適用於入侵偵測之高準確度階層式分群演算法

曹偉駿*　游錦昌　林明孝
大葉大學資訊管理系
***E-mail: wjtsaur@mail.dyu.edu.tw**

摘要

入侵偵測系統發展至今約二十餘年，卻還是無法非常有效的應用在現今網路環境上，探討其原因，無非是因為現有的入侵偵測系統偵測率過低且誤報率過高，其中誤報率過高的問題更是讓管理人員拒絕使用入侵偵測系統的主因。為了提升入侵偵測系統之偵測率及降低其誤報率，本研究設計了一適用於入侵偵測之高準確度階層式分群演算法，讓群集分析可以更準確的分析出正、異常群集，藉此提昇入侵偵測系統之判斷能力。

關鍵字：入侵偵測系統、分群演算法、偵測率、誤報率

1. 前言

現有的群集分析(Cluster Analysis)理論，運用於分析入侵偵測的資料集時，對於正確的分析出正、異常行為並沒有太好的成效，導致運用群集分析技術所設計之特徵庫，並無法有效的提供給入侵偵測系統一個比對的基準。所以入侵偵測系統的偵測率一直無法提升，並不是在於入侵偵測系統本身的比對機制，而是在於沒有一個完善的特徵庫。

為了建構一個完善的特徵庫，本研究希望設計一個有效的群集分析機制，該機制除了可以適用於入侵偵測的資料集外，將特別重視群集內部的同質性(Homogeneity)，讓正常的群集中盡量不要包含異常的樣本，也就是群集在代表正、異常群集上擁有著高準確度。

本論文的結構如下：第二節簡介分群機制及不同類型的分群演算法。第三節提出本論文所設計之「高準確度階層式分群演算法」的設計理念及方法說明。第四節則為利用 DARPA Dataset 對該分群機制進行實驗，並就結果與其他分群演算法比較。最後再提出本研究之結論與未來發展方向。

2. 群集分析

群集分析可以將資料在「沒有預設的條件」下，依資料特性將資料區分成一個或一個以上的群集。在入侵偵測系統的應用上，會將群集分析出來的群集，判讀為正常或異常群集，再依正異常群集來建立比對特徵庫。一般典型的群集分析的流程如圖 1 所示，由圖 1 可發現，樣本(Patterns)經由特徵選取(Feature Selection)、相似度(Similarity)的取得、群組化(Grouping)，最後將資料標示為群集(Cluster)。經過群集分析的過程後，群集內部的資料具有高度的同質性(Homogeneity)，不同的群集則具有明顯的異質性(Heterogeneity)[1]。

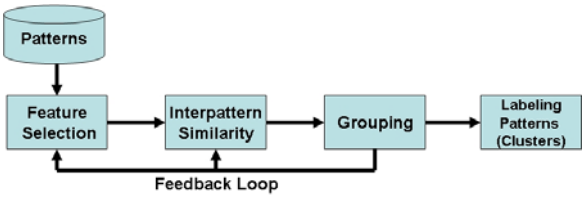

圖 1 群集分析流程圖

2.1 分群演算法

分群演算法(Clustering Algorithm)是基於群集分析的理論產生的工具，目的是為了使用電腦程式來實現群集分析技術。大致上可以將這些分群演算

法分為兩大類，分別為：分割式(Partitional)及階層式(Hierarchical)分群演算法，

● 分割式分群演算法：依據樣本間的距離，分割出群集的界線，如圖 2 所示，將每個群集中心點連線(實線部份)，再對每條連線做垂直平均分割，所得的虛線則為群集分割線，依據這些分割線即可將圖 2 的資料樣本分割為三個群集，分別為(1)、(2)、(3)。

