中華民國第 57 屆中小學科學展覽會
作品說明書

高級中等學校組　電腦與資訊學科

052507

基於使用者輸入行為之監督式身分認證模型

學校名稱：國立政治大學附屬高級中學

作者：	指導老師：
高二 林祥源 高二 周敬恩	李建隆

關鍵詞：打字輸入行為、身份辨識、機器學習

摘　要

在網路科技盛行的時代，帳號及密碼的設置為必要的程序，因此密碼管的安全性成為重要的課題。現行主要的認證機制需要使用者自行記密碼，而便利性高的生物辨識方法則大多成本過高或不夠準確，因此，本專題研究希望提出一個基於使用者鍵盤輸入行為的辨識方法解決此問題。首先，我們分析了使用者的輸入行為，提出 11 種特徵。再來，我們用支持向量機進行分類。支持向量機在超空間中求出超平面，將資料區隔，算出適合的權重並求得最佳參數，以得到最佳效果。在包含 230 位不同年齡層受試者的實驗中，本專題研究所提出的方法優於前人所提出的方法。我們也在較小的基準化資料集中進行實驗，精準度亦高。同時不需要特殊硬體，得以廣泛應用於現實生活中。

壹、研究動機

今年（西元 2016 年）六月全球最大的社群網站 Facebook 創辦人 Mark Zuckerberg 的社交網站帳號傳出被盜用的消息。大家赫然發現，這樣一個網路巨擘所使用的密碼，居然是個安全性極低的常見密碼。

在這個網際網路盛行的時代，每個人都有數個軟體帳號，如果想設置安全性比較高的密碼，必須包含大小寫英文字母、數字、符號，且至少有 12 碼（Morgan 2015）。但記住複雜的密碼並不容易，因此使用者多半會讓所有帳號共用一組密碼；然而，一旦共用的密碼被破解，所有帳戶就可能同時被惡意盜用。

針對身份認證，目前最安全的做法是使用「在線密碼管理器」或者是「兩階段認證法」，但這將會使登入任何帳戶都變得非常繁瑣且耗時，因此，生物辨識系統被認為在未來會取代密碼管理，成為身份認證的主要方式。雖然該技術在最近幾年有長足的發展，但因準確度、實用性以及成本等方面的考量，所以還沒辦法廣泛的落實至日常生活中。例如：虹膜辨識系統所需儀器成本太高；臉部辨識系統會受使用者的化妝與髮型影響；聲音辨識資料容易被側錄盜取；相對廣泛接受的指紋辨識亦存在高硬體成本以及低準確度的缺點。因此，如何另闢蹊徑找到更為實用的方法，便成為相當具研究價值的問題。

貳、研究目的

本專題研究的目的，在於建立以機器學習技術為基礎的打字特徵辨識模型，提出成本低、配置簡易，同時具有高精準度的辨識方法，使得打字特徵辨識得以應用於生活中。

参、相關研究

Gaines 等人利用特殊的感應裝置裝在鍵盤下方來測量受測者按鍵與按鍵之間的打字間隔時間，並記錄平均以做比對（Gaines, Lisowski, Press, & Shapiro, 1980）。雖然此方法在實驗中的入侵成功率（imposter pass rate）為零，誤警率（false alarm rate）亦僅 4%，但其受測者僅為六位專業秘書，無法有效的證明方法的有效性與全面性。Saevanee 與 Bhatarakosol 利用了 Synaptic Touchpad（一種原理與觸控面板相似，有壓力感測功能的特殊鍵盤）進行使用者辨識，並於共十位受測者參與的實驗中達到了 90%的辨識成功率（Saevanee & Bhatarakosol, 2008）。Loy 等人透過壓力感測的方式來辨識使用者，他們將鍵盤連接上 BNC 2110 shielded connector block（一種透過抗雜訊式外殼確保量測完整性的儀器）以及可記錄壓力的感測器來進行辨識（Loy, Lai, & Lim, 2005）。上述方法的準確度雖高，但都需要使用特殊硬體，如壓力感測或利用相機記錄受測者按壓鍵盤的確切位置，因此在日常生活中較難普及。再者，由於硬體設備的限制，前人實驗中大部分只有為數非常少且不足以成為有效樣本的受測者，像是 Gaines 等人（Gaines et al., 1980）只收集六位專業秘書的資料，Saevanee 與 Bhatarakosol 的實驗中亦只有十位受測者（Saevanee & Bhatarakosol, 2008）。因此，我們提出一個新的辨識方法，只需要藉由受試者自己的鍵盤搜集資料，無須昂貴或複雜的硬體即可達成目的，適用於各式各樣的電腦與設備，比起前人的方法成本更低、可行性更高。

肆、研究設備及器材

本專題所使用的設備僅需要一台電腦，其鍵盤為核心的器材，電腦使用的軟體包括 Python、文字編輯器（Brackets）以及由國立台灣大學資訊工程學系機器學習實驗室所推出的 libsvm 工具。

伍、研究方法

一、問題定義

首先，我們在本章節定義此專題研究所使用的符號，並利用符號與數學式正式定義欲解決的問題。

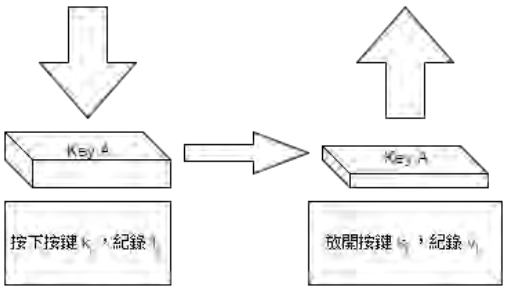
對於每個打字者 $u \in U$，打字紀錄集合 $X^u = \{x_i^u\}$ 為作為訓練資料的打字紀錄。每筆打字紀錄 $x_i^u = \{(k_1, t_1, v_1), (k_2, t_2, v_2), \cdots, (k_{N_i}, t_{N_i}, v_{N_i})\}$，$k$ 為按下的按鍵的代號（ASCII Code），k_j 代表第 j 個打的按鍵，t_j 代表按下按鍵的時間，v_j 代表放開按鍵的時間。利用上述符號，我們在此正式的定義問題：對於任一作為測試資料的打字紀錄集合 $X' = \{x_i'\}$，我們期望提出有效的方法預測出每筆測試打字資料 $x_i' \in X'$ 的打字者 $u_i' \in U$。

二、打字者輸入行為分析

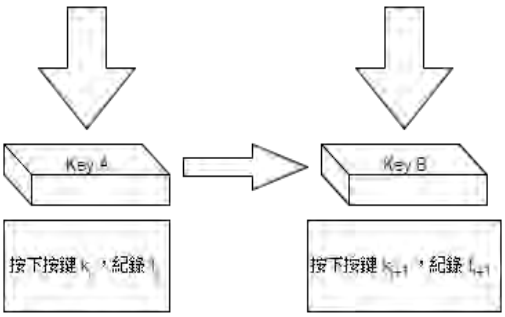
為了找出能有效區隔打字者的行為特徵，在此章節，我們對不同使用者的輸入行為進行了詳盡的分析。

（一）按壓時間以及按鍵間隔時間

我們分析使用者對於按鍵的按壓時間與間隔時間上的行為，並定義「按鍵按壓時間（D）」為 $D_j = v_j - t_j$，與「按鍵間隔時間（I）」為$I_j = t_{j+1} - t_j$。

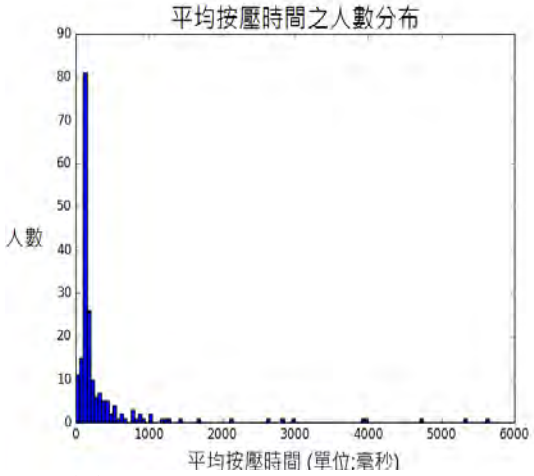


圖（1）：按鍵按壓時間示意圖

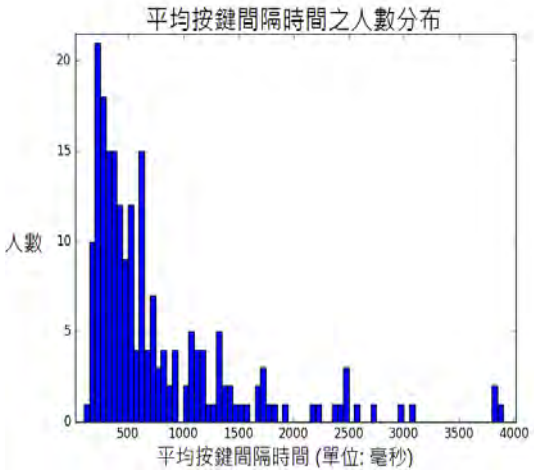


圖（2）：按鍵間隔時間示意圖

圖（3）與圖（4）記錄了不同使用者在打字過程中的平均按壓時間與平均間隔時間，我們期望能夠觀察這兩項行為，判斷其是否能區隔與分辨出使用者的身份。



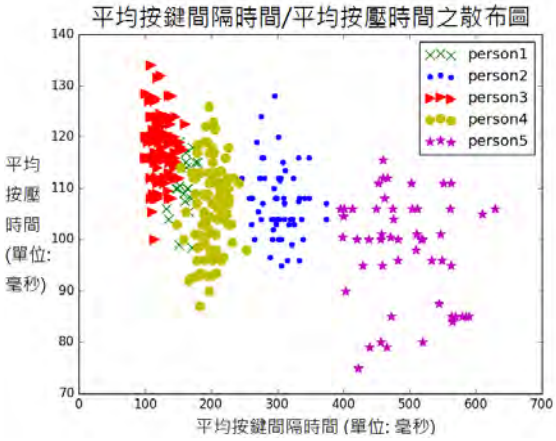
圖（3）：按鍵按壓時間的平均值直方圖



圖（4）：按鍵間隔時間的平均值直方圖

按壓時間在打字行為相關研究中是相當重要的數據。觀察圖（3）的結果，打字者的平均按壓時間大多集中在 150 至 200 毫秒之間，此段打字者無法以按壓時間區隔；但按壓時間為 200 毫秒以上的打字者，分布的分散程度相當高。因此，對於按壓時間較長的打字者，按壓時間有相對顯著的辨別效果。圖（4）的結果也有類似的趨勢，間隔時間較長的打字者較容易被分辨。

按鍵間隔時間和按壓時間是打字行為的核心特徵之一。它們是探討時間上的因素，合在一起時可以分析打字者的連續動作，而非只是記錄單獨動作。按鍵到按鍵之間的轉折，可以透露打字者對於鍵盤的配置熟悉的程度，對於較複雜的特徵（時間和空間合併）有相當高的關聯性。因此我們將不同人的平均按鍵間隔時間與平均按壓時間製作成散布圖（圖(5)），觀察是否能區分不同的打字者。



圖（5）：每個打字者按鍵間隔時間平均與按壓時間平均的相關性散布圖

從圖(5)中，可明顯看出打字者能被按鍵間隔時間與按壓時間有效區分。此外，按鍵間隔時間較按壓時間更能區隔出使用者，而且變異相當小，光從肉眼便可以輕易區分並指出這五位不同打字者的資料。我們也發現大部分打字者的資料分布都十分密集，證明了打字行為表現的獨特性。由於按鍵間隔時間為連續鍵擊間隔時間，所以數值愈低代表打字速度愈快。因此從此圖也可推斷打字速度愈快，變異就愈小；變異愈穩定，數據也相對更具代表性。

（二）進行刪字修正時的點擊習慣

在打字的過程中，常常會發生打錯字的情況，而每個人在打錯字時的修正行為可能會有顯著的不同。例如：有些人會一口氣刪掉整個單詞或句子，有些人只刪到錯誤的地方，有些人則用游標移到錯誤的位置，有些人按刪除鍵會分成多下按，有些人則會長按刪除鍵。

