二、研究計畫內容（以10頁為限）：

(一).摘要

在學習程式設計的過程中，同學在撰寫程式時通常都會互相參考。然而有些同學會將別人的程式碼做一些小修改，比方說變數取代等，就當成自己的作業繳交，甚至有些同學認為老師不會仔細比對程式碼，就直接拿別人的程式碼當成是自己的來繳交。因此老師無法容易分辨同學是否認真撰寫自己的程式。

本研究的目的，是要開發出一套程式碼比對系統。此系統會先根據程式碼定義一些特徵值，並且這些特徵值經過正規化處理。先從過去歷年來同學所繳交的作業，當成是訓練樣本，經過分類器的處理後可以得到一個模型。未來同學繳交作業後，經過這個模型的處理，可以很容易將同學所繳交的作業進行分類。老師可以藉此分類結果，評估同學之間是否有相互抄襲作業。

(二). 研究動機與研究問題

資訊相關科系的同學，最主要的一門課程就是程式設計。學習程式設計沒有捷徑，只有不斷的練習撰寫程式碼，才會愈來愈進步。因此老師在授課過程中，會出非常多的程式設計題目要同學回家練習，只要同學可以按照老師的進度學習程式設計，一定可以精進程式設計的技術。

然而，並非每個同學對於程式設計都有興趣，甚至有些同學排斥程式設計這門課。由於大部份程式設計課程都是必修，因此就算不喜歡，還是必須要修這門課。所以當老師出作業時，同學之間都會相互討論程式碼該如何撰寫。如果同學討論完後各自撰寫自己的程式，則寫出來的程式碼雖然邏輯結構有些類似，但是變數名稱、排版結構、撰寫風格等，必定會有所不同，因此很容易可以分辨出程式是否自行撰寫的。可是就會有一些同學，直接跟其他同學複製程式碼當成是自己的。由於是複製過來的，不見得看得懂程式碼，所以不敢亂改程式碼，只敢做簡單的變數名稱取代，讓程式碼看起來跟別人的不一樣。或者是將程式碼做一些不同的排版或

是增加/移除註解，來偽裝是自己撰寫的。如果老師沒有將這些投機的同學糾正，則只會造成這些同學愈來愈投機，無法好好地學習程式設計這門課。

要將程式碼抄襲的同學抓出來，是一件非常耗時耗工的事。老師必須看完所有的程式碼，並將程式碼類似的同學歸成一類，再仔細判斷是否抄襲。為了解決這個問題，本研究要利用機器學習分類器，透過分類器的判斷，自動將程式碼類似的同學歸類，讓老師可以很快就找出來哪些同學是相互抄襲的，再藉由扣分或警告同學的方式，讓同學願意自己撰寫程式碼，以便能夠提高學生程式設計的能力。

在進行分類之前，必須先把程式碼的特徵定義出來。在本研究中會定義一些程式碼的特徵，比方說程式碼長度、程式碼行數、程式敘述個數、變數個數、變數使用的次數、條件判斷式個數、迴圈個數、副程式個數…等等。當進行完程式碼的讀取掃描後，就可以分別得到這些特徵的相關數值。由於這些數值的範圍大小不一致，為了方便後續分類器的處理，將這些特徵值進行正規化，也就是讓所有的特徵值都介於0到1之間的數值，如此一來，才不會因為特徵值的範圍而影響到分類器的精確度。

定義完特徵值後，必須先取得訓練資料。由於指導老師已經教授多年程式設計相關的課程，也都有把歷年來同學所繳交的作業保留下來，因此可以取得大量的訓練資料。先利用特徵值擷取程式取得程式碼的特徵值，再利用人工的方法進行分類。雖然這部份會花相當多的時間，然而如果能取得更多的訓練資料，對於將來分類器的準確率會有相當大的幫助。

有了足夠的訓練資料後，接下來就是要決定使用的分類器。目前幾個較常用的分類器包括OneR、Naive Bayes、Decision Tree、KNN、Logistic Regression…等等。將訓練資料經過Cross-validation的方法分別進行準確率計算，從中找到一個準確率最高的分類器，並利用此分類器建立模型，此模型可以用來將未來同學繳交作業進行分類，經過分類之後，就可以明確找出哪些同學程式碼有疑似抄襲得嫌疑。

