楊欣哲、黃小玲（2020），『雲端環境下設計混合式泛濫攻擊防禦機制之研究』，*中華民國資訊管理學報*，第二十七卷，第二期，頁 205-234。

雲端環境下設計混合式泛濫攻擊防禦機制之研究

楊欣哲*
東吳大學資訊管理學系

黃小玲
東吳大學資訊管理學系

摘要

雲端運算的虛擬化技術是透過網際網路把計算資源量化後以使用量付費的方式提供給使用者。然而多租戶和共享資源雖是特點，但也隱含資安的風險。在攻擊事件中，造成較嚴重的後果又較難防禦之一就是泛濫攻擊（flooding attack）。為此，本文提出一種基於特徵篩選結合隨機森林分類模型以偵測混合式泛濫攻擊機制，稱為 HFADS。此機制主要分為三個模組：(1)資源監控模組：此模組主要的功能在於監控 CPU 的使用率，當它低於門檻值時以便觸發命令行工具 TShark 中執行程序擷取網路封包；(2)資料特徵篩選模組：此模組即利用 TShark 擷取的封包，匯入資料探勘軟體工具 Weka 後，進行三種特徵的篩選，去除不相關的特徵並選出比重高的特徵後，再由機器學習評估；(3)機器學習評估模組：此模組則使用隨機森林分類模型以偵測 UDP、ICMP、HTTP 此三類常見的泛濫攻擊。根據上述三個模組進行模擬實驗，提出精確率、召回率、總正確率和平均處理時間等四種關鍵績效指標並與 Alkasassbeh 等（M.A.）所提出的隨機森林演算法以偵測混合式泛濫攻擊進行模擬比較，實驗結果顯示本文所提出的 HFADS 機制比 M.A.在偵測網路層 UDP、ICMP 和應用層 HTTP（使用 TCP）等通訊協定上，均有較佳的精確率與召回率外，以及其總正確率為 99.98%提升了 2%和平均處理時間為 65.34 秒亦改善了 5.14%。

關鍵詞：雲端運算、HFADS、混合式泛濫攻擊、特徵選擇、機器學習

* 本文通訊作者。電子郵件信箱：sjyang@csim.scu.edu.tw
2019/12/11 投稿；2020/02/11 修訂；2020/02/20 接受

Yang, S.J. and Huang, H.L. (2020), 'Design a hybrid flooding attacks defense scheme under the cloud computing environment', *Journal of Information Management*, Vol. 27, No. 2, pp. 205-234.

Design a Hybrid Flooding Attacks Defense Scheme under the Cloud Computing Environment

Shin-Jer Yang*
Department of Computer Science and Information Management, Soochow University

Hsiao-Lin Huang
Department of Computer Science and Information Management, Soochow University

Abstract

Purpose－Cloud computing is integrated lots of computing resources which are provided to users over the internet on a Pay-As-You-Go mode. While multi-tenants and resources sharing are its advantages under the cloud computing environment, it also brings new risks in information security. One of the more difficult consequences of an attack is a flooding attack. Hence, this paper proposes a new scheme: HFADS for detecting and resolving the hybrid flooding attacks.

Design/methodology/approach－Based on the existing DDoS detection technology, this paper enhances the Alkasassbeh et al.'s (M.A.) Random Forest algorithm and combines feature selection and random forest classification to propose a new scheme HFADS for detecting and resolving the hybrid flooding attacks. The proposed HFADS scheme is mainly divided into three modules: (1) Resource Monitor Module, (2) Data Feature Selection Module and (3) Machine Learning Evaluation Module:

Findings－Based on three modules in HFADS, we did perform some simulations to analyze and compare with Alkasassbeh et al. (M.A.) to detect hybrid flood attacks according to four key performance indicators including precision rate, recall rate, total accuracy rate and average processing time. The final experimental

* Corresponding author. Email: sjyang@csim.scu.edu.tw
2019/12/11 received; 2020/02/11 revised; 2020/02/20 accepted

results indicate that HFADS has better precision rate and recall rate than the method of M.A. for detecting the protocols: UDP, ICMP and HTTP (with TCP). In addition, the HFADS can obtain 99.98% and 65.34 Seconds in total accuracy rate and average processing time, respectively, thus to increase the 2% in total accuracy rate and shorten the 5.14% in average processing time than the M.A. method.

Research limitation/implications – The hybrid flooding attacks are assumed to be exhausted the network bandwidth and the computing resources of the server, causing the server to fail to provide services due to heavy workload and may affect the service. Other servers in the same infrastructure cause unpredictable losses.

Practical implications – The HFADS scheme is mainly divided into three modules: (1) Resource Monitor Module: This module is to monitor the CPU utilizations to trigger script procedure in a command-line tool TShark in which is to capture network packets, when the CPU usage ratio is lower than threshold value. (2) Data Feature Selection Module: This module uses the network traffic drawn by TShark and imports it into data mining tool Weka to perform three feature selections, remove irrelevant features and select features with high proportion, and then evaluate by machine learning. (3) Machine Learning Evaluation Module: This module utilizes a random forest classification model to detect three common types of flooding attacks: UDP, ICMP, and HTTP.

Originality/value – The proposed HFADS scheme is to combine feature selection and random forest classification to enhance the method of M.A. for detecting and resolving the hybrid flooding attacks. The final experimental results prove that HFADS are much better and more efficient than the method of M.A. in terms of precision rate, recall rate, total accuracy rate, and average processing time.