因此，我們分析以下三項數據（圖(6)至圖(9)）：第一項為按壓刪除鍵（Backspace，鍵盤上的代號（ASCII Code）為 8 號；參見附錄(1)）的次數（C_j）；我們將按壓刪除鍵的次數定義如下，以刪除鍵次數稱之：

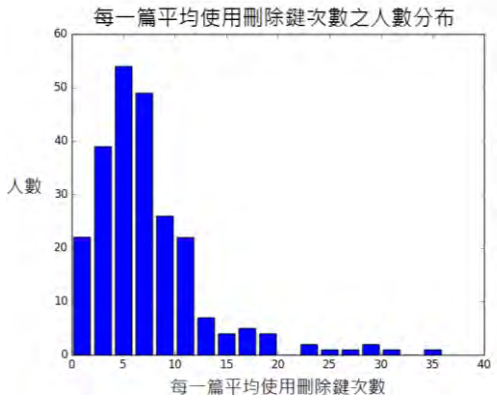
$C_j = k_j, when\ k = 8$（當 k，也就是按鍵代號，等於 8 的時候）。

第二項為連續按壓刪除鍵時，鍵與鍵之間的時間（I_j）；我們將其定義如下，以刪除鍵按鍵間隔時間稱之：

$I_j = t_{j+1} - v_j, when\ k = 8$。

第三項為刪除鍵從按下到放的時間（D_j）。我們將其定義如下，以刪除鍵按壓時間稱之：

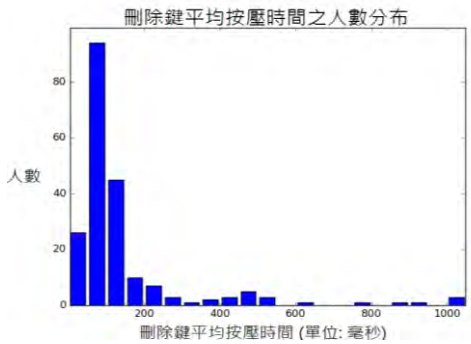
$D_j = v_j - t_j, when\ k = 8$。



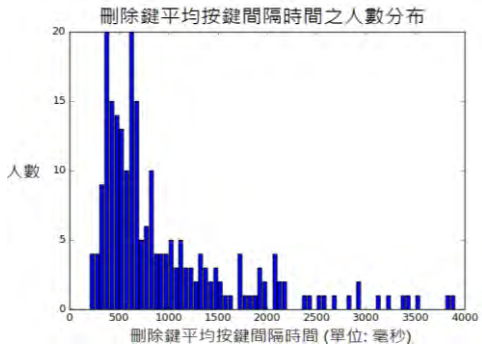
圖（6）：每一篇文章平均使用刪除鍵次數直方圖

圖（6）以大約 20 個英文單字的句子作為基準，計算打字者平均使用刪除鍵的次

數。我們認為每個人刪除錯字的行為不同，連帶會影響刪除鍵所按的次數，如：有的人習慣把錯字用滑鼠選取起來後按一次刪除鍵，有的人則習慣一個一個刪掉打錯的字；前者平均按刪除鍵的次數會趨向圖中左方，而後者則會在上圖中趨向右方。

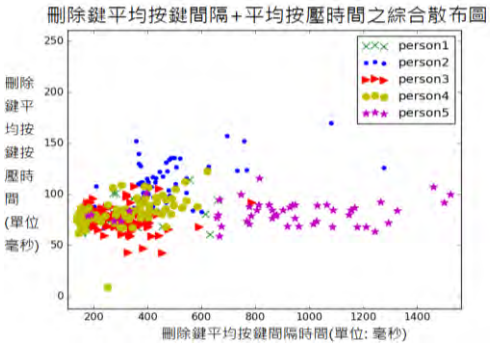


圖（7）：刪除鍵刪字時按壓時間直方圖



圖（8）：刪除鍵刪字按鍵間隔直方圖

從圖（7）觀察到按壓時間分布的密集區相當明顯，高峰前後的區塊人數分布較均勻。大部分的人集中在 50 至 150 毫秒之間，從 200 毫秒到 1000 毫秒之間則較分散，因此這個特徵能用來區分刪除字元時按壓時間較長的族群。從圖（8）中觀察到資料的分布廣泛，在每個區間的人數也不多，能有效區分打字者。

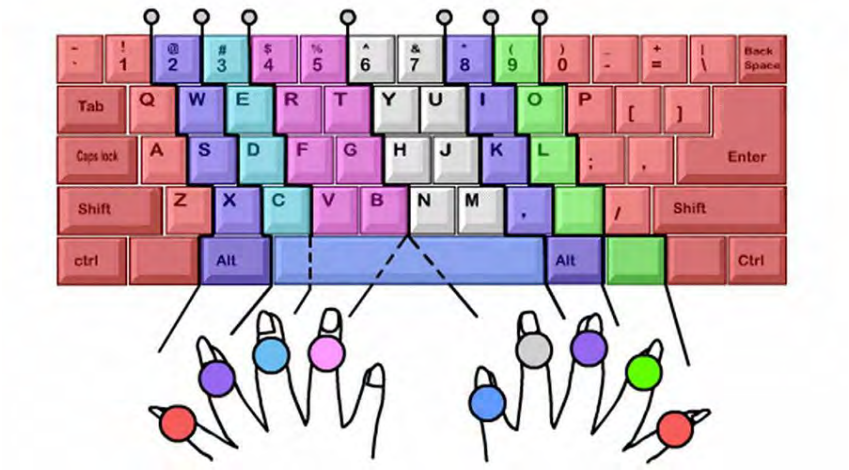


圖（9）：用刪除鍵鍵時的按鍵間隔時間及按壓時間的相關性散布圖。

從圖（9）中，可以觀察到每個人使用刪除鍵的行為差異不大，無法顯著區分不同打字者，但是用刪除鍵按鍵間隔時間能夠將打字者大致分群。

（三）手指在各分區之下的不同行為表現

因手指靈敏度有異，不同手指在打字時會有不同的行為表現，因此，我們想要利用此特性做出一個具有辨別度的特徵。我們應用一般打字教學的示範（圖（10）），將鍵盤依各個手指涵蓋的範圍區分為 8 塊（圖（11）），並分析不同人在打字過程中連續按壓同一區的按鍵時的行為。



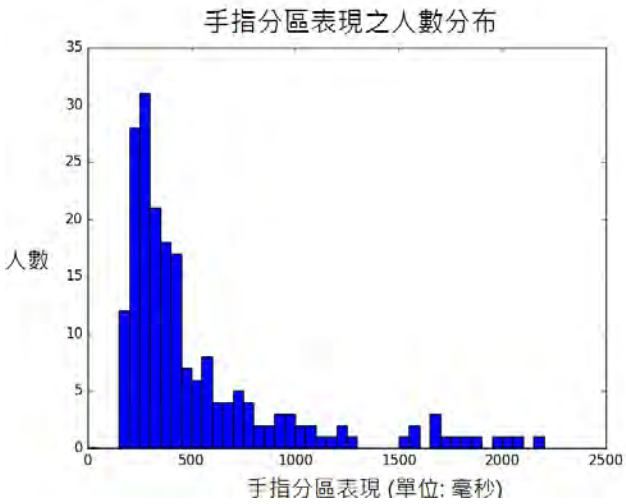
圖（10）：手指分區圖[1]

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	;
Z	X	C	V	B	N	M	,	.	/

圖（11）：實際採計的按鍵範圍

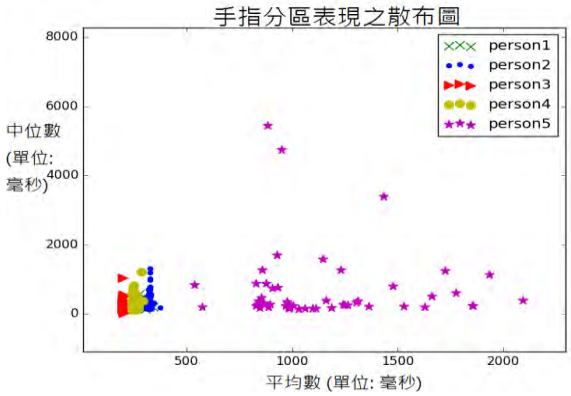
我們假設大部分打字者都是以正確手勢打字，在打字者連續按壓屬於同一手指區域的按鍵時，將兩按鍵間的距離標準化並除以時間差距，當作該打字者手指的打字速度（以下以「手指分區表現」稱之），並進行分析（圖(12)以及圖（13））。

[1] 此圖片選自 http://krs.school.hk/~krs-comp/typing.html，為打字之正確手勢圖

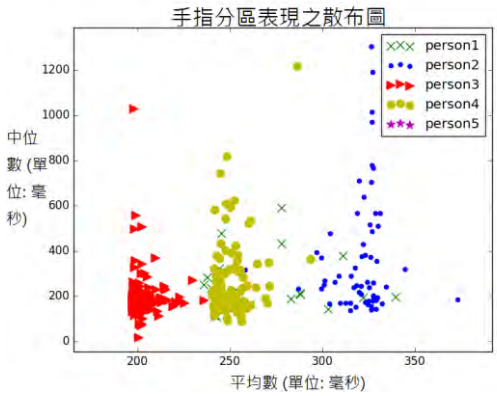


圖（12）：手指分區下的速度之直方圖

從圖（12）可觀察到除了集中在 200 至 500 毫秒之間的打字者以外，其餘打字者的手指分區速度分布較分散，因此可用來辨別不同打字者。



圖（13）：鍵盤分區下連續鍵擊間隔時間的平均數與中位數的相關性散布圖



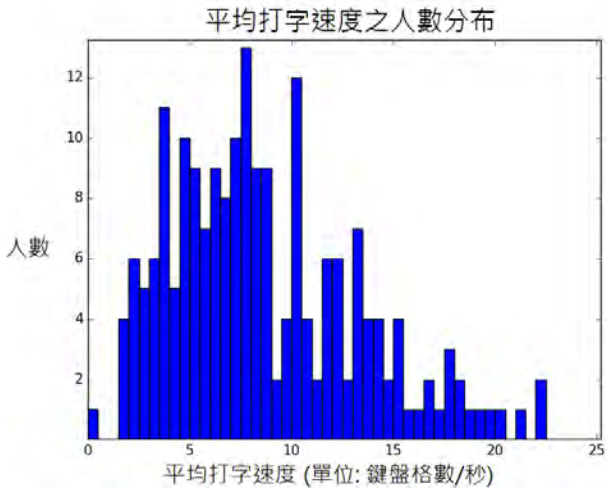
圖（13-1）：圖（13）針對打字者 1~4 號的散布圖局部放大

從圖（13）中可看出這個行為特徵對於部分打字者具有顯著辨別度，且屬於同一打字者的數據分布相當集中，尤其以平均數表現便能明顯區隔。

（四）兩手分別的打字速度

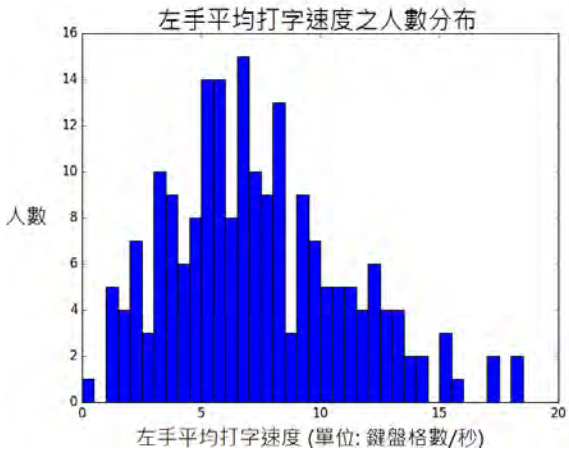
我們假設一般人在打字的時候，慣用手的靈活程度會使雙手的打字速度有所差別，因此可以將兩手的打字速度做為區分的特徵。本專題研究以「按鍵速度」表示打字的速度，以左手和右手的按鍵速度表示打字者兩手的打字速度。

