6$)、初始權值 W_{ij} ($i = 1, 2, 3, 4, 5; j = 4, 5, 6$) 乃隨機產生之亂數值。

(b). 給定第一個訓練樣本 X ，首先將樣本提給網路，並計算每個單元的淨輸入和輸出這些值表示在表 5 中。依轉換函數為 $f(x) = 1 / (1 + e^{-x})$ 計算隱藏層及輸出層之輸出。

第*j*單元輸入向量為 $I_j = \sum W_{ij} * O_i + \theta_j$ ，第*j*單元輸出向量 $O_j = 1 / (1 + e^{-I_j})$

$$I_4 = \sum W_{i4} * O_i + \theta_4 = -0.7$$

$$O_4 = 1 / (1 + e^{0.7}) = 0.332$$

$$I_5 = \sum W_{i5} * O_i + \theta_5 = 0.1$$

$$O_5 = 1 / (1 + e^{-0.1}) = 0.525$$

$$I_6 = -0.105$$

$$O_6 = 1 / (1 + e^{0.105}) = 0.474$$

(c). 接著計算每個單元的誤差，並向後傳播。此誤差值在表 6 中。

輸出層之誤差為 $Err_j = O_j(1-O_j)(T_j-O_j)$ ，隱藏層之誤差為 $Err_j = (O_j(1-O_j) \sum Err_k * W_{kj})$

其中 W_{kj} 是由下一較高層中單元*k*到單元*j*的連接權重，而 Err_k 而是單元*k*的誤差，而是*j*基於給定的已知類別標籤的真正

輸出

$$\text{Err}_6 = (O_6(1-O_6)(T_6-O_6)) = 0.1311$$

$$\text{Err}_5 = (O_5(1-O_5)\sum \text{Err}_6 * W_{65}) = -0.0065$$

$$\text{Err}_4 = (O_4(1-O_4)\sum \text{Err}_6 * W_{64}) = -0.0087$$

- (d). 而更新後的權重和偏差在表 7 中。誤差修正量($\theta_j = \theta_j + L * \text{Err}_j$)，網路權值修正量($W_{ij} = W_{ij} + L * \text{Err}_j * O_i$)

$$\theta_6 = \theta_6 + L * \text{Err}_6 = 0.1 + 0.9 * (0.1311) = 0.218$$

$$\theta_5 = \theta_5 + L * \text{Err}_5 = 0.2 + 0.9 * (-0.0065) = 0.194$$

$$\theta_4 = \theta_4 + L * \text{Err}_4 = -0.408$$

$$W_{46} = W_{46} + L * \text{Err}_6 * O_4 = -0.261$$

$$W_{56} = -0.2 + 0.9 * 0.1311 * 0.525 = -0.138$$

$$W_{14} = 0.2 + 0.9 * (-0.0087) * 1 = 0.192$$

$$W_{15} = -0.3 + 0.9 * (-0.0065) * 1 = -0.306$$

$$W_{24} = 0.4 + 0.9 * (-0.0087) * 0 = 0.4$$

$$W_{25} = 0.1 + 0.9 * (-0.0065) * 0 = 0.1$$

$$W_{34} = -0.5 + 0.9 * (-0.0087) * 1 = -0.508$$

$$W_{35} = 0.2 + 0.9 * (-0.0065) * 1 = 0.194$$

- (e). 重複如此步驟直至學習結束為止。

5. 實作

5.1. 系統環境

產品品質問題分析系統的實作環境，分為以下幾個部份：

1. MES Server: 記錄每日生產交易資料。
2. ERP server: 記錄每日企業各項營運流程交易資料。
3. DataMart Server: 整合 ERP 以及 MES 關於分析系統所需資料綱要及資料。
4. Data Mining Server: 作為資料探勘系統運作環境。
5. OLAP Server: 從 DataMart Server 擷取資料，作 OLAP 分析。
6. Client PC: 以視覺化呈現方式，供使用者操作分析用。

而品質問題分析系統整體環境架構圖如圖 5 所示：

5.2. 資料探勘模式實作

5.2.1. 決策樹方法

1. 即由已建立的資料倉儲中選取品質問題的資料，並定義出屬性變數欄及類別變數欄，屬

性變數為人員、設備、材料、方法，類別變數為品質問題。

2. 設定決策樹演算法的各項參數，如是否修剪決策樹、決策樹最高深度、訓練組及測試組比例等，取得個案公司 2002 年 1 月至 2003 年 12 月計 24 個月資料，資料總筆數 9752 筆，以亂數選取方式將訓練組及測試組以 3:1 比例作分類分析。
3. 決策樹演算法針對先前所設定的參數，其實際與推估列出矩陣如下，其訓練組的總筆數為 7314 筆，訓練組分類正確筆數計 5851 筆，因此正確率高達 80%，如圖 6 所示。
4. 最後依決策樹演算法針對先前所設定的參數以亂數選取測試組的方式進行預測推估驗證，測試組總筆數為 2438 筆，實際與推估列出矩陣如下，其測試組的分類正確比數計 1682 筆，正確率高達 69%，如圖 7 所示。