Keywords: cloud computing, HFADS, hybrid flooding attacks, features selection, machine learning

壹、緒論

雲端運算環境下攻擊者利用通訊協定或系統的缺陷，以海量的資訊流發動多種不同的攻擊。對於被攻擊的目標來說，需要面對不同的通訊協定、不同的來源之分散式攻擊，其分析、反應和處理的成本就會大大增加。

一、研究與動機

雲端運算技術是整合數千台電腦的系統資源以虛擬化的技術把資源利用最佳化及可量化後的型態再透過網路提供給使用者，而使用者可根據特定需求選擇使用 IaaS、PaaS 及 SaaS 的模式與其他租戶共享服務。然而，多租戶和共享服務雖是特點，但卻是資訊安全上的一大挑戰。近年來網路攻擊事件頻傳，在網路攻擊事件中，造成最嚴重的後果也是較難防禦的就是分散式阻斷服務（distributed denial of service; DDoS）的攻擊。攻擊者藉由入侵大量安全防護較薄弱的電腦並植入可被控制的木馬程式做為發動攻擊的跳板，這些大量被控制的殭屍電腦形成殭屍網路，一旦接收攻擊者的命令，將會同時間對同一目標發動特定類型的攻擊。龐大的流量傳遞到受害伺服器，導致伺服器負荷過重，以致快速耗盡伺服器的網路及系統資源，其中較難防範的又以泛濫攻擊（flooding attack）莫屬。

雲端運算所提供極具彈性及可擴充的機制，是目前全球在 IT 架構上強力採用的解決方案，國際調研機構 IOD Cloud Technologies Research 與 Cloudify 指出雲端運算已成為企業主要的架構模式。相較於傳統的網路封閉架構，雲端運算則是一個共享的計算資源之環境，無論是外部系統的漏洞或者內部系統的惡意使用者，皆能輕易擷取共享環境裏的資料。雲端中存有大量組織和企業敏感和機密的數據資料，若因遭受泛濫攻擊而當機或斷線時，組織之業務將會瞬間停擺，對組織或企業來說是無法估計的損失。

二、研究目的與範圍

隨著人工智慧的興起，機器學習已經開始被運用在入侵偵測的機制上並成為熱門的研究方向。機器學習透過資料不斷的訓練和測試，從資料中得到樣本，找到運行規則而辨識出運作模式，並用來做偵測。為了偵測網路層和應用層的攻擊，採用了三種機器學習的分類模型，針對由網路流量收集而來的封包資料集進行學習和訓練，並用來偵測網路層和應用層的攻擊（Alkasassbeh et al. 2016）。雖然有不錯的偵測效果，但由於資料量龐大以致在偵測攻擊過程中耗費多時，影響了偵測結果和處理時間。為了提升效能和偵測結果，本文提出一個以特徵篩選機

制做為機器學習法的前置處理，再以隨機森林分類模型分別對 UDP、ICMP、HTTP 這三類攻擊建立模型的混合式泛濫攻擊防禦機制，稱為 HFADS（hybrid flooding attacks defense scheme）。

所提出的 HFADS 機制與 Alkasassbeh 等（M.A.）方法主要不同之處在於藉由隨機森林分類前多採用特徵的篩選之步驟。由於每一特徵是一個或一個以上封包屬性的組合，為了構建隨機森林模型而需從網路流量中篩選相關封包特徵，因此特徵選取的好壞對偵測系統的偵測和效能有很大的影響。再者，在入侵偵測的學術研究領域裡，大部份的文獻都只針對單一攻擊類型提出防禦或偵測機制，較少有單一機制同時偵測不同通訊協定在不同網路層的泛濫攻擊。總而言之，本文主要的研究目的與範圍內容包含下列幾項：

1. 泛濫攻擊是 DDoS 演進的攻擊方式，傳統的 DDoS 偵測方法較難做到主動監控或即時偵測，針對此一型式的攻擊，本文結合特徵的篩選機制和隨機森林分類模型以提出一個偵測不同的通訊協定之泛濫攻擊防禦機制：HFADS。
2. 本文所提出的 HFADS 防禦機制透過 Python 中所提供之資源監控模組（Keep_monitor_CPU）以監測 CPU 使用率來觸發命令行工具 TShark，通過指令擷取網路封包並導出到文件，然後擷取並過濾網路封包以做為防禦攻擊的基礎。
3. HFADS 方法在有限的計算資源下，可以改善泛濫攻擊的平均處理時間和提高判斷異常流量的總正確率。
4. 提出四個關鍵績效指標：精確率、召回率、總正確率和平均處理時間，將 HFADS 與 M.A.偵測方法使用隨機森林偵測方法進行模擬實驗，所得出的模擬結果進行分析後，以證明本文所提出的 HFADS 在防禦泛濫攻擊的關鍵績效指標有顯著的提升。

三、論文章節架構說明

本文於第壹節緒論說明泛濫攻擊偵測機制的研究背景與動機，以及研究範圍與目的。第貳節文獻探討除了說明雲端運算的便利和雲端安全的重要性外，也說明泛濫攻擊的類型，並列舉當前所做的偵測對策並闡述泛濫攻擊的行為和特徵及特徵篩選方法的運作原理與隨機森林的建構過程。第參節為 HFADS 之模組運作與演算法設計，說明所提出的 HFADS 方法之模組運作原理與其功能，並設計其演算法。第肆節是模擬實驗建置、定義關鍵績效指標以及模擬實驗後的結果分析。第伍節為結論，闡述本文之研究成果和貢獻以及未來持續研究的方向。

貳、文獻探討及相關研究

本章主要透過國內外文獻蒐集，除了闡述雲端運算的便利和安全外，同時探討泛濫攻擊的相關類型，包括攻擊的定義及行為，以協助研究者更了解泛濫攻擊的特性。

一、雲端運算與安全

雲端運算有 IaaS、PaaS 與 SaaS 的三種服務層次和公有雲、私有雲、混合雲和社區雲的四種部署模式，以及自助式隨需服務、廣泛式網路存取、共享式資源使用、快速且靈活部署、量測式服務選用的五種服務特徵（Mell & Grance 2011）。架構即服務（Infrastructure as a Service; IaaS）是將伺服器、儲存及網路資源等硬體設備租借出去，讓使用者可以依據需求租用，節省了採購、建置及維護硬體的費用，如 Amazon AWS、IBM SmartCloud Enterprise 等都是 IaaS 著名的應用之一。平台即服務（Platform as a Service; PaaS）是在雲端提供一個有平台和工具的開發環境給軟體開發者，讓軟體開發者可以自訂或測試所開發的應用程式，無需考慮擴充性、相容性或容量等問題，減少了維護開發平台及工具的成本，如 Bluemix、Force.com 等。軟體即服務（Software as a Service; SaaS）則是提供使用者透過瀏覽器直接登入並使用線上軟體，顛覆以往需要把軟體安裝在每台機器上的使用方式，減少了使用者購買、安裝及維護軟體的時間與成本，如 Facebook。