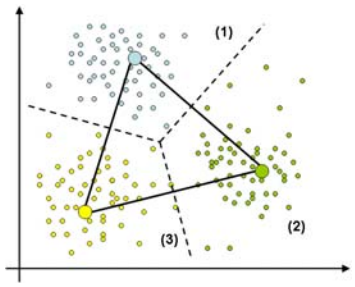

圖 2 分割式分群演算法概念圖

● 階層式分群演算法：透過凝聚(Agglomeration) [2]或分離(Divisive)的概念來進行分群。其概念，可透過圖 3 來解說。首先，在圖 3 (a)中有 A 到 I 九個樣本，分別分屬於三個不同的群集，而圖 3 (b)則是階層式分群演算法對這九個樣進行分群的過程，若使用「凝聚」的方法來實現階層式分群演算法，會先將這九個樣本視為單一的群集，然後再由上而下漸漸的合併為一個大群集；而「分離」的概念則剛好相反，是先將九個樣本視為一個大群群集，然後由下而上漸漸的將較小的群集區分出來。階層式分群演算法的終止條件是設立一門檻值，如圖 3 (b)中的虛線，當分群動作進行到設定的門檻值時，代表所劃分出來的群集結果為最佳的分群結果。

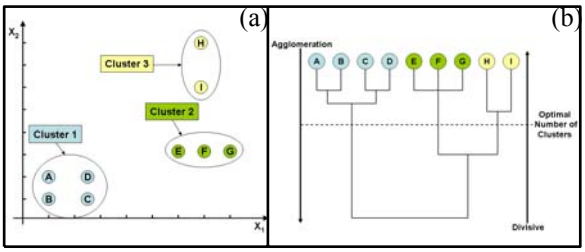

圖 3 階層式分群演算法解說範例圖

2.2 不同類型的分群演算法比較

一般來說分割式分群演算法在效能上會優於階層式分群演算法。這是因為使用分割式分群演算法時，樣本只需與每個群集中心進行比較，而階層式演算法的任一樣本卻需與其他全部的樣本進行比較。但也因為階層式分群演算法的任二樣本全都進行過分析比較，所以階層式分群演算法在群集準確度上是優於分割式分群演算法的，而且分割式分群演算法的群集通常都會受到離散值(Outlier)的牽引，造成群集偏移的情況。

本研究將此兩種不同類型分群演算法的相異處整理後陳列於表 1。由表 1 中可以發現，當研究是使用大型資料集，可以選擇使用分割式分群演算法，以獲得較佳的效率，但所得到的群集可能與真實狀況有著些許的誤差。若研究需要群集有較高準確度時，使用階層式分群演算法則是較佳的選擇。

表 1 分割式與階層式分群演算法比較表

比較項目	分割式	階層式
時間複雜度	K-means: $O(nkl)$	$O(n^2 logn)$
空間雜度	K-means: $O(k)$	$O(n^2)$
適用資料集	中、大型資料集	小型資料集
預設群集數	固定	不固定
離散值處理	經改良後可	可
群集準確度	較差	較佳

本研究是使用大型資料集來建構入侵偵測系統所需的特徵庫，使用分割式分群演算法，來取得較佳的效率似乎是合理的作法。但在入侵偵測的研究中，效率的重要性並不如有效的提升偵測率似。而且分群機制只是為了建構一完善特徵庫，因此本論文認為使用階層式分群演算法來替入侵偵測資料集進行群集分析，是一可行且有意義的研究。

3. 高準確度之階層式封包分群演算法

本研究所設計的高準確度之階層式封包分群

演算法，主要由樣本特徵選取(Feature Selection)、相似度(Similarity)計算方式及分群演算法的設計等三個階段來著手。

3.1 特徵選取

在實際檢驗過入侵行為的封包後發現，除了某些系統漏洞與 Probe 類型的攻擊外，幾乎大部份的入侵行為都需要使用一個以上的封包來進行，因此要能正確的判斷出入侵行為是需要考慮多個封包的組合，多個封包的組合即為「連線(Connection)」。