5.2.2. 類神經網路方法

1. 將品質問題的資料建立起單一表格資料庫，定義出屬性變數欄及類別變數欄。
2. 設定類神經網路演算法的各項參數，如學習率、隱藏層層數、訓練組及測試組比例等，個案公司 2002 年 1 月至 2003 年 12 月計 24 個月資料，資料總筆數 9752 筆，以亂數選取方式將訓練組及測試組以 3:1 比例作類神經網路分類分析。
3. 類神經網路演算法針對先前所設定的參數，其訓練組的模擬學習的正確率，如圖 8 所示跑出實際與學習推估分類矩陣圖，資料總筆數 9752 筆，其訓練組的總筆數為 7314 筆，訓練組分類正確筆數計 4535 筆，正確率高達 62%。
4. 最後依類神經網路演算法針對先前所設定的參數以亂數選取測試組的方式進行預測推估驗證，其實際與推估列出矩陣如下，測試組總筆數為 2438 筆，其測試組的分類正確比數計 1292 筆，正確率高達 53%，如圖 9 所示類神經網路實際與預測分類推估矩陣圖。

5.3. 效益分析

根據所建置的資料庫，決策樹及類神經網路的方法，所達成的效益有下列幾項：

1. 降低內引腳不良顆數：

以 2004 年 1 至 6 月系統使用前內引腳不良顆數如表 8 所示，而分別以決策樹及類神經網路的方法在 2004 年 7 至 12 月系統使用後內引腳不良顆數如表 8 所示，決策樹方法總平均改善率為 10.4%；同樣以類神經網路方法總平均改善率為 2.95%，而內引腳不良顆數改善前後統計曲線圖如圖 10 所示。

2. 降低樹脂問題不良顆數：

以 2004 年 1 至 6 月系統使用前樹脂問題不良顆數如表 9 所示，而分別決策樹及類神經網路的方法在 2004 年 7 至 12 月系統使用後樹脂問題不良顆數如表 9 所示，以決策樹總平均改善率為 9.2%；同樣以類神經網路方法總平均改善率為 4.4%，而樹脂問題不良顆數改善前後統計曲線圖如圖 11 所示。

3. 降低晶片問題不良顆數：

以 2004 年 1 至 6 月系統使用前晶片問題不良顆數如表 10 所示，而決策樹及類神經網路的方法在 2004 年 7 至 12 月系統使用後晶片問題不良顆數如表 10 所示，決策樹總平均改善率為 11.6%；同樣以類神經網路總平均改善率為 4.6%，而晶片問題不良顆數改善前後統計曲線圖如圖 12 所示。

針對個案公司以資料探勘決策樹演算法及類神經網路演算法針對品質問題資料進行分類分析，資料總筆數為 9752 筆，以隨機取樣方式將資料分為訓練組及測試組，其比例為 3:1，訓練組筆數 7314 筆，測試組 2438 筆，以決策樹演算法作分類分析其訓練組正確筆數 5851，正確率達 80%，測試組正確筆數 1682，正確率達 69%；而類神經網路演算法作分類分析，其訓練組正確筆數 4535，正確率達 62%，測試組正確筆數 1292，正確率達 53%；另外分別以決策樹及類神經網路的方法在

2004 年 7 至 12 月系統使用後，決策樹在內引腳不良顆數總平均改善率為 10.4%，類神經網路方法總平均改善率為 2.95%；決策樹在樹脂問題不良顆數總平均改善率為 9.2%，類神經網路方法總平均改善率為 4.4%；決策樹在晶片問題不良顆數總平均改善率為 11.6%，類神經網路方法總平均改善率為 4.6%，由此可知決策樹演算法較適用於品質問題分類分析，如表 11 所示。

6. 結論

本研究的總結主要提出適用於 LCD 驅動 IC 封裝廠產品品質問題分析模式並以案例公司研究，以 LCD 驅動 IC 封裝廠來說，為提升產品良率及加速了解產品不良原因及找出對策，建置一產品品質問題分析模式，可使工作更有效率，更快取得真正所需的資訊，並提高客戶對資訊系統的滿意度。

在品質問題分析的效益上，決策樹在內引腳不良顆數總平均改善率為 10.4%，樹脂問題不良顆數總平均改善率為 9.2%，晶片問題不良顆數總平均改善率為 11.6%；類神經網路方法在內引腳不良顆數總平均改善率為 2.95%，樹脂問題不良顆數總平均改善率為 4.4%，晶片問題不良顆數總平均改善率為 4.6%，決策樹方法較類神經網路方法適用於品質問題分類分析。

此方法可適用於電腦組裝業、電子零件製造、半導體封裝製造業品質問題分析。