(三).文獻回顧與探討

李昶宏等人所提出的『利用序列比對演算法辨識抄襲之 C 程式作業』，比對演算法是利用改良過後的 DNA 比對演算法來進行比對。如果單純地利用程式間的 edit distance 來估算，可能會忽略許多問題，導致誤判產生。因此他們決定採用區域性比對來求得相似度，即 local alignment by dynamic programming。採用簡單的編碼技巧，可把所需記憶體縮小成四分之一。利用“邊算邊紀錄”的方法讓程式在時間上不需要重算。

林世唐等人提出『學生程式碼相似度之研究—以抄襲偵測之應用為例』，是以IDF為主的新方法，幫助他們找出發生頻率低的相似片段組，視為較有抄襲可能性之片段。並用開放式(open book)的一次考試和一次作業程式來作驗證。

黃福助等人提出『利用多個相似度演算法實作程式碼抄襲系統』，利用三種不同的方法計算相似度，藉此能夠客觀地找出抄襲作業。研究的方法分別是文字分析方法、結構分析方法及屬性分析方法。在文字分析方法中，建立文字處理流程並應用 winnowing 演算法計算相似度；在結構分析方法中，增強 F(p)演算法用以轉換類別結構變成文字，再使用 winnowing演算法比對文字；在屬性分析方法中，提出變數分析方法比較兩個程式變數的相似度，以及利用統計方法比較類別相似度，而透過測試得到提出的系統比較能夠有效找到抄襲。

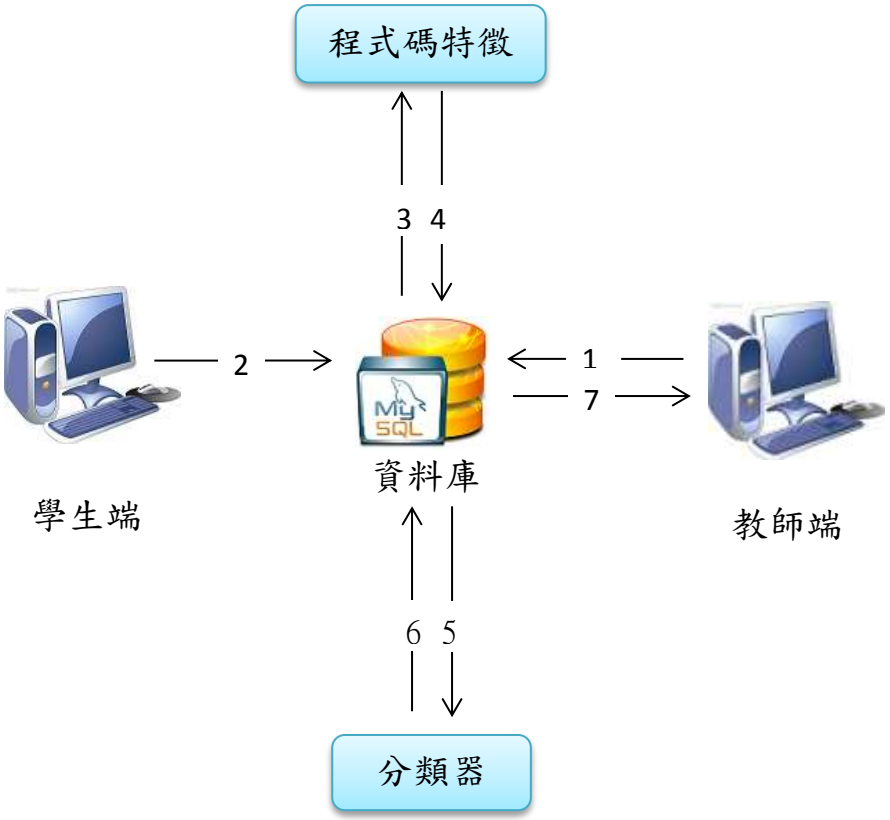
以上的文獻都是針對演算法的研究和特定的程式。而本研究的方法與上述文獻不同的是，本系統是採用機器學習的方法，利用訓練資料讓準確率可以提高。同時，老師也能新增自己想新增的特徵，讓程式碼比對的條件更有彈性。同時因為分析是透過老師給的特徵去分析，所以在程式上沒有限制。