連線是由一個或多個封包所組成，在確認連線的過程中，本研究參考了 Zhang 及 Paxson [7]針對後門程式的連線定義，與 Paxson 及 Floyd [6]對廣域網路所進行的流量分析，將其整合並加以修正。最後以連線定向性(Connect Directionality)和封包接續抵達時間(Packet Interarrival Time)，做為本論文對連線的定義。而封包連續抵達的時間將使用 Paxson 及 Floyd 所求出的最佳設定 0.2 秒。

以圖 4 的範例來說明封包轉換程序，在圖中共包含了 P1、P2、P3、P4、P5、P6 及 P7 等七個封包，每個封包內都包含了來源及目的位置。首先，P1 封包抵達時新增連線 1，P1 屬連線 1 內之封包；而後 P2 封包抵達時會搜尋到前 0.2 秒(連線門檻值)內，有一相同目的及位置的封包，所以 P2 也隸屬於連線 1；而 P3 雖然在 0.2 秒前有封包 P2，但因為 P2 的來源及目的位置與 P3 並不相同，所以新增連線 2，且 P3 屬於連線 2 內的封包。以此類推，最後該 7 筆封包資料會被轉換為 4 筆的連線記錄。

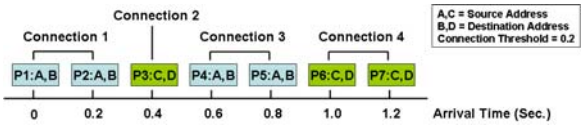

圖 4 封包轉換連線解說範例圖

圖 5 連線特徵與封包特徵關係圖

3.2 相似度的計算

入侵偵測的樣本特徵通常都是非量化特徵，所以使用傳統的空間距離公式來計算其樣本相似度是會影響到分群的結果的，因欲將質化特徵進行量化動作時，依靠的都是研究者的主觀定義，而這些主觀定義的特徵值，通常都會帶有些許的誤差。同時在且使用空間距離公制時，還需進行正規化(Normalization)，而在正規化過程中，樣本特徵所代表的關係及意義也會被影響或破壞。

Oh及Kim [5]在2004年提出一種不同於以往空間距離的概念來計算樣本相似度，其算式如公式(1)所示，其中 S_i, S_j 代表兩連續資料樣本，$|E_i \cap E_j|$ 代表兩樣本中任一對項目(item)交集的個數，而 $|E_1|$ 及 $|E_2|$ 代表該樣本中任一對相目的總數。舉例來說：若 $S_1=\{A,B,C,D\}$、$S_2=\{A,C,D,E\}$，則 $E_1=\{AB,AC,AD,BC,BD,CD\}$，$|E_1|=6$，$E_2=\{AC,AD,AE,CD,CE,DE\}$，$|E_2|=6$，$E_1 \cap E_2=\{AC,AD,CD\}$，$|E_1 \cap E_2|=3$，而 S_1 與 S_2 的相似度 $Sim(S_1,S_2)$ 為1/2。

$$Sim(S_i,S_j)=\frac{\frac{|E_i \cap E_j|}{|E_i|+|E_j|}}{2} \quad (1)$$

使用Oh及Kim的相似度計算方式，可以避免傳統歐基理德距離在資料量化及正規化下的風險，而且依該篇論文最後的實驗結果，將該相似度計算方法應用於階層式分群演算法時，也有著較使用歐基理德距離更佳的分群結果，但因為這個方法並沒有使用樣本空間的概念，因此是無法使用於分割式分群演算法。

3.3 高準確度階層式分群演算法

高準確度階層式封包分群演算法其概念可透過圖 6 來解說。所有的樣本在演算法初始階段(Level 0)都被視為一個完整的群集*Cluster 0*，而之後每一階層的分群都是使用分離的概念來進行，並使用代號來辨視群集，如$Cluster_{l,p,k}$即代表該群集位於l層，其父節點為$(l-1)$層的第p個群集，而該群集是該階層的第k個群集。同時，在每一個階層內所產出的