提供安全和可靠的雲端運算環境是雲端服務供應商對於將使用雲端服務的企業和組織所要面臨的首要挑戰。雲端安全聯盟（Cloud Security Alliance; CSA）在 2016 年的報告中提出雲端運算有十二個安全威脅：資料洩露、身份及憑證和訪問管理不足、不安全的應用程式介面、系統漏洞、帳戶被盜、惡意的內部員工、進階持續性威脅、資料遺失、評鑑不足、雲端運算服務的濫用、阻斷服務、系統共用的隔離問題。

二、泛濫攻擊類型及其說明

全球已進入數位化時代，網際網路的存取和線上應用程式與服務的依賴性與日俱增，相對地遭受攻擊或入侵的風險也大幅提升，其中以遭受分散式阻斷服務（DDoS）攻擊的損傷最為嚴重，至今仍無法提出真正解決的機制，如圖 1 所示。

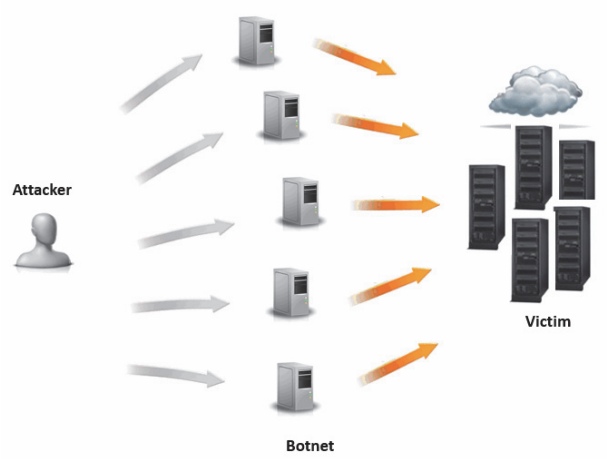

圖 1：DDoS 攻擊示意圖

TechRepublic（2018）的報告中，在 2018 年 3 月，著名的 GitHub 即遭遇 1.35Tbps 的 DDoS 攻擊，造成 GitHub 的服務停止 5 分鐘以上。在雲端環境一旦遭受僵屍電腦的攻擊，受害端主機不但要面對不同的通訊協定、不同來源的龐大攻擊流量，更要在有限的時間和資源內做分析、反應與處理，偵測也就更不容易。

雲端上分散式阻斷服務攻擊的對策分為三種類型：攻擊防禦、攻擊偵測及攻擊緩解（Somani et al. 2017）。DDoS 的攻擊防禦是在這些可疑的攻擊者請求開始影響伺服器之前就被過濾或丟棄。然而，攻擊防禦機制會造成伺服器過大的開銷。攻擊偵測則是根據其預先設定的監測指標來判斷是否符合攻擊行為。由於判斷攻擊的方式是以預先定義好的攻擊行為為基礎，因此，無法有效偵測新型攻擊。攻擊緩解是為了讓伺服器保持運作的狀態，當有攻擊行為發生時，立即進行流量引導規劃。但是緩解設備的成本隨著攻擊規模的持續增加而倍增，以致影響到整體的 TCO（total cost of ownership）。為此，本文認為雖然攻擊偵測機制無法有效即時偵測新的攻擊行為，但相較於攻擊防禦與攻擊緩解，攻擊偵測的對策不但能在有限的計算資源情況下，減少在遭受泛濫攻擊時的損失，而且不會增加企業的營運成本，只需時常更新攻擊行為模式即可確保使用者的權益。

網路報告中（TechRepublic 2018）指出，當攻擊者使用網路上兩個或以上被攻陷的電腦形成「殭屍網路」並向特定的伺服器同時發動大量的服務請求或傳送大量封包，堵塞頻寬或消耗伺服器資源，迫使該伺服器無法提供正常服務時稱為泛濫阻斷服務攻擊，此攻擊是目前 TCP/IP 協定上常見的攻擊方式之一，其資料流程如圖 2 所示。其攻擊特點在於攻擊者可針對每個階層提出和一般使用者一樣的服務請求，所以很難從流量分辨是正常或是惡意的封包。

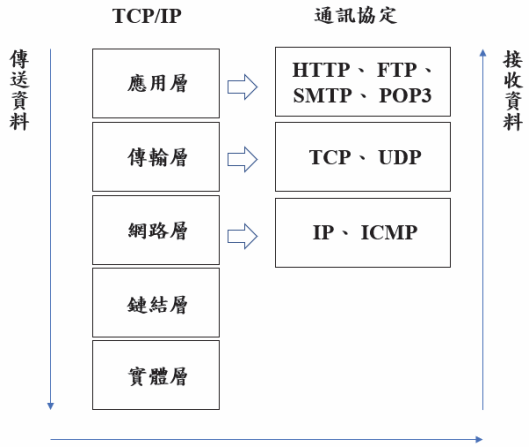

圖 2：TCP/IP 資料流程架構圖

在 35 Types of DDoS Attacks Explained（JavaPipe 2018）文中指出在泛濫攻擊中根據攻擊的方式可分為二種類型，分別是頻寬消耗型和資源消耗型，如圖 3 所示。由於一般常見的泛濫攻擊皆發生在網路層、傳輸層和應用層，本文針對頻寬消耗型在傳輸層為 UDP 協定，透過網路層 ICMP 協定的泛濫攻擊和資源消耗型在應用層為 HTTP 協定透過 TCP 協定的泛濫攻擊進行研究、分析並建立偵測機制。