(四).研究方法及步驟

4.1 研究方法

4.1.1 系統架構

本系統的主要兩個功能分別是程式碼特徵選擇以及分類器的分析。老師開啟作業後，學生將作業上傳，利用程式碼特徵模組擷取出程式碼的特徵，並將特徵存進資料庫中以供分類器分類使用，分類的結果，即為程式碼抄襲的判斷。系統架構如圖一所示。其運作步驟說明如下。



圖一 系統架構

1. 教師新增繳交作業並指定程式碼判斷特徵，將相關輸入的資料傳到資料庫中。
2. 學生登入後，選擇教師在資料庫中開啟的作業上傳。
3. 程式碼特徵模組分析學生上傳之程式碼，並進行特徵分析。
4. 將擷取的特徵結果存入到資料庫中。
5. 將特徵當成分類器的輸入，進行程式碼相似度的比對。
6. 將分類器分類完成的結果存入資料庫。
7. 老師可以從資料庫中找出比對的結果。

4.1.2 特徵選擇

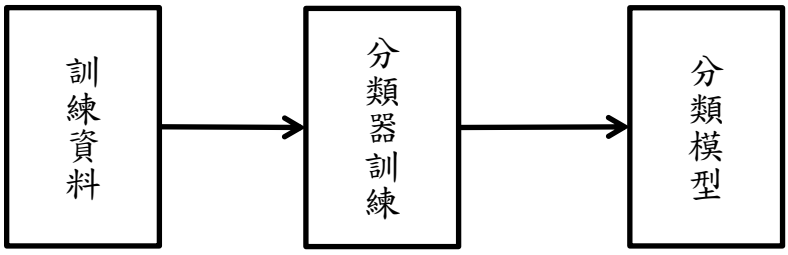
隨著每一門課教學的程式語言不同，而產生了不一樣的撰寫方法。老師可以新增其特徵並利用特徵選擇功能，選定目前想擷取的特徵數值，每一次的作業都可以擷取不同的特徵，因此可以不限定任何程式語言、任何作業的使用，大幅增加老師使用上的方便性。

當特徵選擇模組執行時，先掃描同學所上傳的作業，並利用老師一開始選定的特徵，在程式碼中擷取老師設定特徵的相關數值，取得所有特徵的相關數值後，即可將這些相關數值放入分類器中進行分類。特徵選擇功能優點如下。

1. 不限定程式語言。
2. 可以隨著不同的語言及老師需求增加不同的特徵。
3. 每次使用時可選定不同的特徵。
4. 新增特徵時可選擇計算的種類，可分為此特徵出現次數以及計算宣告的變數。

4.1.3 分類器

分類器是利用大量已經修過課的同學作業為訓練樣本，來產生分類器的模型，再利用資料庫中所記錄的特徵進行分類，讓老師得知那些作業的相似度較高。分類器運作的架構如圖二所示，並說明如下。