我們在打字者連續按壓屬於同一半邊的按鍵時，將兩按鍵間的距離標準化並除以時間差距，計算出打字速度。這項行為的分析可以分成兩項，第一項是算出左右手分別的平均速度得到兩個數值，第二項則是全部平均，得到一個人兩手的平均速度（圖（16）至圖（18））。

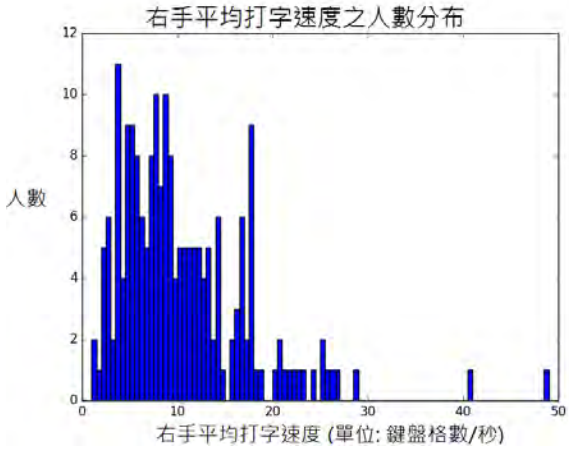


圖（14）：左右按鍵速度平均數直方圖

圖（14）呈現了不同使用者左右手打字速度的分布。相對於其他使用者行為，左右手打字速度的分布更接近於常態分布。這是因為當探討的空間範圍擴大至整個鍵盤時，對於使用者行為的統計將更為全面。



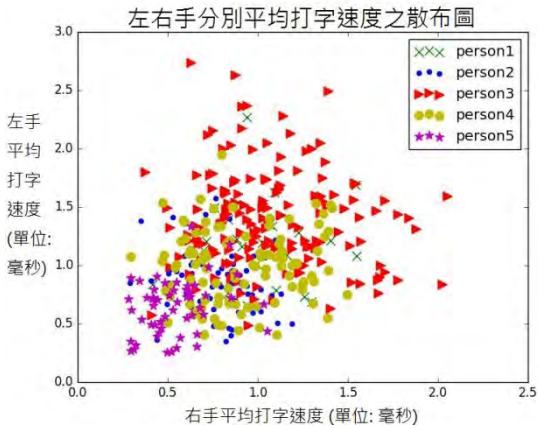
圖（15）：左手平均打字速度直方圖



圖（16）：右手平均打字速度直方圖

左右手打字速度的散布圖（圖(17)）雖不像其他特徵一樣能顯著的區隔出使用者，但由於每個使用者各自的數據仍有集中在同一區的趨勢，因此此特徵可能仍可以用來做身份辨識。我們在訓練完模型後會再以 Leave one out 方式分析每個特徵對於辨識模

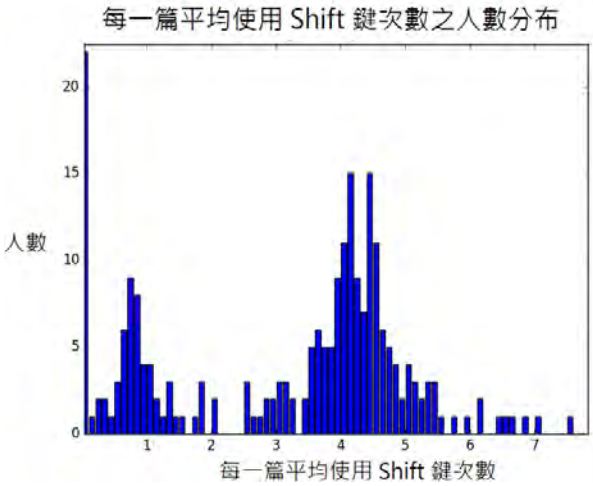
型的影響。



圖（17）：左右手分別的按鍵速度相關性散布圖。

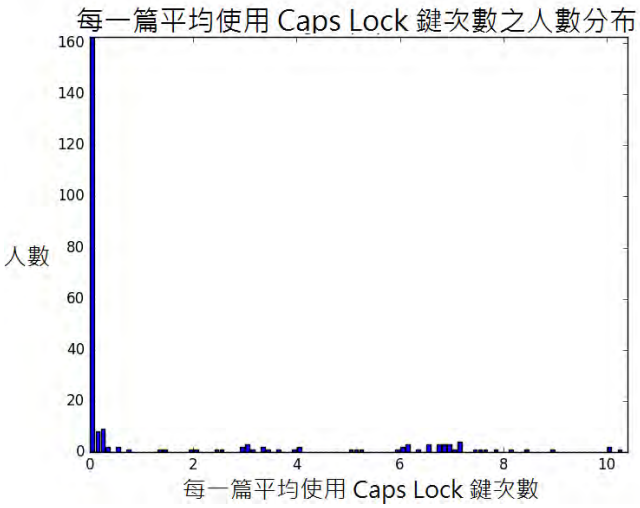
（五）Shift 鍵與 Caps Lock 鍵的使用習慣

個人的習慣在本次專題研究扮演非常重要的因素，因為習慣不容易改變，可以做為代表一個人的特徵。我們平常觀察到不同打字者對於切換大小寫的習慣有明確差異，有些人只用 Shift 鍵，有些人只用 Caps Lock 鍵，有些人會交替使用，因此能用來將打字者分群。我們記錄幾個人的打字過程後，分析他們對於 Shift 鍵及 Caps Lock 鍵的使用並製作成圖表（圖（18）至（20））。



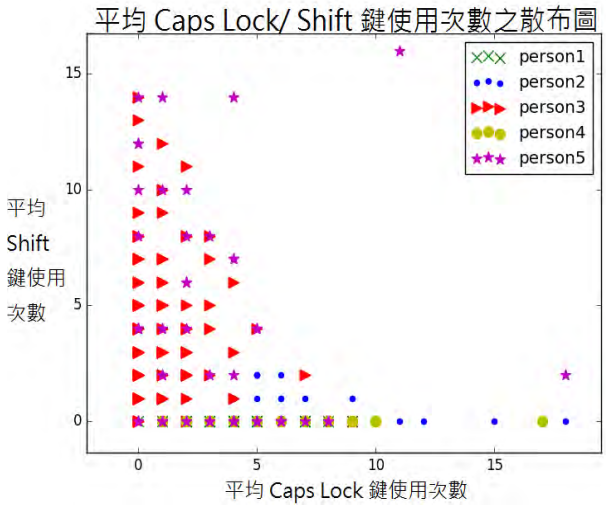
圖（18）：打字者平均每一篇平均按 Shift 鍵的次數的人數分布。

圖（18）顯示一般人使用 Shift 鍵的習慣能明顯分成兩類，因此我們可以使用此特徵來將使用者分群。



圖（19）：打字者每一篇平均按 Caps Lock 鍵的次數。

Shift 鍵與 Caps Lock 鍵常用以切換大小寫字元，從圖（19）觀察到，大部分我們收集的打字紀錄中都沒有使用 Caps Lock，除了可能因為打字者輸入的內容原本就沒有大寫字母以外，也可能是因為打字者是使用 Shift 鍵切換大寫功能，因此此特徵可能可用於辨識少數習慣使用 Caps Lock 的打字者。

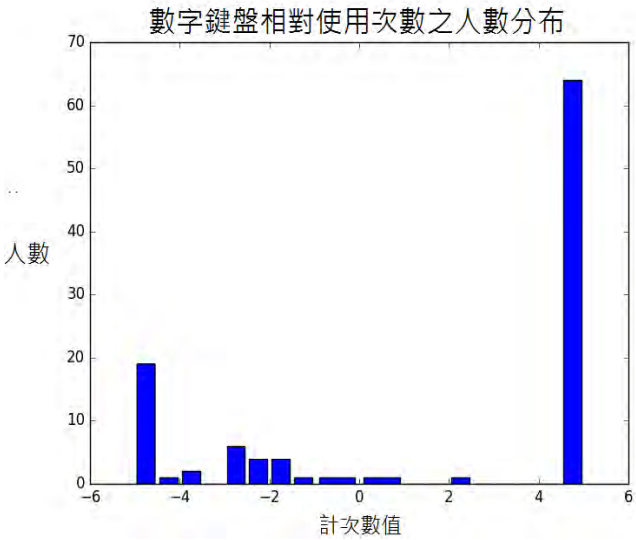


圖（20）：利用 Caps Lock 鍵及 Shift 鍵來輸入大寫字母頻率相關性散布圖。

從散布圖觀察到打字者切換大小寫的習慣的確可以用 Shift 鍵及 Caps Lock 鍵的使用分成三群：常使用 Shift 鍵（圖中 3 號打字者）、常使用 Caps Lock 鍵（圖中 1 號及 4 號打字者）、兩者交替使用（圖中 5 號打字者），因此此特徵亦可用來將打字者分群。

（六）數字鍵輸入習慣

我們把輸入數字的行為分成兩種：第一種是使用鍵盤上方的數字列，第二種則是某些鍵盤才會有的側邊或外接數字鍵盤。我們用相對計次來判斷打字者使用數字鍵的情況。對於每次使用數字列的輸入，將數值（初始值為 0）加上 5，每次使用數字鍵盤的輸入則將數值減 5，得到的數值除以數字輸入的次數，將收集到的打字紀錄做成圖表（圖(21)）後分析。

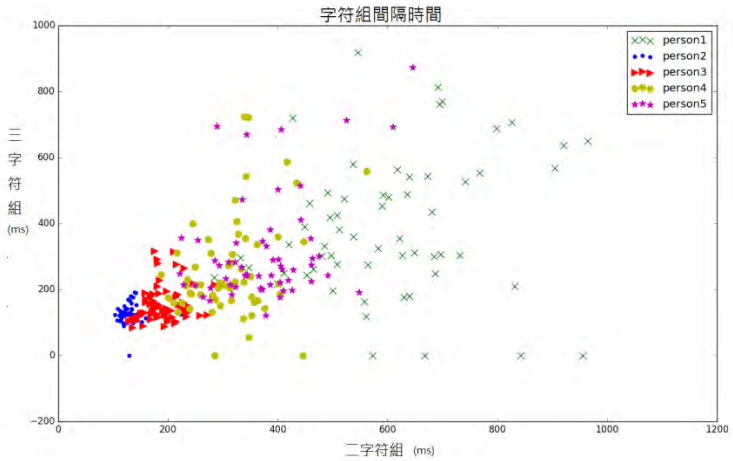


圖（21)：數字鍵盤相對計次之直方圖。

數值愈偏右的正數值代表使用鍵盤上方數字列的次數愈多；反之，愈偏左的負數值代表使用鍵盤右方數字盤的次數愈多。我們從上圖觀察到，打字者有不同的習慣，但其中按壓上方數字列的打字者比較多。直方圖的圖形與預期吻合度極高，數字輸入的行為固定者（數值極高或極低）為最明顯的兩族群，行為不固定者則為少數。但此特徵也有可能受硬體限制影響（有的鍵盤不具備右側數字鍵），因此我們會在最後的 Leave one out 分析觀察此特徵 是否具足夠的辨識度。

（七）字符組的按壓間隔時間

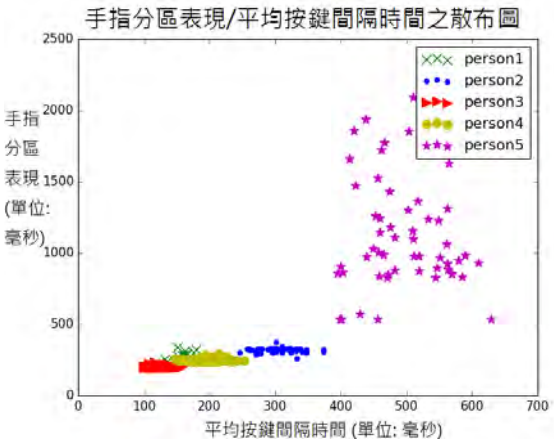
我們認為打字者對於常見的字串會有一定的熟悉感以及不同的習慣，所以我們上網查了英文字中最常出現的字符組，在打字者輸入這些字符時，記錄其按壓間隔時間。例如，“T-H” 或是 “T-H-E” 都是常見的字符。我們將兩個字符組以及三個字符組都記錄下來並繪製散布圖（圖(22)）。