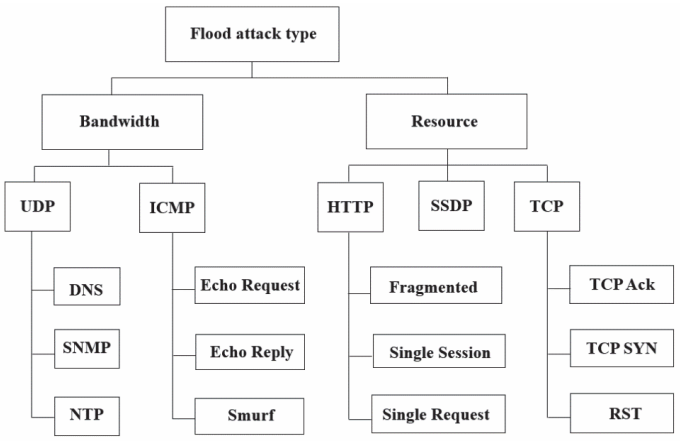

圖 3：泛濫攻擊類型

（一）資源消耗型

客戶端與伺服器需完成「三向式交握（three-way handshake）」，TCP 連線

才可以建立，此為連接導向的特性，而 UDP 為非連接導向。當客戶端向伺服器傳送 TCP 封包，即表示有 TCP 連線的要求，當伺服器返回 SYN-ACK，證明伺服器有收到此要求而回覆給客戶端，最後，客戶端再次傳送 ACK 給伺服器，這個連線要求才算是完成，如圖 4 所示。由於這種協定方式被攻擊者找出其弱點，即對該特定伺服器不斷的送出連線請求，但這些請求的 IP 卻是偽造的，導致該伺服器回覆連線無法獲得確認，而讓這些半連接（half-open）狀態的連線佔滿了有限的佇列空間，無法完成「三向式交握」的程序，伺服器資源耗盡的結果，除了無法即時回應及處理正常用戶的請求，嚴重者更會導致雲端伺服器停止而無法運作。

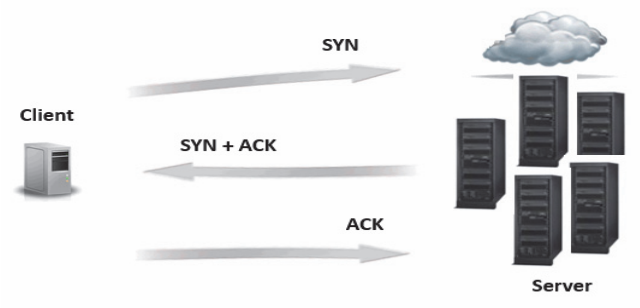

圖 4：TCP 三向式交握示意圖

HTTP 泛濫攻擊又稱 CC 攻擊，使用者在應用層與伺服器建立連線時，會發出 request 封包，當伺服器收到此 request 封包時，即會回傳 response 封包給使用者。攻擊者在短時間內蓄意發出大量的 request 封包時，伺服器為了處理大量湧入的 request，在處理過程中即耗費相當大的系統資源，幾秒內就會把伺服器資源耗盡，如圖 5 所示。

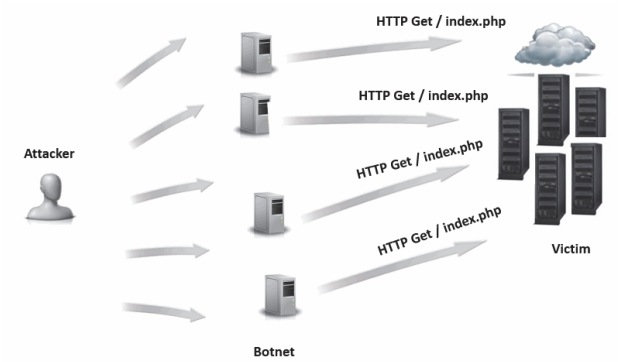

圖 5：HTTP 泛濫攻擊示意圖

（二）頻寬消耗型

ICMP 是一個錯誤處理與回報的機制，它的用途為傳遞關於網絡是否暢通、主機是否可達、路由是否可用等網絡本身的控制消息。攻擊者便利用此非連接導向的特性利用 ping 工具不斷向目標系統發送大量的 ICMP port、host、network、unreachable 的回應訊息，導致系統資源負荷過重而當機，如圖 6 所示。

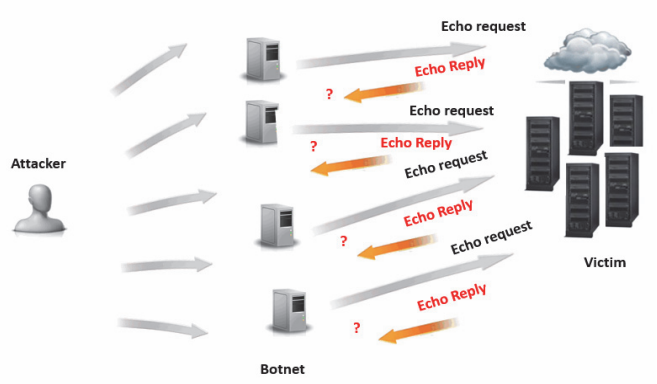

圖 6：ICMP 泛濫攻擊示意圖

由於 UDP 為非連接導向，因此它不需經過三向交握的程序就可直接向伺服器請求服務。伺服器一旦接收到 UDP 封包後則會指派應用程式來處理，假設沒有應用程式可處理 UDP 封包，伺服器即會回傳「無法到達」的 ICMP 封包訊息給來源 IP。攻擊者以偽造的來源 IP 大量且密集的發送 UDP 封包到目標伺服器，導致目標伺服器的頻寬飽和與癱瘓網段，而造成無法正常存取之服務如圖 7 所示。