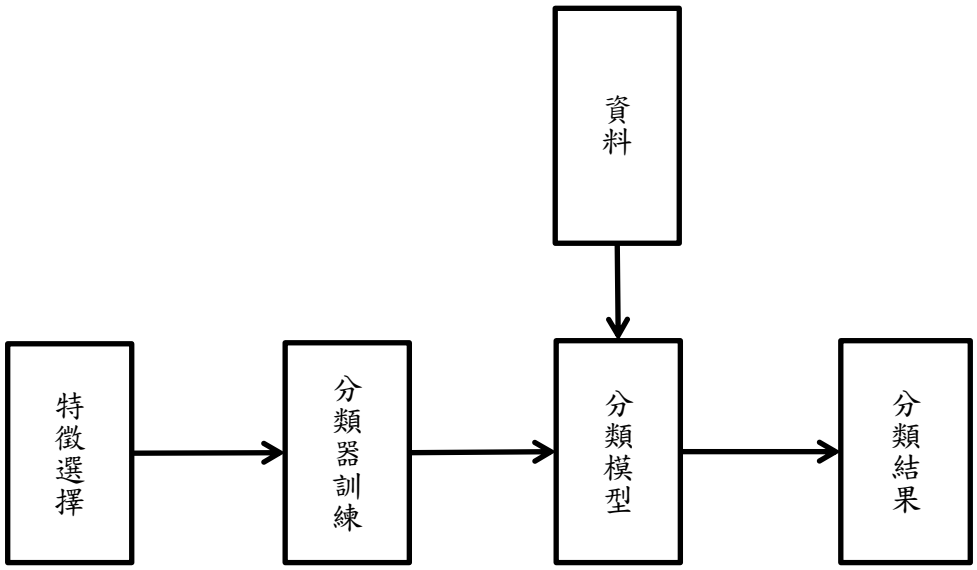


圖二 分類器運作架構

1. 使用分類器分類資料前需先有一群訓練資料，當成分類器的輸入。
2. 將訓練資料經過分類器訓練。

3. 產生分類器的分類模型。

因此，結合特徵的選擇、分類器訓練以及將同學作業進行分類，整體的運作流程如圖三所示。

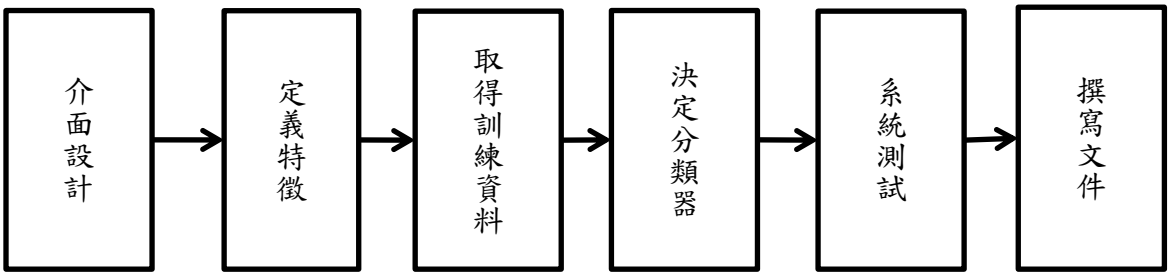


圖三 資料分類流程

4.2 研究步驟

本研究的進行步驟說明如下，研究流程如圖四所示。

1. 老師端開啟作業及查詢結果介面設計，學生端上傳作業介面設計。
2. 定義程式碼的特徵，並將特徵值正規化。
3. 使用歷年學生繳交的作業，以人工方式進行分類，取得大量的訓練資料。
4. 比對資料庫中的訓練資料及測試資料，決定一個最適合的分類器。
5. 進行系統測試測試。
6. 完成文件之撰寫。