Outlier集合都將視為異常行為。

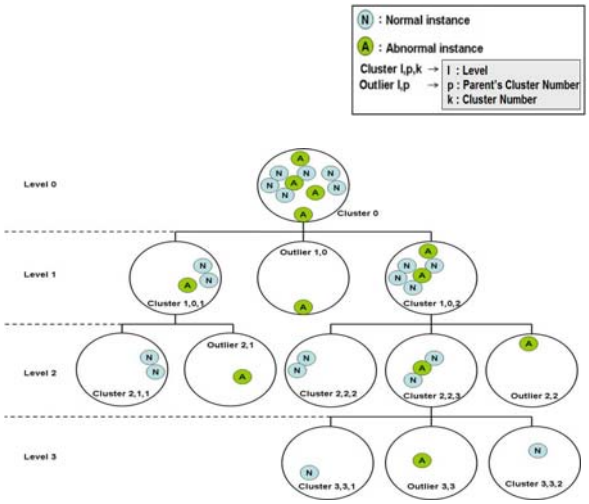

圖 6 高準確度階層式分群演算法

高準確度階層式封包分群演算法如表 2 所示，其中TrainingFunction為分群的主程序，首先需輸入分群用的資料集D、相似度門檻值ST、終止分群的門檻值ET及每階層門檻值的遞增值(03 行)，接著對任兩樣本計算其相似度，若其相似度大於相似度門檻值時，則該兩樣本視其為相同群集(05 至 13 行)，而當某一樣本無法找到其他鄰近樣本(near neighbor)時，該樣本則會被歸類至$Outlier_{l,k}$ (15 行)。當某一階層進行完群集分析後，若該階層的相似度門檻值尚未到達終止計算的門檻值時，則繼續下一階層的分群動作(19 至 22 行)。

表 2 高準確度階層式分群演算法

01	class **TrainingFunction**{
02	int **k;**
03	TrainingFunction(Dataset **D**, Sim_Threshold **ST**, End_Threshold **ET**, Decrease_Threshold **DT**, int **l**, int **p**){
04	if (ST > ET){End;}
05	for (int i = 0 ; i < n-1 ; i++){
06	if (S_i in any cluster){
07	k = cluster number of S_i ;
08	}else{
09	k = max(cluster number of h) +1; }
10	for (int j = i+1; j < n ; j++){
11	if (Sim(S_i , S_j) > ST/2){
12	$C_{lpk} \leftarrow S_j$;}
13	}
14	If (C_{hfk} is empty){
15	$outlier_{lk} \leftarrow S_i$;
16	}else{
17	$C_{lpk} \leftarrow S_i$;}
18	}
19	If (ET < ST){
20	for all cluster of h level{
21	Training Function(C_{lpk} ,(ST+DT), ET,DT, (h+1) ,(cluster.k));}
22	}
23	}
24	}

4. 實驗結果與分析比較

本研究的實驗是使用 DARPA 資料集做為輸入來源。先以 Week 2 的記錄來訓練群集，得到群集結果，再將 Week 4 及 Week 5 的資料來與群集結果進行比對，以得到偵測率及誤報率

4.1 群集訓練與結果

在訓練的過程中，分別以值 0.05、0.1 及 0.2 作為相似度遞增值，當使用遞增值 0.05 來建構群集時，在某些階層中並無法分析出異常的 Outliers 集合；當遞增值設為 0.1 時，每階層的群集可以有效的分離，也可以將某些異常行為，正確的分至 Outliers 集合；當遞增值設定為 0.2 時，雖然每一層也都可以有效的區分出 Outliers 集合，但此時的 Outliers 集合內，所包含的異常行為數低於遞增值設為 0.1 時所包含的異常行為數。因此，本論文最後將相似度的遞增值設定為 0.1。