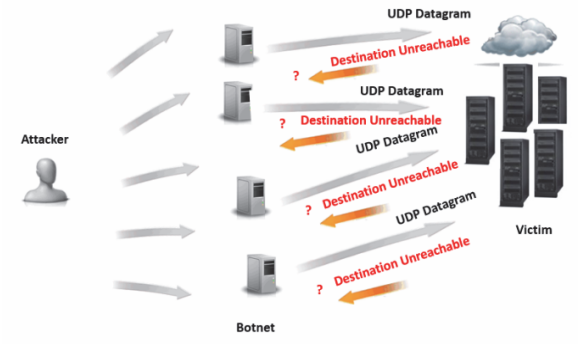

圖 7：UDP 泛濫攻擊示意圖

三、泛濫攻擊偵測與對策

一般偵測到有攻擊的跡象時，往往是將受害的雲端伺服器與網路斷開連接後再修復其衍生的問題。由於遭受泛濫攻擊而浪費了伺服器大量的計算資源，任何泛濫攻擊偵測機制的最終目標都是儘快發現攻擊並阻止它們。目前泛濫攻擊偵測機制，大多結合特徵選擇和機器學習法來偵測 SYN 的泛濫攻擊（Al-Hawawreh et al. 2017）。首先，資料集是擷取 TCP/IP 表頭屬性再分別利用三種特徵選擇方式去除不重要的屬性，並交叉比對執行特徵選擇後的結果，選取共有的特徵再使用四種機器學習法評估，結果偵測 SYN 泛濫攻擊的正確率有 99.33%。只利用經過特徵篩選後的資料集並設定流量閥值及建立異常偵測機制來判斷是否有 SYN 泛濫攻擊發生，結果有效的偵測到攻擊跡象並降低 CPU 的負荷（Kshirsagar et al. 2016）。採用防火牆規則來識別 SYN 泛濫攻擊，結果有 97.5%的正確率（Hussain et al. 2016）。結合多種的機器學習法分類模型來識別 TCP 泛濫攻擊，偵測後的結果有 97%的正確率（Sahi et al. 2017）。使用 SNMP MIB 的資料運用不同的機器學習方法經過學習與測試後的結果在 UDP 泛濫攻擊偵測的正確率可達 99.03%（Namvarasl & Ahmadzadeh 2014），雖然沒有使用特徵篩選，其偵測後的結果仍有如此高的正確率，判斷其原因是採用 SNMP MIB 的資料集，由此可見，過濾及擷取後的特徵仍是偵測泛濫攻擊主要的關鍵因素之一。

為了偵測網路層和應用層的攻擊，採用了三種機器學習的分類模型，針對由網路流量收集而來的封包資料集進行學習和訓練，並用來偵測網路層和應用層的攻擊（Alkasassbeh et al. 2016）。此種偵測方法是利用機器學習之隨機森林的分類模型針對由網路流量收集而來的封包資料集，經過機器學習後的結果在偵測網路層和應用層的泛濫攻擊有 98.02 %的正確率。本偵測方法雖為目前較新且常用的氾濫式攻擊偵測方法之一，但這種偵測方法缺少徵篩選機制做為機器學習法的前置處理，因此，本論文以此偵測方法為基礎並加以改善。

四、特徵選擇與隨機森林機器學習演算法

網路報告中（TechRepublic 2018），說明特徵選取也可稱為變量選擇、屬性選擇或變量子集選擇，意思是為了構建機器學習模型而選擇相關特徵的過程。該過程自動搜尋資料集中的最佳屬性子集，而最佳的概念則與嘗試解決的問題有關。良好的特徵選取方法可由大量特徵中，挑選出有助於分類之特徵，特徵選取為偵測機制在機器學習評估前的處理方法，每個特徵是一個或一個以上封包欄位的組合，其重點在於如何利用特徵之間的關連性，選取後的特徵可藉由機器學習演算法來評估泛濫攻擊的準確性，由於資料中不相關、多餘及不恰當的特徵會降低偵測的準確度，所以特徵選取的好壞對偵測系統的判斷和效能有很大的影響。而特

徵選擇的方法更有助於機器學習建構準確的偵測模型，特徵選擇有三個主要好處：減少過度擬合、提高準確性、減少訓練時間（Brownlee 2014）。關於特徵篩選機制的算法在資料探勘軟體工具 Weka 中，有三種設定：Filter、Wrapper 和 Embedded，分類如圖 8 所示，其說明如下：

1. Filter：即應用統計學對每一個特徵進行評等，並按分數排序。屬於單變量，可獨立考慮特徵。
2. Wrapper：選擇一組特徵作為搜索問題，利用不同組的合併與其他組合進行比較。它可以是隨機，也可以使用啟發法。
3. Embedded：此方法在於找出創建模型時哪些功能是較有助於模型的準確性。

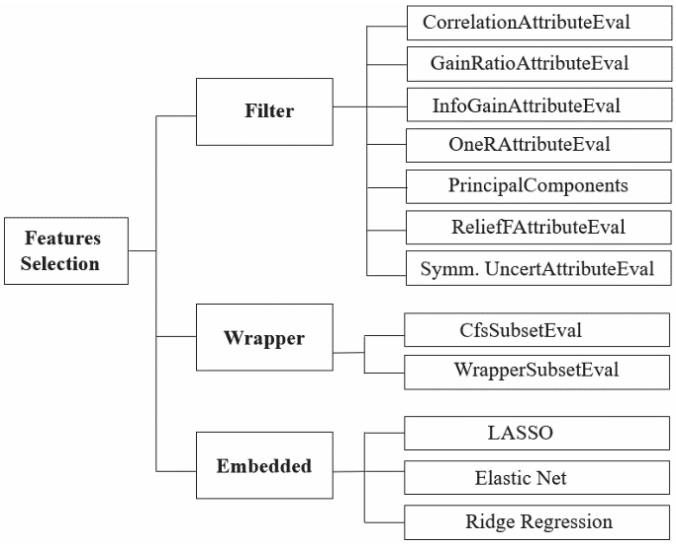

圖 8：在 Weka 中的特徵類型

網路報告中（TechRepublic 2018），說明在機器學習中，隨機森林（random forest）是一個包含多個決策樹的分類器，輸出的類別是由個別樹輸出的類別的眾數而定。近年來，在分類方法中，隨機森林演算法受到許多專家及學者的青睞，它所執行的效能及偵測能力不但很高，而且容易理解和使用。隨機森林中的每一棵樹都會自行分類，並進行投票，最後票數最多的則為分類結果。使用隨機森林分類模型對 DDoS 攻擊進行分類偵測，實驗結果證明該分類模型可以準確的區分正常流量和攻擊流量（于鵬程等 2017）。隨機森林建構的過程如圖 9 所示，其演算法步驟為樣本抽樣→母體抽樣→決策樹→結合，說明如下：