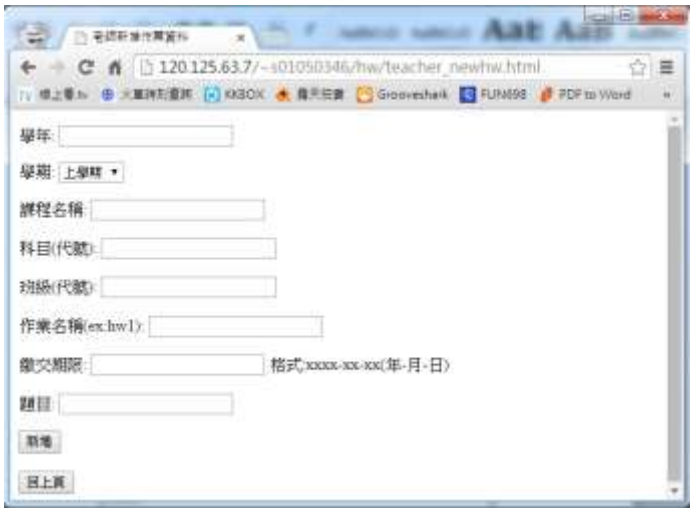
圖四 研究步驟流程

(五). 預期結果

當老師進入本系統，並完成帳號密碼驗證後，會出現要選擇查詢作業分類結果或是新增作業，如圖五所示。

圖五 選擇查詢結果或新增作業

當選擇【新增作業】後，老師即可輸入作業相關設定，如圖六所示。

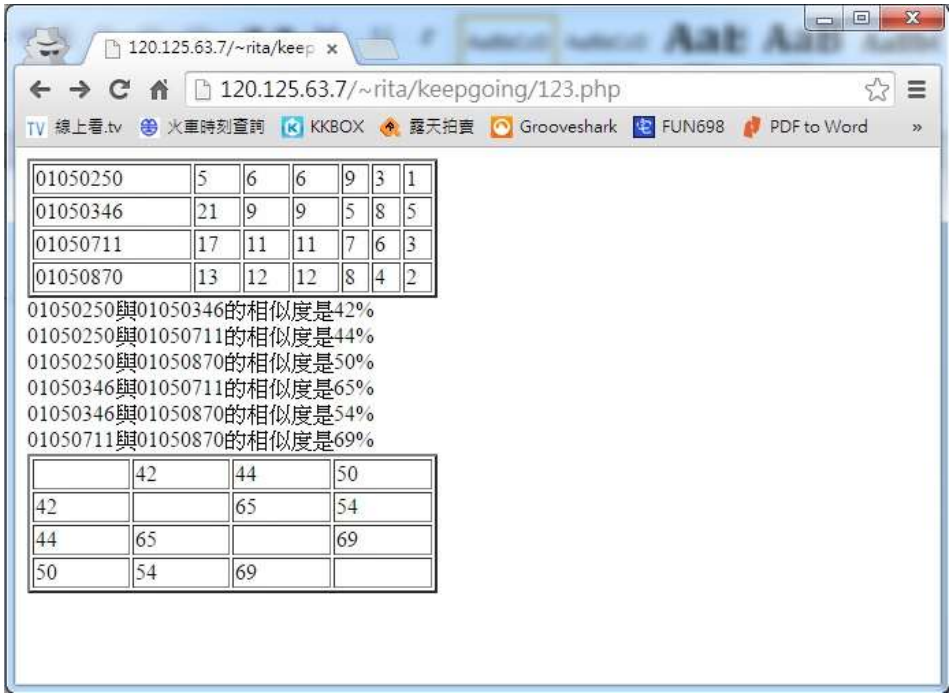


圖六 新增作業畫面

學生登入本系統，並經過帳號密碼驗證後，即進入選擇作業及作業上傳頁面，如圖七所示。

圖七 學生上傳作業畫面

如果全部學生已經完成上傳作業後，老師可登入點選【查詢】，系統會依選擇的作業，根據選擇的特徵進行相似度比對,比對結果的數值如圖八所示。最後顯示出分類結果，如圖九所示。



01050250	5	6	6	9	3	1
01050346	21	9	9	5	8	5
01050711	17	11	11	7	6	3
01050870	13	12	12	8	4	2

01050250與01050346的相似度是42%
01050250與01050711的相似度是44%
01050250與01050870的相似度是50%
01050346與01050711的相似度是65%
01050346與01050870的相似度是54%
01050711與01050870的相似度是69%

	42	44	50
42		65	54
44	65		69
50	54	69	

圖八 相似度比對結果

圖九 分類結果

（六）.參考文獻

[1] 李昶宏,利用序列比對演算法辨識抄襲之 C 程式作業，國立暨南國際大學資訊工程學系碩士論文，2004 年。

[2] 林世唐，學生程式碼相似度之研究—以抄襲偵測之應用為例，淡江大學資訊管理學系碩士論文，2004 年。

[3] 黃福助，利用多個相似度演算法實作程式碼抄襲系統，臺北科技大學資訊工程系研究所論文，2013 年。
[4] 潘忠建，程式碼相似度比對應用於侵權鑑定之研究，國防大學中正理工學院資訊科學研究所論文，2007 年。
[5] 孫士烘，使用 Edit-Distance 比對 C 程式碼相似度，中原大學資訊工程學系碩士學位論文，2002 年。
[6] 李書雅，應用資料探勘技術分析學生程式碼，國立雲林科技大學資訊管理系碩士班碩士論文，2009 年。
[7] 游景翔，混合式電腦程式抄襲偵測，國立臺灣科技大學資訊工程學系碩士論文，2006 年。
[8] 洪士軒，軟體設計相似度的計算，東海大學資訊工程與科學研究所碩士論文，2003年。

(七).需要指導教授指導內容

1. 專題研究相關方向、內容、問題。
2. 程式寫作相關想法、問題、演算法。
3. 介面架設相關觀念、指令操作、問題。
4. 正式文件撰寫。