當相似度遞增值設為 0.1 時，在第一層的分群結果中，共分成了兩個群集及一個 Outlier 集合，

Outlier 1,0 中則包含了五種實際的入侵行為：back、eject、neptune、pod 及 ps；在第二層的分群結果中則包含了五個群集及 *Outlier 2,1*，其中包含了 Crashiis、Dict、Ftp-write、Guest、Httptunnel、Mailbomb 及 Multihop 等攻擊；而第三層的分群結果包含了 9 個群集及一個 Outlier 集合，*Outlier 3,1* 中包含了：Format、Imap、Portsweep、Syslog 及 Perlmagic 等攻擊；第四層則又將群集切分為 29 個群集及 2 個 Outler 集合，其中 *Outlier 4,1* 中包含 Loadmodule、Satan 及 Warezmaster 攻擊，而 *Outlier 4,9* 則包含了 Land、Neptune、Nmap 及 Smuf 攻擊。

在第五層中群集已被分割得非常細小，群集內資料已失去可讀性，而且此時的 Outlier 集合不但已無內含入侵行為，反而還包含了許多的正常行為，因此在本研究中，將最後的分群門檻值設於 0.4，此時的分群結果可從 Outlier 集合中發掘八成以上的攻擊，而且 Outlier 集合內部也僅有不到七十筆的正常連線資料。

4.2 測試結果

採用 DARPA 資料集 Week 4 及 Week 5 的資料與本論文的群集結果進行比較，比較結果如表 3。

表 3 實驗結果一覽表

入侵類型	總入侵數	正確警報	錯誤警報	偵測率	誤報率
Probe	44	36	1424	81.81%	0.00069%
DoS	58	56		96.55%	
U2R	32	27		84.37%	
R2L	61	54		88.52%	
Data	6	3		50.00%	
Total	201	180	1424	87.56%	0.00069%
註：總連線數：2,126,307 正常連線數：2,126,106 (=2,126,307 － 201)					

檢視其結果，DoS 類型的入侵行為可偵測高達 96.55%，這是因為 DoS 類型的攻擊無論是傳送大量封包或特殊指令封包，在本分群機制的設計下都是可以檢視出來的。而 Probe 類型的入侵行為的偵測率只達 81.81%，因為有些入侵行為是由一台主機分散探測多台主機，所以在 Connection 的機制下，將會視其為多個正常行為，因而無法偵測，如 Ipsweep 就是屬於這種類型的攻擊。另外 U2R、R2L 及 Data 類型的攻擊也因為有些攻擊的差異性要檢視封包 Payload 才能發覺，如 Sqlattack 入侵需檢視封包內容中的 URL 才得已偵測。

在誤報率的部份，因為本機制所判斷出的群集具有高度的準確度，所以在 Outlier 中顯少有正常的行為。也因為正常與異常行為合適的分割，造就了本研究中顯少的誤報警訊，使本研究的誤報率低於 1.00%。

4.3 優越性比較

在優越性比較方面，本論文與同樣探討入侵偵測分群演算法的相關論文來進行比較。首先，就 H-means+及 Y-means[3]演算法進行實作，並使用本論文相同的樣本特徵及相同的 DARPA Week 2 資料進行訓練，並判讀出正、異常群集，再以 Weeks 4,5 的資料進行測試比對，得到的結果如表 4 所示。但最後的結果較 Guan 等人[3]所得的 H-means+偵測率 79%、Y-means 偵測率 86%為低，而誤報率則較高 (Guan 等人所得的 H-means+誤報率為 1%、Y-means 誤報率為 1.53%)，這可能是因為樣本特徵的取樣不同，及對正異常群集的認定不同。另外該論文的實驗資料，僅使用部份的資料集，而不是使用完整的資料集，也可能是造成差異的原因。