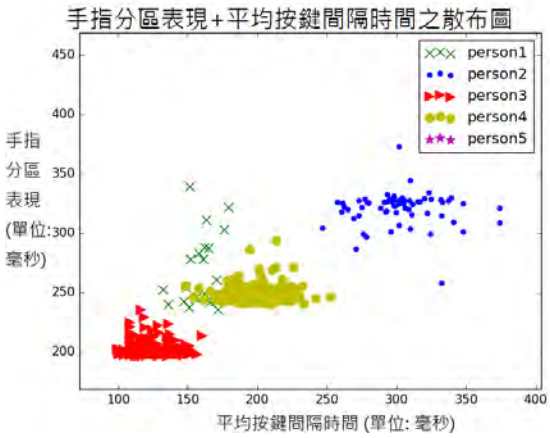
圖（22）：兩字符組與三字符組之間隔時間散布圖。

二、綜合分析

分析完以上的特徵之後，我們將辨別度最明顯的幾個行為特徵抽取出來，分別繪製散布圖並分析結果（圖（23）及圖（24））。



圖（23）：手指分區表現與平均按鍵間隔時間的相關性散布圖



圖（23-1）：圖(23)針對打字者1~4 號的散布圖局部放大

圖（23）採用先前最具辨別度的按鍵間隔時間與手指分區速度特徵，其結果相當顯著。不僅分區明顯，且每個人對應這兩個特徵的數值非常集中。

我們分析各個特徵的散布圖後，挑了三個容易區分打字者族群的特徵做同時分析，三個特徵分別為：按壓時間、手指分區速度、以及 Caps Lock 鍵的使用。從圖（24）觀察到利用這些特徵可以顯著區分不同的打字者。

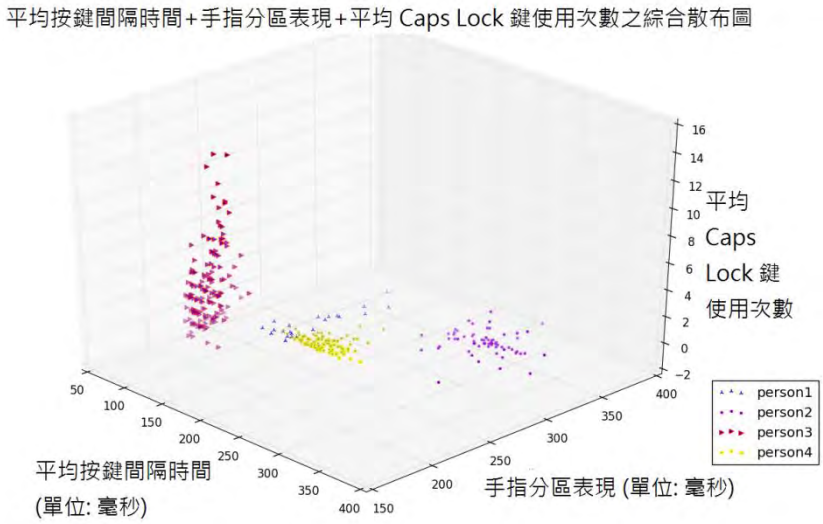


圖 (24)：三維散布圖。以平均按鍵間隔時間、手指分區表現以及平均 Caps Lock 鍵按壓次數為參數所得的散布圖。

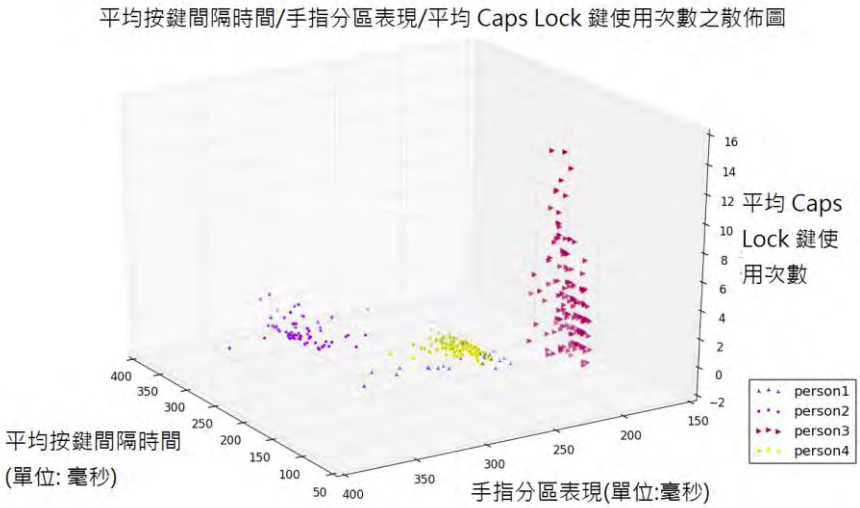


圖 (24-1)：三維散布圖。同上，但從另一角度觀察。

三、打字者輸入行為特徵抽取

我們將所有數據平分為一個訓練檔案以及一個測試檔案。在訓練時，我們著重於抽取特徵，並利用支持向量機建立機器學習模型。

透過完整的統計資料，我們可以分析出使用者的行為表現；而這些行為表現便成了作為辨識依據的「特徵」。套入模型前，我們從直方圖與散布圖所提供的線索，提出一些假設。從最簡略的 Shift 鍵以及 Caps Lock 鍵為例子，發現很顯著比例的打字者用 Shift 鍵的次數相較於 Caps Lock 鍵的使用多出很多，對於二元分類有優勢，與模型效能比較，效果與預期的差不多，方法儘管簡單，卻扮演著重要的角色。最核心的「按

壓時間」以及「按鍵間隔時間」還有衍生出來特別針對刪除鍵鍵的四個數值，進入支持向量機模型內比照產生行為的方法計錄第 j 個鍵的按下時間（t_j）放開的時間（v_j）後相減前項得到（D_j），最後把全部的按壓時間相加求其平均（$\frac{1}{n}\sum_{j=1}^{n} D_j$）。在刪除鍵的條件之下再加入限制按鍵代碼（$K_i = 8$）。字符組間隔時間的取法與計算刪除鍵間隔時間的方法基本上是一樣 的，不過會在按鍵代碼多加一些限制，將常見的連續組合記錄下來。

至於按鍵間隔（I_j）則是第 j+1 個鍵按下的時間 （t_{j+1}） 減掉第 j 個按鍵放開的時間（v_j），之後再加總取平均值（$\frac{1}{n}\sum_{j=1}^{n} I_j$）。

手指分區也是基於前兩類行為所建構出來的：當一串字元都是在同一區塊（參考圖（10）） 就會不斷的計錄時間（$\sum_{j=1}^{n} I_j + \mathrm{D}_j$）最後除上按鍵次數（$\frac{1}{n}\sum_{j=1}^{n} I_j + D_j$） 得到某區域中，平均輸入一個字元所耗時間，綜合每一個區塊的平均時間(可能重複)，最後再做一次平均，得到「手指分區」這項特徵。

最後一個類型為打字速度。同樣的，記錄第 j 個鍵的按下時間（t_j）放開的時間（v_j），但是條件是只記錄同一手的輸入區域，這個是從手指分區速度特徵所衍生出來的，分區方法簡化了，但在距離的地方考慮較深入，加上模型內數據是兩次輸入就運算一次，在同一區的條件下以最後一個時間點（v_{j+1}）減去最初時間點（t_j）先取得過程的時間，因為以毫秒計算所以乘上 1000 轉換為秒的單位(以 t 簡稱)，將按鍵間的距離標準化以後取得兩個點的座標 （（$x_1, y_1), (x_2, y_2$））， 再套入距離公式（$\sqrt{(x_2 - x_1) + (y_2 - y_1)}$）取得過程的距離（以 d 簡稱），定義速度（以 S 表示）為距離除上時間（$\frac{d}{t}$），最後一步依舊是求平均值（$\frac{1}{n}\sum_{j=1}^{n} S_j$），取得一個具代表性的數值，也就是按鍵速度的特徵。

我們將所有特徵整理於表（1)，時間、空間、特殊鍵，以及連續鍵共四類特徵被涵蓋於我們所提出的模型。

時間上的分析	**空間上的分析**	**特殊鍵的分析**	**字符組的分析**
按鍵按壓時間	手指分區表現	刪除鍵的使用習慣	連續兩個字符
按鍵間隔時間	右手打字速度	Shift 鍵 / Caps Lock 鍵按壓次數	連續三個字符
打字速度	左手打字速度	數字鍵相對計次	
表（1): 特徵整理			

四、模型訓練過程

本專題研究中所提出的特徵，可以有效的描繪打字行為，將每個受測者個別的行為分開。理論上，任何一種分類模型（classification models）都可以利用這些特徵來達到辨識受測者的效果。經常被採用的方法包括人工神經網路以及支持向量機等，近期興起以神經網路的深度學習（deep learning）方法進行機器學習領域的研究，但由於深度學習需要非常大量的訓練資料，一旦資料過少便容易發生過度擬合（overfitting）的情況，因此本專題研究使用了目前學術界上分類效果最優良的模型之一的支持向量機（Support Vector Machine, SVM）作為分類模型，驗證特徵的有效性。

將 SVM 應用於身份辨識的概念為：對於每個受測者，我們將其視為一個類型（Class），並將第 i 筆輸入資料所抽出的特徵表示為 (x_i, y_i)。x_i 為一向量，長度為特徵的數量 N；而 y_i 則以 +1 與 -1 代表此資料是否為該受測者輸入。透過這樣的表示法，每筆資料等價於在 N 維的超空間中，找到一個超平面可以將兩種資料給區隔開來，並可以下列數學式來表達其超平面：

$$w \cdot x - b = 0,$$

w 與 b 為模型所需要學習的權重，w 為超平面的法向量，b 則是從原點位移的偏移量。直觀而言，超平面在超空間中離最近資料點的距離理當越遠越好。在數學上，我們期望可以由 $w \cdot x - b = 1$與 $w \cdot x - b = -1$兩個超平面來表達。透過計算，我們可以得到兩個超平面的距離為 $2/|w|$，$|w|$為向量w的長度。因此，整個問題的目標，便是在滿足測試資料符合被超平面區隔的狀態下，將w最小化。

以下方數學式表示：

$$argmin_{w,b}\ |w|^2/2, s.t. y_i\,(w \cdot x_i - b) \geq 1。$$

此外，為了增加模型的複雜程度，使其能夠適應更多元的資料，我們也使用了 Radial basis function kernel（RBF Kernel）來擴展 SVM。 Kernel 在 SVM 的功能是將原本線性的權重擴展為非線性。假設 kernel function 為 k，可以數學式 $k(x, x') = \phi(x) \cdot \phi(x')$ 來表示。而 RBF Kernel 則是將 kernel function 訂為 $exp(-||x - x'||^2/2\sigma^2)$，$\sigma$則是模型中需要手動調整的參數。

實作方面，我們使用了由國立台灣大學資訊工程學系機器學習實驗室所推出的 libsvm 工具來進行模型訓練。（Chang & Lin, 2011） 此外，我們也利用 Grid Search 與 Cross Validation 的方式來進行參數調教，找出能使 SVM 模型達到最佳效果的參數。

為驗證本研究所提出特徵與模型的有效性，我們設計了實驗來進行評估，
並透過特徵重要性分析以及案例分析來探討模型有效的原因。

陸、研究結果與討論

一、實驗設計

（一）實驗資料

我們邀請年齡從十歲到七十歲不等的兩百三十位受測者（圖 (25)）各輸入數十句英文句子，每句約二十個單字，來源是從知名新聞網 BBC 的資料庫隨機剪取（Greene & Cunningham, 2006）。英文難度適中，偶爾會出現較不常見的單字。網頁會記錄受測者輸入的按鍵編號、按下到放開的時間，以及放開到按下的時間並存至資料庫。最後所收集的資料數共計 9000 篇。