1. 對於每棵樹而言，隨機地從資料集中抽取 N 個樣本作為訓練。

2. 每個樣本的特徵維度為 M，隨機地從 M 個特徵中選取 D 個特徵子集，當該樹進行分裂時，則從特徵子集 D_1 至 D_M 中選擇最優。
3. 沒有針對分類樹做剪枝，讓它盡其可能的生長。
4. 最後生長出最多分類樹則構成了森林，而每棵樹會做分類選擇，選擇結果為票數最多的做為分類的選項。

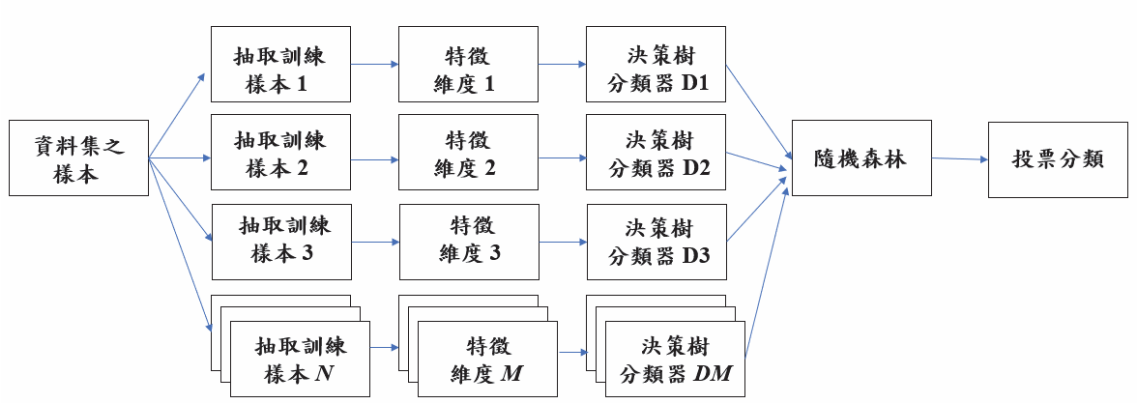

圖 9：隨機森林建構過程

參、HFADS 之模組運作與演算法設計

本節說明本文所提出的 HFADS 方法之模組運作原理與其功能，並設計其演算法。所設計的 HFADS 防禦機制可根據篩選後封包的特徵再透過機器學習的分類模型而偵測是否具有泛濫攻擊特性的跡象。

一、HFADS 模組運作與功能說明

根據文獻探討對於泛濫攻擊的了解，若伺服器上發現在短時間內有疑似大量連線的要求，勢必會有巨量的網路流量產生而佔據網路頻寬，導致網路運作異常緩慢。本論文為了能夠在最短的時間內發現攻擊跡象，必須先過濾、擷取和篩選封包的特徵後再藉由機器學習演算法建立分類模型，透過訓練與學習，最終偵測是否有攻擊行為發生。若要能夠監控伺服器虛擬化的網路效能，必需安裝虛擬機器監控程式，即 Hypervisor。目前市面上常見的監控軟體（hypervisor）產品有 VMware ESX Server、Microsoft Hyper-V、Linux KVM（kernel-based virtual machine）等。本文所提出的 HFADS 防禦機制可以執行在虛擬化雲端平台上的虛擬機器監控軟體（hypervisor）所在的虛擬機器上，如圖 10 所示。

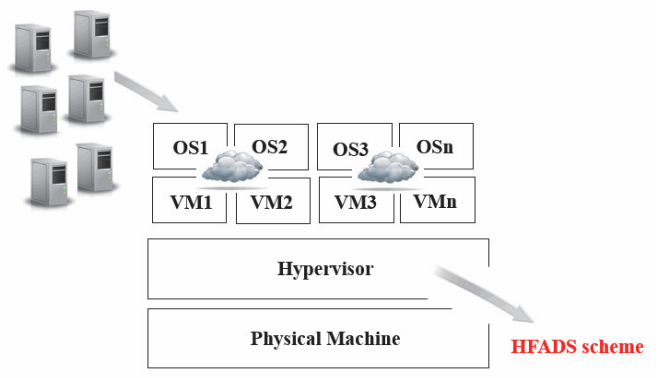

圖 10：HFADS 執行位置示意圖

本文所提出的 HFADS 防禦機制主要分為三個模組，分別為：資源監控、資料特徵篩選和機器學習評估，其各模組功能說明如下：

（一）資源監控模組（Resource Monitor Module）

執行 Keep_monitor_CPU 以監控 CPU 的使用率，以判定 CPU 使用率是否在正常範圍內，若有持續偏高的 CPU 使用量比率或者使用率有短暫突然增加的跡象，即表示有異常狀況發生，如果使用率低於門檻值時，可能代表佇列中已有處理序在等候處理器時間，此時即會觸動命令行工具 TShark 以擷取網路封包並過濾封包協定為 UDP、ICMP、HTTP 的資料並存成一個新的封包檔。

（二）資料特篩選模組（Data Features Selection Module）

根據第一個模組所產生的封包檔，以統計的方法進行特徵的前置處理，整理成 Excel 表格，並存成 csv 檔，再把該 csv 檔交給資料探勘軟體工具 Weka 做資料的特徵篩選。此工具可提供資料處理、特徵選擇、分類、分群、迴歸、關聯規則、視覺化等功能。為了協助隨機森林機器學習分類法建立準確的偵測模型，使用三種不同且是目前流行的特徵選擇方法之評等後的特徵執行。首先，啟動 Weka 後選擇 Explorer 在 Preprocess 載入資料集後再切換到 Select attributes 標籤頁分別選擇三種特徵方法進行特徵的篩選，記錄被三種特徵方法挑選之特徵子集合，找出共有的特徵（Combine Multiple Features Selection; CMFS）。三種特徵篩選方法說明如下：