表 4 與其他分群機制比較一覽表

	偵測率	誤報率
H-means+	65.48%	12.39%
Y-means	64.32%	10.33%
本機制	87.56%	0.00069%

另外，Liu 等人[4]的研究是將基因分群演算法應用於入侵偵測上，該機制先採用 Nearest Neighbor 分群演算法進行分群，再利用基因演算法的「淘

汰」、「交配」及「突變」等方法來調整群集。雖然該機制成功得利用基因分群演算法完成了網路封包的分群，但在分群的過程中需控制 2 個權重值及淘汰、交配與突變等三個門檻值，在實際的應用上當資料樣本不同時，此 5 種變數值皆需重新測試、評估多次，才能得到最佳的分群結果。而本論文僅需設定終止群集的門檻值及每階層遞增的門檻值，由上而下逐層分群，若群集已過度分群，也只需捨棄該階層之分群結果即可，因此在實際的應用上，本論文的群集建置成本是低於 Liu 等人所提出的基因分群機制。而 Liu 等人在實驗的方面是使用了自製的資料集來進行分析比較，其最後所得之偵測率為 61%，誤報率為 0.4%，但因其並非使用 DARPA 資料集，所以無法將本論文的實驗結果直接與其比較。

在優越性比較中，本論文比較了 Guan 等人及 Liu 等人所針對入侵偵測設計之分群演算法，無論是在偵測率與誤報率的結果上，以及在實際建置群集的成本上，本論文皆有著一定的優勢，也因此確認本論文所提出的「適用於入侵偵測之高準確度階層式分群演算法」，的確具有相當的優越性。

5. 結論

本論文提出了一「適用於入侵偵測之高準確度階層式分群演算法」，該演算法從分析樣本特徵開始著手，藉由分析封包得到封包特徵，同時發現若以連線來計算可以得到更多的入侵行為特徵。雖然本研究所提演算法在效率上雖不及於目前現有的分群演算法，但是應用於入侵偵測時，可以有效的分割正、異常行為，主要的原因在於選擇了合適的樣本特徵，而且利用階層式的觀念可以有效的將攻擊行為列為 Outlier 集合。最後在比較了一些應用於入侵偵測上的特殊分群演算法結果之後，本論文所得的偵測率的確是優於目前的分群演算法，而且在誤報率也優於其他的分群演算法。

另外，本論文所分析的階層群集，實際上還可應用於封包分類上，例如在本機制的第一階的群集結果中，即可將封包分為 IP 及 ICMP 兩大類。若可有效的利用每一階層的群集關係來進行封包分類，除了可以減少封包等待的時間外，還可有效的加速入侵偵測系統的比對時間。

在未來發展方面，針對同一來源位置與多個目的位置的入侵攻擊，應藉由連線間彼此的關係來進行分析，即在一段時間內的同一來源與不同目的位置的連線關係，並建立該樣本特徵來進行比較，如此應可補足本論文無法偵測 Ipsweep 入侵類型的缺失，藉此也可再提升本論文的偵測率。

參考文獻

[1] M. Berry and G. Linoff, Data Mining Techniques: for Marketing, Sales, and customer support, Published by Arrangement with Wei Keg Publishing Co., 1997.

[2] K.C. Gowda and G. Krishna, “Agglomerative Clustering Using The Concept of Mutual Nearest Neighborhood,” Pattern Recognition, Vol. 10, No. 2, pp. 105-112, 1978.

[3] Y. Guan, A. Ghorbani and N. Belacel, “Y-Means: A Clustering Method for Intrusion Detection,” Canadian Conference on Electrical and Computer Engineering, pp. 1083-1086, 2003.

[4] Y. Liu, K. Chen, X. Liao and W. Zhang, “A Genetic Clustering Method for Intrusion Detection,” Pattern Recognition, Vol. 37, No. 5, pp. 927-942, 2004.

[5] S.J Oh and J.Y. Kim, “A Hierarchical Clustering Algorithm for Categorical Sequence Data,” Information Processing Letters, Vol. 91, No. 3, pp. 135-140, 2004.

[6] V. Paxson and S. Floyd, “Wide-Area Traffic: The Failure of Poisson Modeling,” IEEE/ACM Transactions on Networking, Vol. 3, No. 3, pp. 226-244, 1995.

[7] Y. Zhang and V. Paxson, “Detecting Backdoors,” Proceedings of the 9th USENIX Security Symposium, pp. 157-170, 2000.