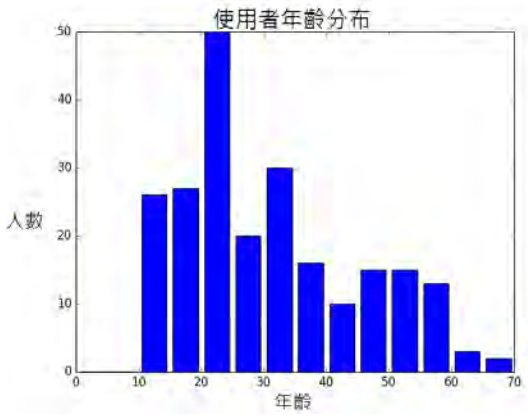


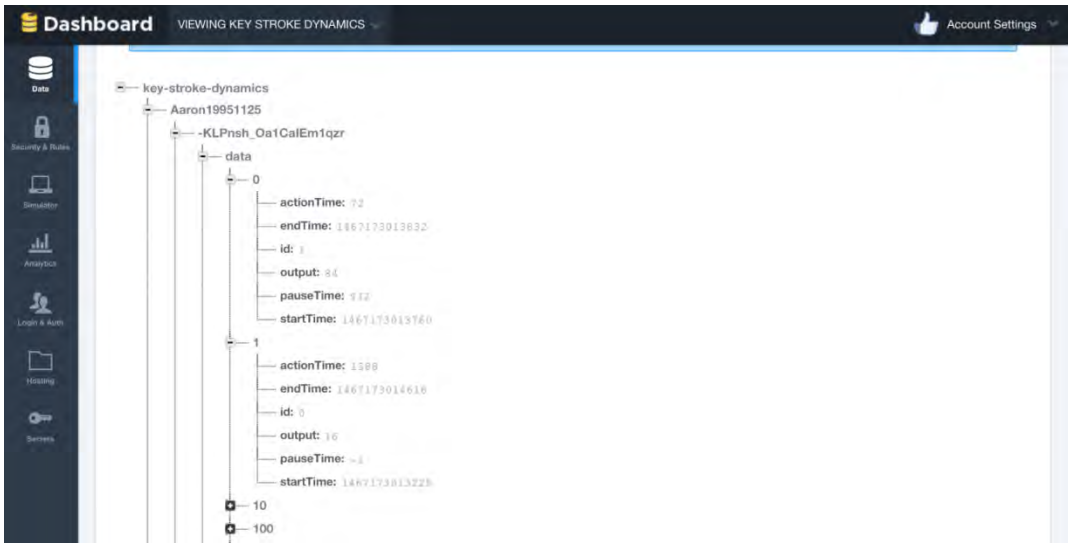
圖 (25)：受測者年齡分布直方圖

（二）實驗介面

許多前人研究的受測者人數都相當少，且每位受測者所輸入的通常都是極短且相同的單字，我們希望能重新設計實驗，收集更多資料以驗證本專題研究的成效。

我們建立一個網頁介面（圖 (26)） 供受測者打字，網頁使用 Javascript 語言偵測並記錄受測者打字行為，並將受測者輸入的每筆數據以 JSON 格式儲存至 Firebase 線上即時資料庫（realtime database）（圖 (27)）。

圖（26）：供受測者打字頁面



圖（27）：本專題研究之資料格式及資料庫介面

其中「Aaron19951125」為一個類型（Class），「-KLPnsh_Oa1CalEm1qzr」為此受測者輸入的其中一篇文章，「data」為一個陣列，陣列內容為該受測者此篇文章的所有鍵擊，每一個鍵擊都包含按壓時間、間隔時間、按鍵編號等資料。

（三）評估方式

本專題使用以下四種數據評估所提出特徵在實驗資料的辨識表現：

準確度（Accuracy）：$A = \frac{所有預測對的項數}{總項數}$

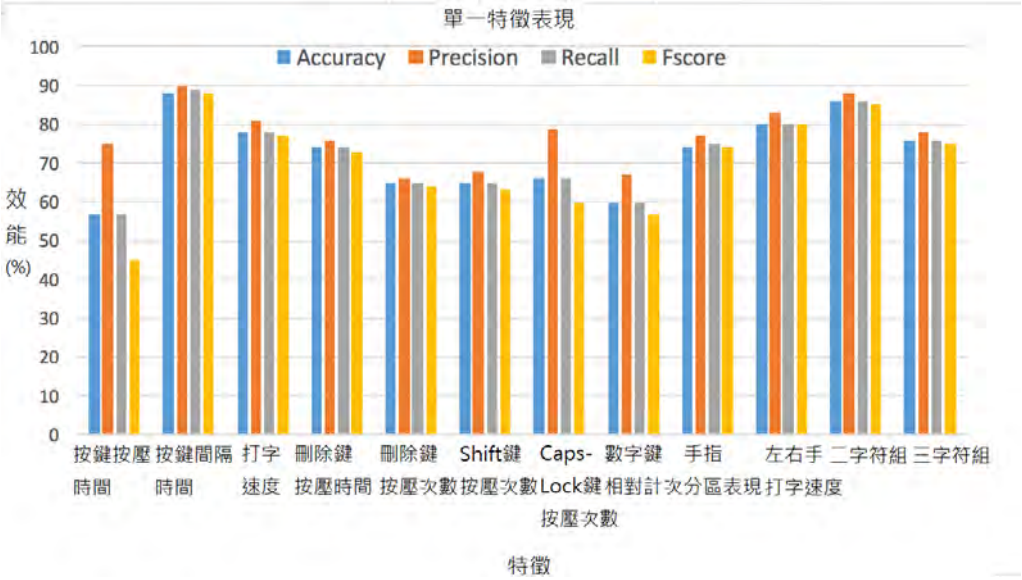
精確度（Precision）：$P_i = \frac{正確預測為\ i\ 的項數}{預測為\ i\ 的項數}$

召回率（Recall）：$R_i = \frac{正確預測為\ i\ 的項數}{確實為\ i\ 的項數}$

F-score 值：$F_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}$

二、特徵分析與討論

我們利用直方圖分析單一特徵與其他特徵相比之下的準確度。



圖（28）：單一特徵的準確度比較

圖（28）分析了僅使用不同單一特徵所能達到的效能。在所有特徵中， 按鍵按壓時間最為有效。此結論也與前述的使用者行為分析的結果一致。而效果最差的特徵為左右手分別打字速度。這是因為當數據的分布為常態分布時，將會有相對多數人擁有類似的左右手分別打字速度，也造成此特徵無法有效區隔出使用者。

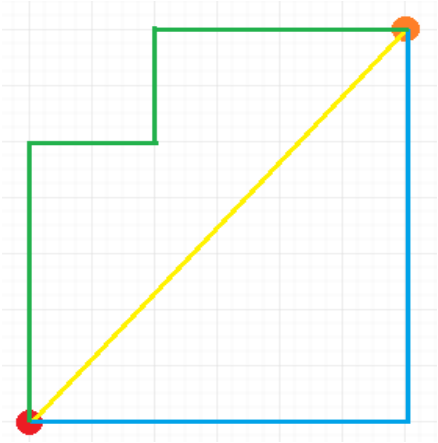
三、基準化資料集（Benchmark Data Set）分析與比較

在生物辨識的範疇，Equal Error Rate（EER） 是一個常見的評估方式，它取的是誤警率以及入侵成功率相同時的座標點，使得辨識系統能夠找到合適的平衡點。EER 愈低代表系統準確度愈高。

為了驗證我們所提出方法的有效性，我們與以下五個現有的方法進行比較：

（一）Euclidean：當 $q=(q_1,q_2,\ldots,q_n)$，$p=(p_1,p_2,\ldots,p_n)$，歐幾里德距離的算法為 $\sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}$ ，代表在高維度中 p 與 q 之間的距離。歐幾里德距離在機器學習中用來測量兩向量之間的距離。是各種最短距離分類器中用最常用以計算距離的工具。「距離」表示相似的程度，即距離越短相似度越高(Kramer, 1991)。

（二）Euclidean (normed)：x 為一個任意維度的向量，當 $x=(x_1,x_2,\ldots,x_n)$，$||x||=\sqrt{\sum_{i=1}^{n}x_i^2}$，可用來計算向量至原點的距離。此距離與相似程度呈正相關，此算法亦為最短距離分類器中一種常見的方法（Celebi, Celiker, & Kingravi, 2011）。



圖（29）：歐幾里德距離表現兩點之間的距離，以此二維座標圖為例，取長度為 $\sqrt{85}$ 的黃色直線

（三）**Fuzzy Logic**：Fuzzy Logic 常在控制理論或是人工智慧的領域中使用，它能夠陳述「不精確」的情形，讓答案不一定只是「0」或「1」的二元項，而是視情況折衷。De Ru 和 Eloff 便使用 fuzzy logic 於他們的辨識系統之中，使他們所採集的特徵在極相似的範圍之間也能被有效區分，如此的機制與人思考較為接近，也讓電腦有辦法處理感知相關問題（De Ru & Eloff, 1997）。

（四）**k－Means**：k-Means 是一種聚類（cluster）的演算法，將相似的物件聚集以做分類。這種具類分析的方法，最常應用到上面所述之歐氏距離。 El Bakrawy 等人便在他們的辨識系統中以 k-平均聚類把 n 個點（可以是樣本的一次觀察或一個例項）劃分到 k 個聚類中，使得每個點都屬於離自己最近的均值（此即聚類中心）對應的聚類，以之作為聚類的標準（El Bakrawy, Ghali, ella Hassanien, & Kim, 2011）。

（五）**Neural Network**：Neural Network 是常見的一種機器學習方法，已被用於處理各種問題，包括語音辨識。它模仿大腦的神經網路，透過學習來加強效能。表（2）及圖（30）為本專題研究所提出的方法以及五個前人方法在 benchmark dataset 的成效比較。

辨識方法	平均 EER
Euclidean	0.1706
Euclidean (normed)	0.2153
Fuzzy Logic	0.2213
k Means	0.3722
Neural Network	0.8283
本專題提出之方法	0.1460

表 (2)：辨識方法以及平均 EER 之比較

在鍵盤生物辨識這個區塊的分類中，演算法必須有辨法區分大量且雜亂的數據。因此，在高維度進行分類是有必要的，否則在一平面上系統無法做有效的區分。以歐幾里德距離為例，因為這種演算法能在高維度進行演算，故明顯達到較低的 EER。我們使用的 SVM 也應用到歐幾里德距離來演算，加上系統調整參數，達到更高的準確度。反之，以 *k* -Means 為例，因為它在超過三維後計算會變得太複雜，難以快速求精確的值，因此不適用於鍵盤生物辨識這個區塊。