1. 資訊獲利（Information Gain）：根據與分類有關的每個屬性之資訊增益進行評估。資訊獲利即「測試前的資訊量」減去「測試後的資訊量」。資訊獲利可計算每個屬性的輸出變量（又稱為熵），資訊提供越多，其屬性的獲利將會越高。換句話說，資訊獲利越大，代表這個特徵的選擇性越好。此種算法依賴「Entropy（熵）」如公式(1)所示（Al-Hawawreh 2017）。

$$IG\ (B|A) = E(B) - E(B|A) \quad (1)$$

2. 獲利比率（Gain ratio）：根據與分類有關的每個屬性的獲利比率進行評估。獲利率的算法則是考慮每個特徵值選擇特徵時的概率。它利用熵對資訊獲利進行一致化，選擇過程偏向多值特徵來處理資訊獲利的缺陷。屬性值 A 和屬性值 B 的獲利比率如公式（2）所示（Al-Hawawreh 2017）。

$$\text{Gain Ratio}=\frac{\text{Information Gain (B,A)}}{\text{Intrinsic Value (A)}} \quad (2)$$

3. 基於相關性的方法（Correlation-based Feature Selection; CFS）：一種簡單的過濾算法，基於相關性的啟發式評估函數，對屬性子集進行排名，根據屬性子集中每一個特徵的預測能力以及關聯性進行評估（Namvarasl & Ahmadzadeh 2014）。

（三）機器學習評估模組（Machine Learning Evaluation Module）

重新在 Preprocess 裡載入資料集，並挑選主要的二種特徵方法所產生的共有特徵屬性後，把多餘且無相關的特徵刪除再切換到 classify 標籤頁，選擇隨機森林分類模型，分別執行以下二種不同的測試選項，測試選項說明如下：

1. 使用訓練集（Use training set）：訓練和測試使用同一份資料集，對文件中的所有資料進行測試和訓練。
2. 10 次交叉驗證方式（Cross-validation with 10 folds）：使用交叉驗證的方式作為評估，所用的數值則填在 Folds 欄位中，預設值為 10。意思是將資料集分成十份，輪流將其中的 9 份做訓練、1 份做測試，如此循環 10 次，將其平均後即為最後的結果。

最後根據第二種機器學習評估模組所產生的結果，若偵測到有封包協定類型為 UDP、ICMP、HTTP 等泛濫攻擊跡象，即立即產生警告訊息並發送給網路管理員做適當的處理。HFADS 防禦機制的運作流程圖，如圖 11 所示。