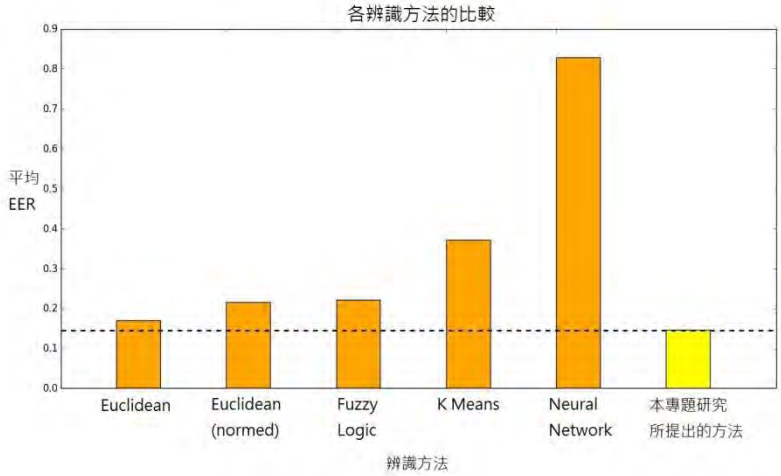


圖 (30)：各種辨識方法的 equal error rate 比較。

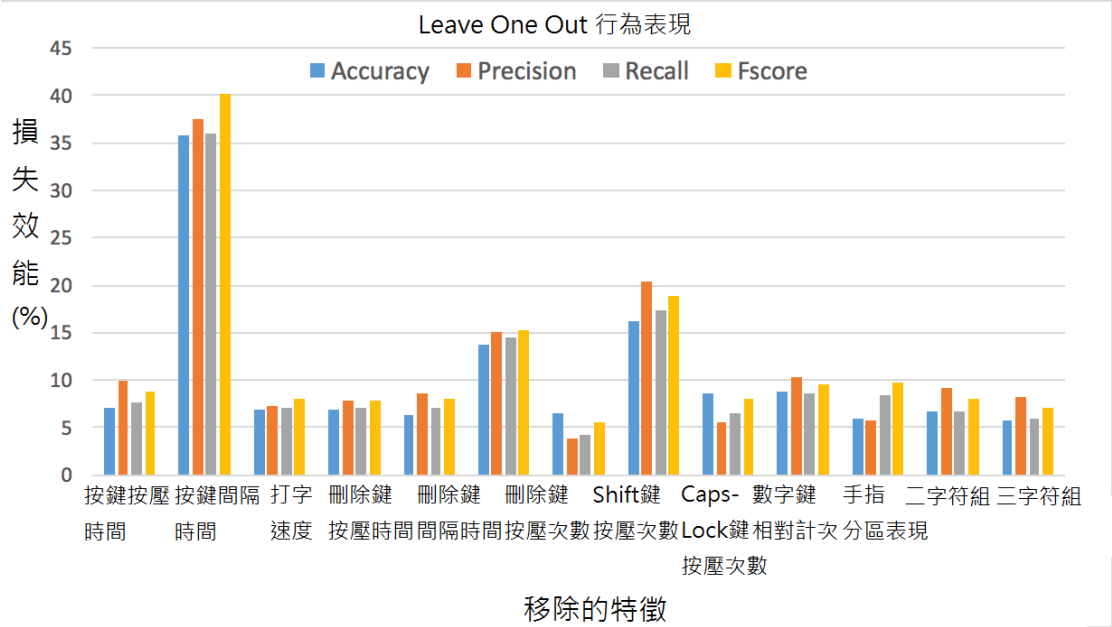
四、特徵重要性分析

我們使用 Leave one out 特徵分析（圖 (31)）從整體的角度探討特徵之間相互的互補性。觀察到排除按鍵間隔時距特徵之後，四個數值都顯著降低；排除左右手打字速度特徵後，準確度反而有正成長。

Leave one out 的分析有趣之處在於它明確顯示一個特徵單獨的準確度不一定代它與其他特徵能夠搭配得好。相對的，一個特徵單獨的表現不顯眼，也不代表它的互補性

不高。例如，Caps Lock 鍵單獨的準確度在 5% 左右雖然算低，但相對於其他特徵，它的互補性卻排第二。反之，左右手分別打字速度在 8% 左右，相較於其他的特徵能排到第二名，但是它的互補性卻不佳，與其他特徵共同判斷後準度甚至變得更差了。

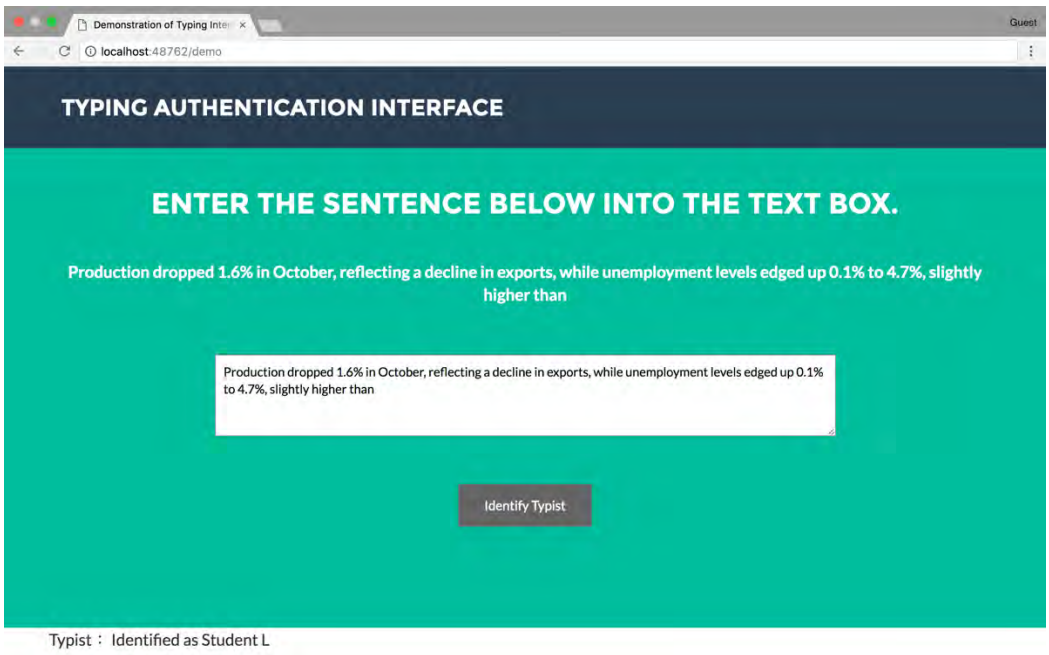
我們的特徵普遍互補性很高，基本上都能與其他特徵結合達到更高的準確度。也就是說，除了前人有略微提到的類似於記錄時間的特徵，我們所提出的新的特徵也都具代表性。



圖（31）：拿掉各個特徵後與所有特徵合併的成效差異

五、建立個人模型

因為身分辨識會以單一使用者來做區分，我們決定對每個使用者建立單獨的辨識模型。我們將目標類型（Class）標記為 1，將剩下所有的類型標記為 0，對於 1 這個標記做訓練。因為一個類型的資料數在整體之中容易被 SVM 覆蓋或忽略，所以為了調整抽樣偏差（sampling bias），我們使用兩種方法來訓練這些模型：超取樣(oversampling）以及調整權重。訓練時，我們發現超取樣的效果較好，因此選擇它來跑所有人的模型。最後訓練完，我們得出的平均結果為：準度 95.3%、精度 96.4%、召回率 95.3%、F-score 值 95.0%。為了驗證我們系統在日常生活中的實用性，我們設立一個以 python 語言為基礎的伺服器：Tornado。把個人的模型建立好後，使用者在一個介面上輸入一個句子，將數據從伺服器傳到系統，讓打入的資料跟模型比較，並回傳結果。辨識所需的時間很短，且效果相當不錯，也將我們系統應用的可行性提高不少。圖(32）為辨識介面。



圖（32）：辨識打字者的介面。

六、案例分析

我們將所有受測者個人的精度、召回率以及 F-score 值取出並做排序，探討數值最高以及最低的幾位受測者的行為表現。我們發現受測者精度、召回率以及 F-score 這三個測量標準的數值集中；精度高的受測者，其他的測量結果也很好。平均準確度最高的是第 114 號受測者，其三項標準的平均達 98.4%。第 11 號受測者名列第二，三項標準的平均達 97.6%。反之，有 12 位受測者三項標準的平均數值為 0。觀察這些人各個特徵的行為表現，發現準確度最高的幾個受測者都有較多的樣本數（30 至 160 句英文句子）；另外，他們大多有一到兩個數值較為突出的特徵。像是第 114 號受測者的同一鍵擊按壓時間的數值相對其他受測者而言非常低，我們推測是這樣的特殊打字習慣使得某些特定的受測者能夠被輕易區分。13 個為準確度為 0 的受測者，樣本數皆非常的少，因此無法獲得準確的判斷結果。上述結果可以顯示，當打字者輸入的行為資料量到達一定的程度以後，我們的模型便能有極高的準確度。針對單一特徵作分析，發現結果與原先相同。數據原本偏高的受測者在單一特徵下的表現也都不差。像是跑同一鍵擊的按壓時間的時候，上段所提到的第 114 號受測者的數據同樣非常好，準確度高達 99%。

柒、結論與應用

有別於過去繁複的手續，本專題研究提出了一個運用打字行為特徵的身份辨識方法，經過分析後，提出能有效用於辨識身份的打字特徵。過程中收集不同人的打字資訊，對時間、

空間，以及特殊鍵的使用進行分析，並觀察不同行為表現的分布。本專題研究除了觀察單一特徵的準確度之外，從 Leave one out 分析和案例分析中，了解到各個特徵的互補性以及對於身份辨識的影響。最後訓練出來的模型，其 equal error rate 為 0.1460，比前人的研究成果更好。我們另外為所有打字者建立個人模型，發現與伺服器接上後能夠在非常短的時間內達到有效辨識。

本專題所提出之方法只需要一般鍵盤，成本低且可行性極高，預期可應用之範圍相當廣泛，涵蓋銀行、海關、個人通訊設備等。在帳號註冊及往後登入時，只要讓打字者輸入一段足夠提取所需特徵的文字，便能判斷鍵入者是否為同一人。打字行為不易被他人模仿或複製，我們的系統也具備便宜、精確及應用性廣泛等優勢。指紋可以採集，密碼可以盜取，證件可以偽造，然而我們的鍵盤辨識系統，必須出於本人意願方能認證，乃高度安全的系統。

捌、參考文獻

Celebi, M. E., Celiker, F., & Kingravi, H. A. (2011). On Euclidean norm approximations. *Pattern Recognition, 44*(2), 278-283.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 27.

De Ru, W. G., & Eloff, J. H. (1997). Enhanced password authentication through fuzzy logic. *IEEE Expert, 12*(6), 38-45.

El Bakrawy, L. M., Ghali, N. I., ella Hassanien, A., & Kim, T.-h. (2011). *A rough k-means fragile watermarking approach for image authentication.* Paper presented at the Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on (pp. 19-23). IEEE..

Gaines, R. S., Lisowski, W., Press, S. J., & Shapiro, N. (1980). *Authentication by keystroke timing: Some preliminary results*. (No. RAND-R-2526-NSF). RAND CORP SANTA MONICA CA.

Greene, D., & Cunningham, P. (2006). *Practical solutions to the problem of diagonal dominance in kernel document clustering.* Paper presented at the Proceedings of the 23rd international conference on Machine learning. (pp. 377-384). ACM.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal, 37*(2), 233-243.

Loy, C. C., Lai, W., & Lim, C. (2005). Development of a pressure-based typing biometrics user authentication system. *ASEAN Virtual Instrumentation Applications Contest Submission*.

Saevanee, H., & Bhatarakosol, P. (2008). *User authentication using combination of behavioral biometrics over the touchpad acting like touch screen of mobile device.* Paper presented at the Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on (pp. 82-86). IEEE..

附錄

附錄 (1): 鍵盤代碼 (ASCII Code)	
Key	code
0~9	48~57
a~z	65~90
Shift 鍵	16
backspace	8
delete	46
space	32
Caps Lock	20

附錄 (2): 特徵表現				
	Acc	Precision	Recall	F-Score
按鍵按壓時間	3.61%	34.49%	40.00%	13.35%
按鍵間隔時間	16.34%	15.57%	21.49%	15.77%
打字速度	8.23%	8.77%	31.85%	12.13%
刪除鍵按壓時間	4.25%	10.93%	15.95%	8.48%
刪除鍵間隔時間	5.33%	4.65%	24.14%	6.64%
刪除鍵按壓次數	4.47%	9.46%	18.83%	11.00%
Shift 鍵按壓次數	4.30%	4.26%	14.24%	8.12%
Caps Lock 鍵按壓次數	5.11%	4.04%	16.79%	5.19%
數字鍵相對計次	3.96%	3.97%	16.30%	6.29%
手指分區表現	6.54%	8.59%	16.29%	8.22%
左右手分別打字速度	8.55%	5.99%	13.56%	7.02%
二字符組	10.98%	10.07%	16.09%	10.28%
三字符組	7.42%	4.11%	12.50%	4.84%
全部特徵	31.72%	27.77%	26.96%	26.36%

附錄 (3): Leave one out 表現				
	Acc	Precision	Recall	F-Score
按鍵按壓時間	-0.46%	-15.92%	-16.21%	-16.16%
按鍵間隔時間	-31.22%	-41.65%	-41.99%	-45.00%
打字速度	-0.08%	-13.50%	-15.70%	-15.46%
刪除鍵按壓時間	-0.08%	-13.88%	-15.68%	-15.33%
刪除鍵間隔時間	0.39%	-14.60%	-15.70%	-15.55%
刪除鍵按壓次數	-7.51%	-20.71%	-22.51%	-22.11%
Shift 鍵按壓次數	0.15%	-10.18%	-13.22%	-13.27%
Caps Lock 鍵按壓次數	-10.07%	-25.62%	-25.02%	-25.44%
數字鍵相對計次	-2.01%	-11.79%	-15.23%	-15.53%
手指分區表現	-2.17%	-16.28%	-17.14%	-16.99%
左右手分別打字速度	0.93%	-11.93%	-16.88%	-17.16%

附錄（4）：個人模型特徵表現				
	Acc	Precision	Recall	F-Score
按鍵按壓時間	57.21%	75.49%	57.00%	45.35%
按鍵間隔時間	88.34%	90.57%	89.49%	88.77%
打字速度	78.68%	81.77%	78.85%	77.13%
刪除鍵按壓/間隔時間	74.13%	76.93%	74.95%	73.48%
刪除鍵按壓次數	65.05%	66.46%	65.83%	64.00%
Shift 鍵按壓次數	65.58%	68.26%	65.24%	63.12%
Caps Lock 鍵按壓次數	66.81%	79.04%	66.79%	60.19%
數字鍵相對計次	60.47%	67.97%	60.30%	57.29%
手指分區表現	74.36%	77.59%	75.29%	74.22%
左右手分別打字速度	80.32%	83.99%	80.56%	80.02%
二字符組	86.86%	88.24%	86.33%	85.48%
三字符組	76.49%	78.48%	76.89%	75.24%
全部特徵	95.33%	96.41%	95.33%	94.96%

附錄（5）：不同年齡層的表現				
	Acc	Precision	Recall	F-Score
10-14	49.45%	44.59%	45.33%	43.96%
15-19	58.96%	54.45%	51.52%	51.46%
20-24	50.31%	48.14%	44.14%	43.83%
25-29	53.91%	45.83%	47.36%	45.80%
30-34	57.16%	55.35%	52.35%	52.68%
35-39	59.58%	53.14%	52.92%	52.48%
40-44	67.68%	62.07%	59.27%	58.36%
45-49	54.55%	48.68%	51.33%	49.47%
50-54	77.20%	77.54%	72.71%	73.04%
55-59	64.44%	54.80%	54.82%	54.51%
>=60	86.11%	92.86%	66.67%	71.15%

【評語】052507

本作品提出以鍵盤輸入行為分辨使用者身分之研究。

主要使用機器學習方法中的SVM技術來分析使用者在鍵盤上按鍵的行為模式做為身分辨認的一種方法。

若能考慮、分析以及比較不同機器學習方法所能達成的效能，可使此研究更具科學探究精神以及研究工作的完整度。

作品海報

Introduction

At the height of our current era of internet technology, the setting of usernames and passwords has become an essential process, making the issues of convenience and security of password management extremely critical. In this project, we aim to develop an identification mechanism through user keyboard input to circumvent the problems of high costs and low identification accuracy. First, we isolated nine distinct typing characteristics that will contribute positively to user identification. Then, we used Support Vector Machines (SVM) to separate and classify the users. In recruiting 230 typists across different age groups, our system turned out to be more accurate in identifying users than those from previous research studies. Moreover, our system does not require any specialized or customized hardware to work, allowing the system to have a much more extensive deployment and application in everyday life.

Objectives

- To create a typing behavior authentication method based on machine learning
- Propose a method that is low-cost, easy to setup, and highly accurate, allowing typing behavior authentication to be applied in our daily lives

The Proposed Approach

SVM Model Training

Support Vector Machine (SVM) was used as a classification model to validate the effectiveness of different features. We expressed the value of the features of the i th input data as (x_i, y_i). Every data entry is located in a multiple dimension space, so a hyperplane is needed to separate two types of data. This hyperplane can be shown as:

$$w \cdot x - b = 0$$

Because there are many sets of data that require many hyperplanes, we express the distance between two hyperplanes as $2/|w|$. We want to minimize the value of |w| in order to maximize the margin:

$$argmin_{w,b}|w|^2/2 \quad s.t.\ y_i(w \cdot x_i - b) \geq 1$$

By using an RBF Kernel to expand the SVM, the original weight is expanded from linear to non-linear, creating a space transformation:

$$exp(-||x - x'||^2/2\sigma^2), \text{or } \phi(x) \cdot \phi(x')$$

Finally, we used grid search and cross validation to optimize the parameters.

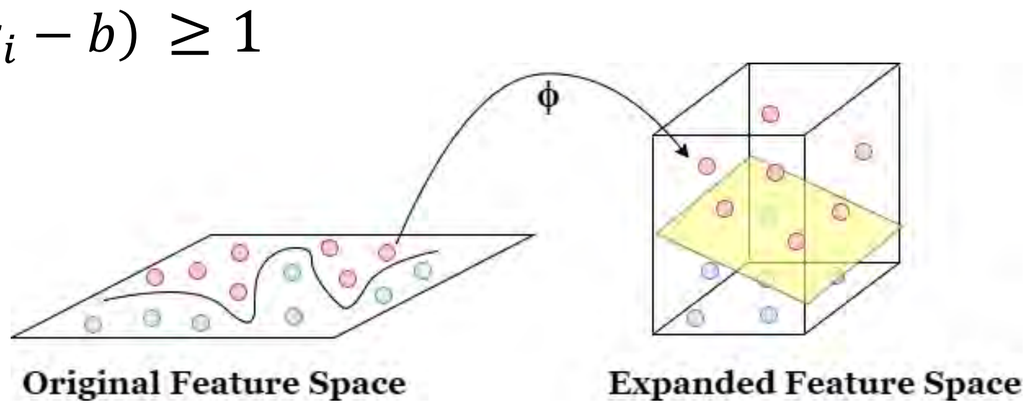


Fig. 1 Data division via kernel methods

Data Collection and Subject Distribution

In our experiment, 230 typists were recruited. The corpus for data collection was the BBC dataset. This dataset consists of 2,225 articles from BBC news.

For each request from typists, the interface randomly selected one sentence from the corpus. The ASCII codes and metadata were collected and sent to a real time database in a JSON format.

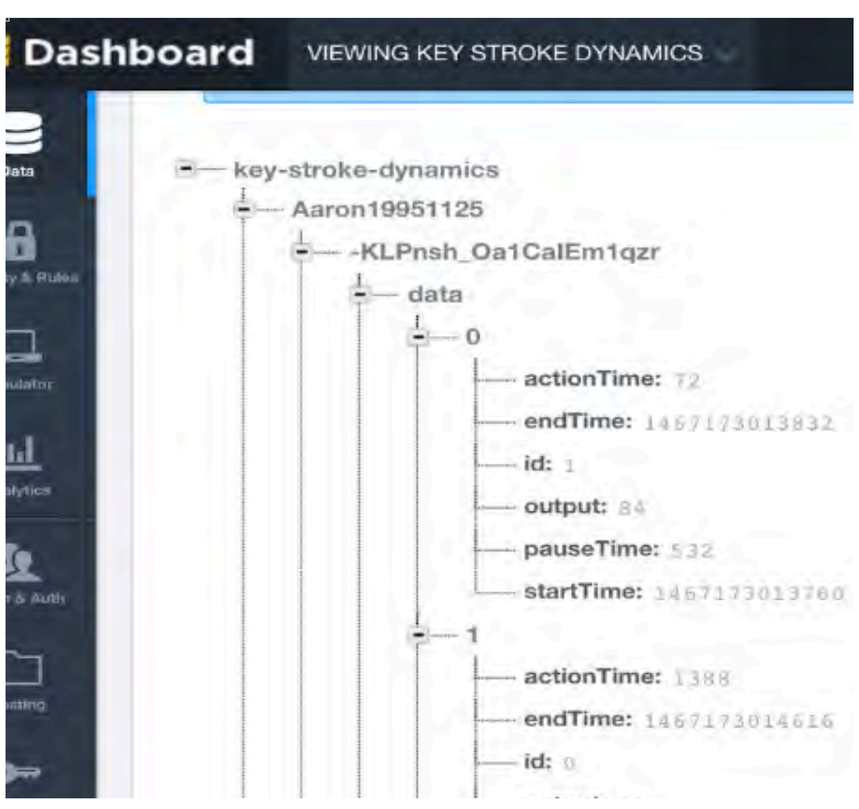


Fig. 2 Real-time database used to store data

Fig. 3 Interface for collecting data

Feature Analysis

We extracted different types of features from typists (Fig.4) and analyzed the effectiveness by how well each feature distinguishes typists.

Durational Analysis: Timing differences of pressing keys

Regional Analysis: Speed of typing across physical distance

Special Function Keys: Usage of Backspace, Shift, Caps Lock, and numbers

Combination: Pattern of consecutive keystrokes

Fig.4 Table of features

From Fig. 5, we can see how the features effectively distinguish typists.

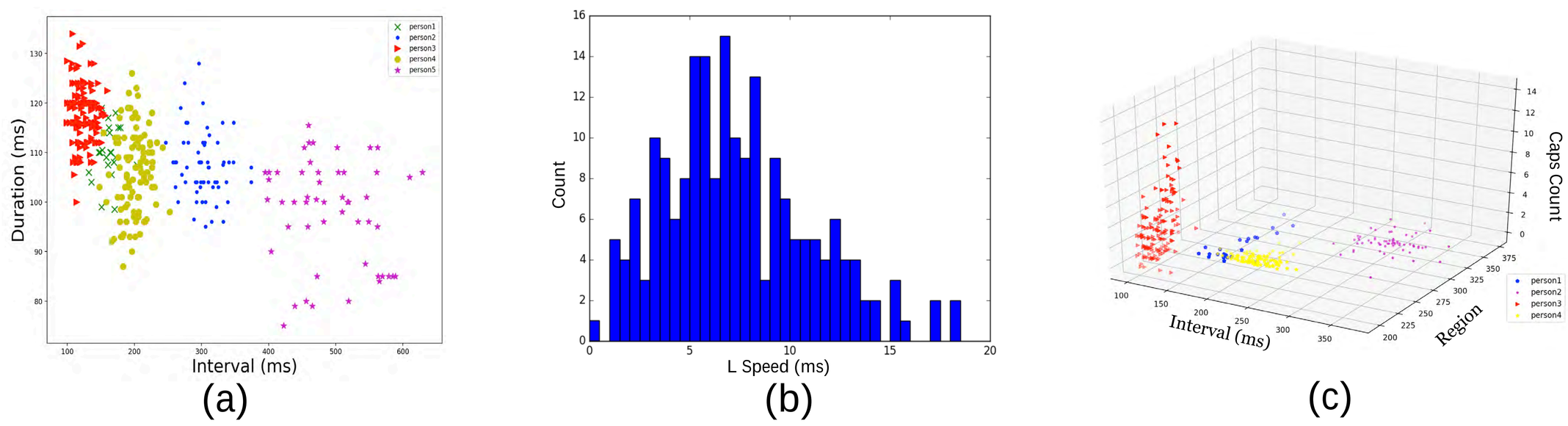


Fig.5 Analysis graphs: (a) Duration vs interval scatter (b) L speed histogram (c) Interval vs region vs caps 3D scatter

Results and Discussion

Results and Comparison

Using four performance indexes, we evaluate the performance of our features and the system.

Accuracy: $A = \frac{Entries\ predicted\ correctly}{Total\ entries}$

Recall: $R_i = \frac{Entries\ of\ i\ predicted\ correctly}{All\ i\ entries}$

Precision: $P_i = \frac{Entries\ of\ i\ predicted\ correctly}{Entries\ predicted\ as\ i}$

F-Score: $F_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}$

We created individual models for each typist. On average, these models reached an accuracy of 95.3%, precision of 96.4%, recall of 95.3%, and a F-score of 95.0%.

We used Equal Error Rate (EER) to validate the performance of the proposed method.

EER is the point where the False Alarm Rate and False Recognition Rate are at the same values, an equilibrium point of the authentication system. The lower the EER is, the higher the accuracy will be.

We compared our result with that of previous studies, resulting with a significantly lower EER (Fig. 6).

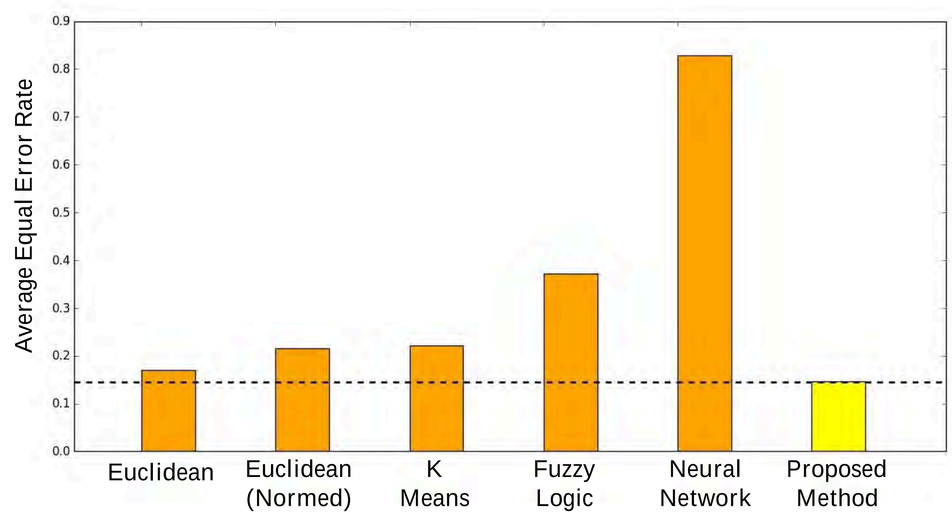


Fig.6 Comparison with previous studies

User Interface

The section of user interface introduces the process of which our authentication method would possibly work in real life.

We set up a server on a Python-based web server, Tornado, to connect to a typing interface.

The server obtains the typing data and sends it to our system, which compares the new data to the individual model. Our system then identifies the typist and returns the result.

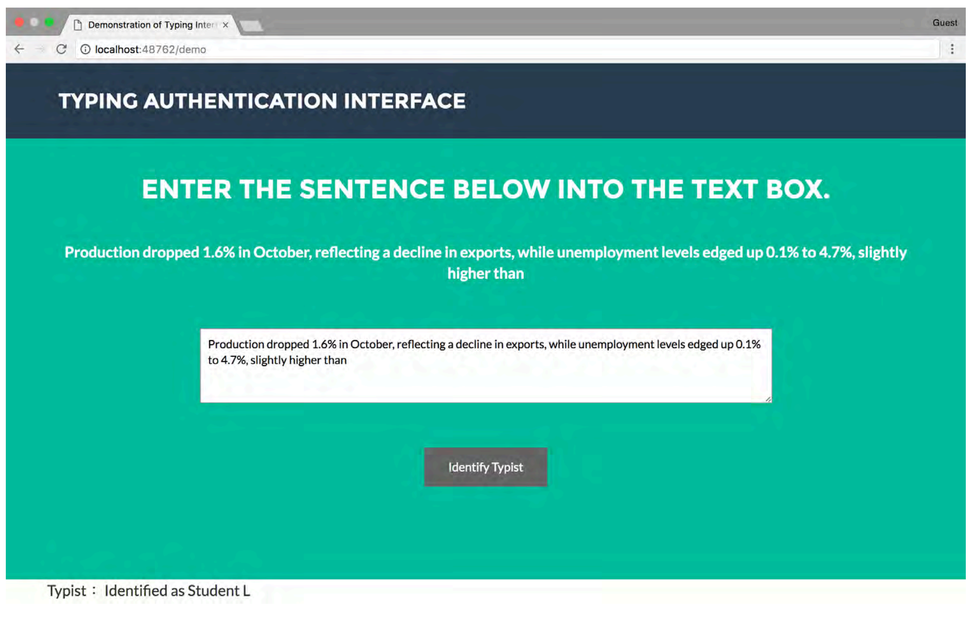


Fig.7 User identification interface

Discussion

Definitions of extracted features are listed in Fig.8. t_j represents the time that the key was pressed, and v_j represents the time of release.

Interval (I): $I_j = t_{j+1} - t_j$

Duration (D): $D_j = v_j - t_j$

Typing speed: Normalization of the physical distance of keys and measuring of the corresponding time needed to reach a certain distance

L/R Speed: The respective typing speeds of the left and right hand

Region: The typing speed of all fingers, each in its assigned region

Count (Backspace, Caps Lock, Shift): The average usage of the special function keys

Number: The average count difference of the number pad and the number row

Digraph/Trigraph: Interval of frequently used two/three consecutive keystrokes

Fig.8 Definition of features

In Figure 9 and 10, we evaluated the effectiveness of all features and their abilities to work with each other. To do so, we tested the performances of features both individually and holistically. The leave-one-out evaluation method tests a feature's effect on the overall performance. Different combinations of features target different user groups, and so we tested the results of different combinations.

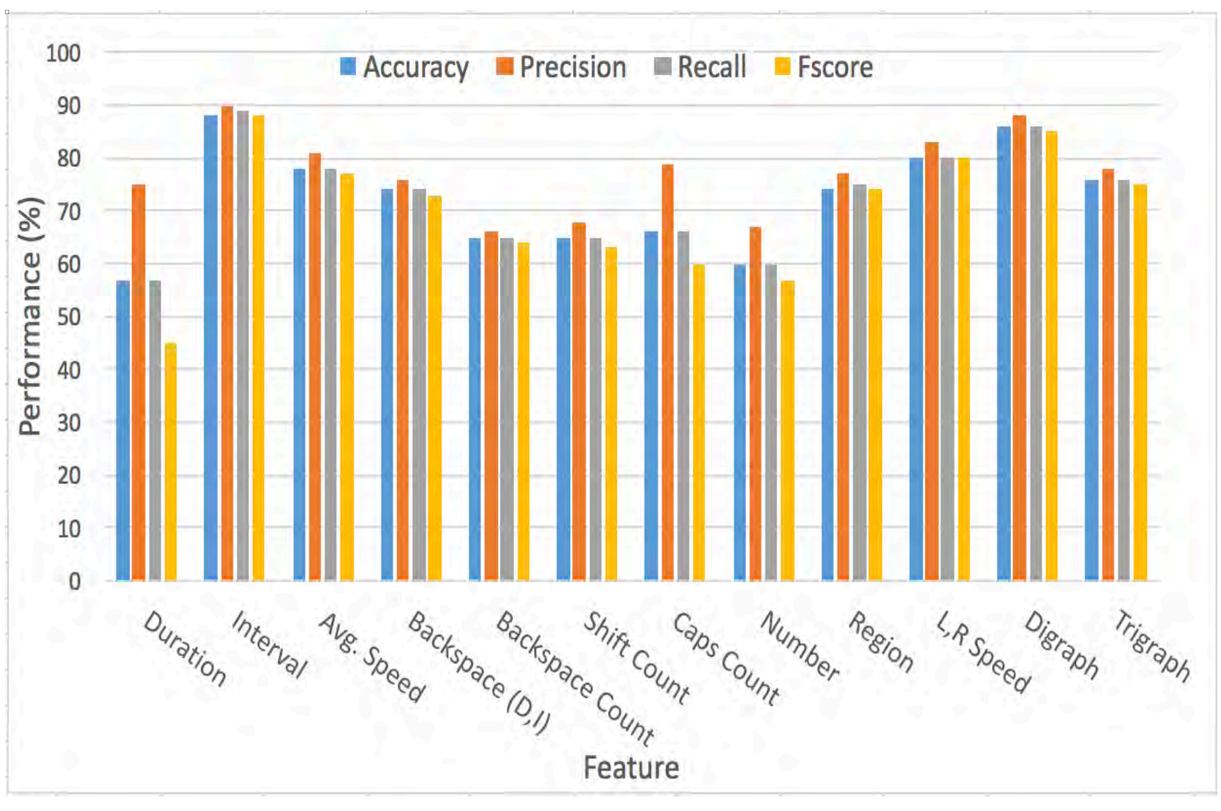


Fig.9 Individual feature performance

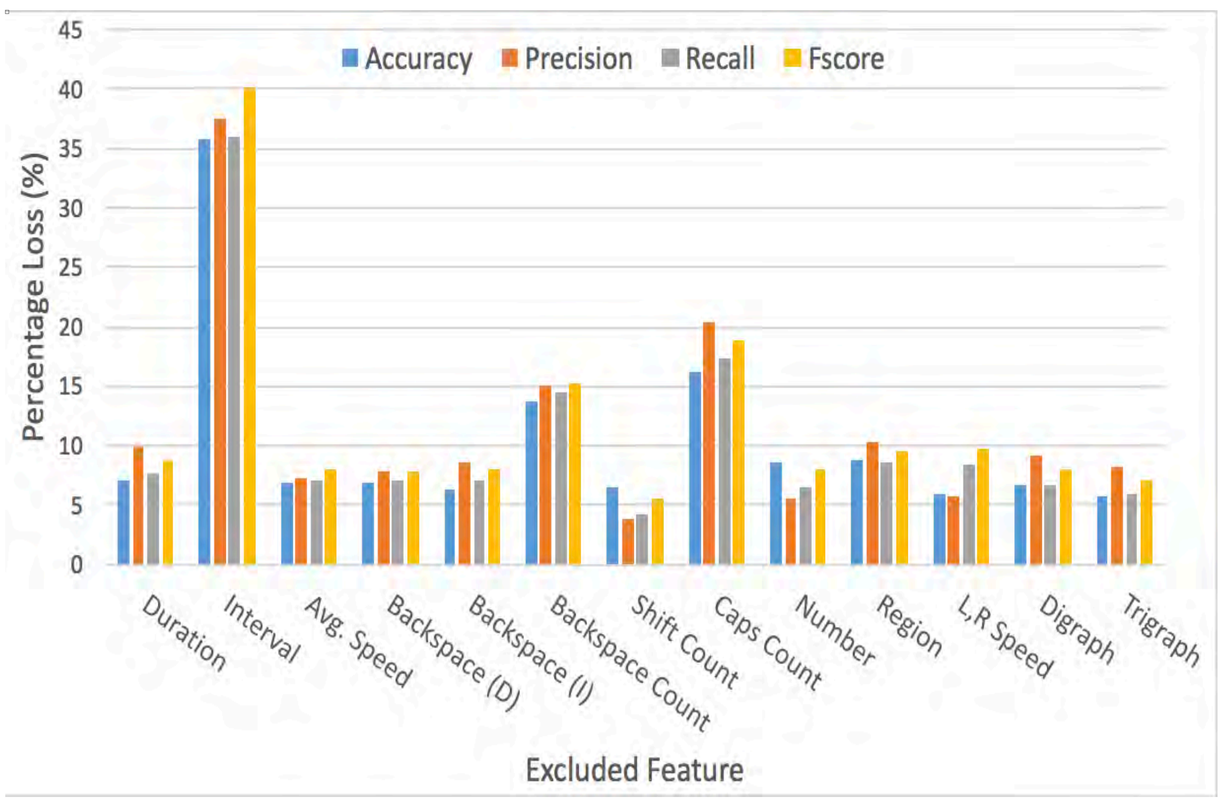


Fig.10 Leave-one-out performance

Conclusions

1. Satisfactory Performances

The method has a lower equal error rate than that of previous methods, and the individual models of typists reach F-score values of 95.0%.

2. Distinguishing Features

This project proposes a variation of novel features, which perform well and make up for the gaps of others, increasing overall effectiveness.

3. Extensive Applications

The authentication system can be easily applied to the security of banks, customs, and personal information.

References

[1] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[2] Kevin S. Killourhy and Roy A. Maxion. "Comparing Anomaly Detectors for Keystroke Dynamics," in Proceedings of the 39th Annual International Conference on Dependable Systems and Networks (DSN-2009), pages 125-134, Estoril, Lisbon, Portugal, June 29-July 2, 2009. IEEE Computer Society Press, Los Alamitos, California, 2009.