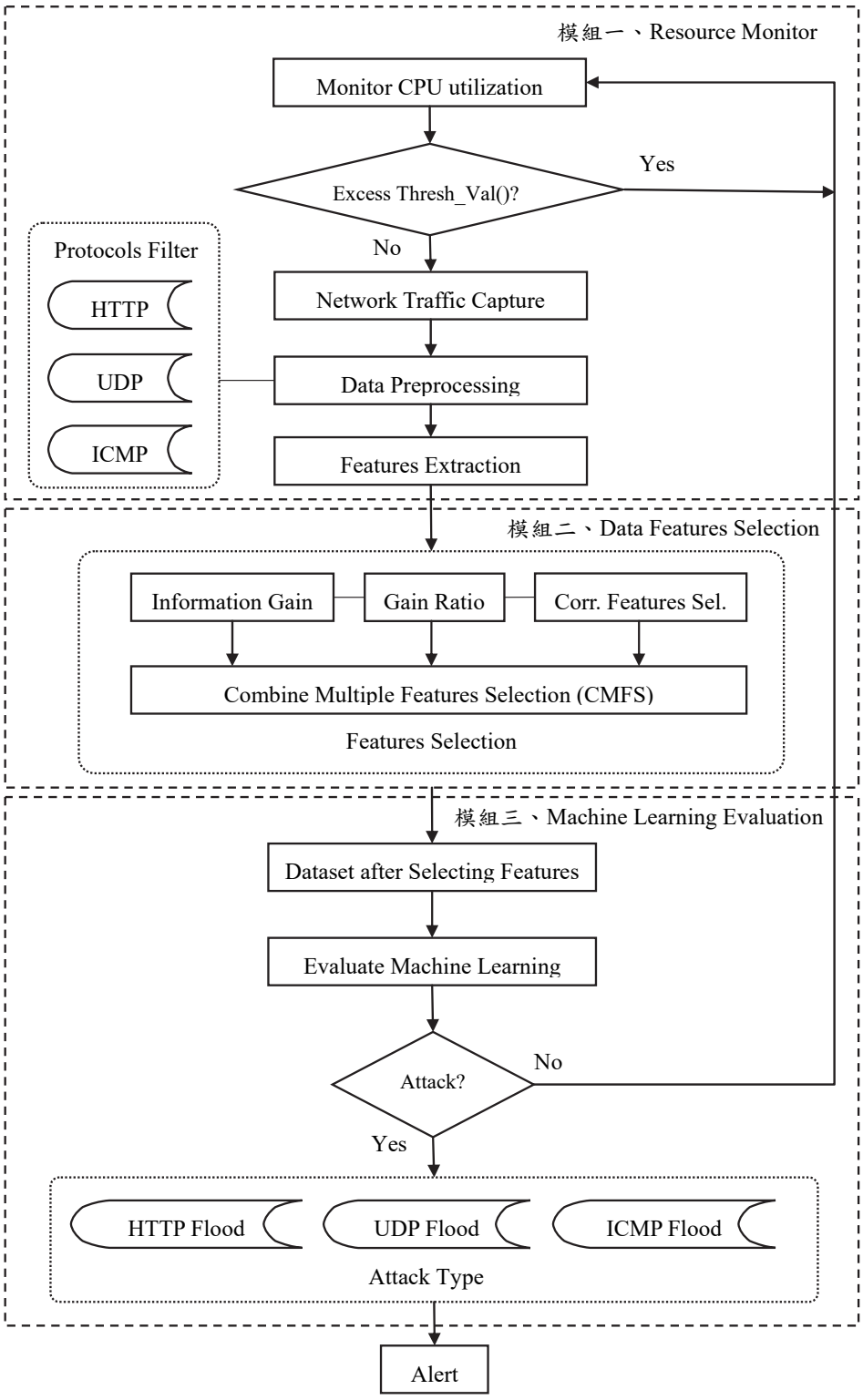

圖 11：HFADS 運作流程圖

二、HFADS 演算法設計

根據圖 11 之 HFADS 運作流程圖以設計 HFADS 方法，其演算法之虛擬碼及說明如下：

Algorithm HFADS

```
Algorithm HFADS {
    Input:
        String[] NP                          //封包集合
        Load[] NP                            //匯入封包
        Filter[] NP                          //過濾封包
        FeatExtr[] NP                        //特徵擷取
        SelectFeature[] NP                   //特徵篩選
        FeatComb[] NP                        //綜合三種篩選方法特徵
        ML_RF_Eval[] NP                      //隨機森林偵測
        bool ifAttacker
        String[] Alert
        Int Thresh_Val
    Output:
        To complete HFADS Method.
    Method:
        Input Thresh_Val;
  BEGIN {
      Module 1. Resource Monitor :           //模組 1：資源監控
          if CPU <= Thresh_Val():            // CPU 使用率是否低於門檻值
              NP_Load();                     //匯入封包
              NP_filter ();                  //過濾封包
              NP_FeatExtr();                 //截取特徵
       else
              Keep_monitor_CPU;              //監控 CPU 使用率
       endif
      Module 2. Data Features Selection :    //模組 2：資料特徵篩選
            NP_SelectFeature ();             //執行三種特徵篩選方法
            NP_FeatComb();                   //綜合三種篩選方法特徵
      Module 3. Machine Learning Evaluation: //模組 3：機器學習評估
```

```
            NP_ML_RF_Eval ();                  //使用隨機森林
            If (ifAttacker = true)             //有攻擊跡象則發送警告
            alertUserToAdmin                    訊息給管理員
    } END
    Procedure NP_Load () {                     //匯入封包
        String np = getNP()
        NP = np.Split(‘ ’)
    } END NP_Load
    Procedure NP_Filter() {                    //過濾封包
        Suspicious = ifUDP_ICMP_HTTP()
    } END NP_Filter
    Procedure NP_FeatExtr() {                  //特徵擷取
        HFADS_FeatExtr= NP.getAverage()
    } END NP_FeatExtr
    Procedure NP_SelectFeature() {             //特徵篩選方法
        HFADS_FeatSel=IF()                     //執行 Information Gain 方法
        HFADS_FeatSel=GR()                     //執行 Gain ratio 方法
        HFADS_FeatSel=CA()                     //執行 Correlation-based Feature
                                                Selection 方法
    } END NP_ SelectFeature
    Procedure NP_FeatComb() {                  //綜合三種篩選方法特徵
HFADS_FeatComb=CMFS()                          //執行 CMFS（Combine Multiple
                                                Features Selection）方法
    } END NP_FeatComb
    Procedure NP_ML_RF_Eval() {                //隨機森林偵測
        if HFADS_ML_RF_Eval = Attack.feat :
          return true
        else
          return false
    } END NP_ML_RF_Eval
  Procedure alertUserToAdmin(){                    //發送警告訊息給管理員
        Display Alert
    } END alertUserToAdmin
    }
    END HFADS.
```

肆、實驗環境建置與結果分析

本論文利用 M.A.所建立的資料集來源進行模擬實驗，透過資料探勘軟體工具 Weka 的特徵篩選功能所篩選後的特徵值並配合隨機森林機器模型進行分類。最後，再根據實驗後的結果進行分析，以驗證所提出的 HFADS 機制之雲端泛濫攻擊之防禦績效。

一、實驗環境建置

實驗環境的建置採用虛擬機器與實體機器模擬雲端的環境。首先，在一台 Windows 10 的實體機器上安裝 VMware Workstation 12 PRO 虛擬軟體，並建立二台虛擬機。一台安裝 Windows 2008 並執行資源監控模組以監控 CPU 的使用率，同時啟動 WireShark 進行網路流量的監控，一旦 CPU 的使用率有突然衝高但低於門檻值現象時，即觸動命令行工具 TShark 中執行程序以擷取封包資料，並運用資料探勘軟體工具 Weka 的特徵選取裡的屬性模組針對擷取的資料集來選取適合的特徵。最後，再以 Weka 的機器學習模組來評估結果及偵測攻擊行為。另一台則安裝 Kali Linux 當作是攻擊者，直接對 Windows 2008 發動攻擊。模擬實驗之電腦軟、硬體規格說明如表 1 所示。資料集來源為 M.A.從網路入侵偵測系統（network intrusion detection system; NIDS）收集並統計後的網路封包（資料集來源：Alkasassbeh et al. 2016）。資料集的特徵屬性說明，如表 2 所示。模擬實驗工具說明，如表 3 所示。

表 1：實體機和虛擬機配置規格

軟、硬體配置	實體機	VM 1 （victim）	VM 2 （attacker）
OS	Windows 10	Windows 2008	Kali Linux
CPU	Intel® Core™ i5-7300U CPU @2.60GHz2.71 GHz	1 Core 2.6 GHz	1 Core 2.6 GHz
Memory	8 GB or above	2 GB	2 GB
Disk	235 GB or above	30 GB	80 GB

表 2：資料集特徵屬性說明

特徵編號	特徵名稱	特徵說明
1.	SRC ADDress	封包來源 IP
2.	DES ADDress	目的地 IP
3.	PKT ID	封包 ID
4.	FROM NODE	事件發生的地點（開始端）
5.	TO NODE	事件發生的地點（目的端）
6.	PKT TYPE	封包類型
7.	PKT SIZE	封包大小
8.	FLAGS	旗標
9.	FID	封包所屬資料流
10.	SEQ NUMBER	序號
11.	NUMBER OF PKT	封包數量
12.	NUMBER OF BYTE	位元數
13.	NODE NAME FROM	網路節點名稱（開始端）
14.	NODE NAME TO	網路節點名稱（目的端）
15.	PKT IN	接收到的封包總數
16.	PKT OUT	傳送封包總數
17.	PKTR	PKT IN/PKT OUT
18.	PKT DELAY NODE	封包延遲節點
19.	PKT RATE	封包速率
20.	BYTE RATE	位元速率
21.	PKT AVG SIZE	封包平均大小
22.	UTILIZATION	利用率
23.	PKT DELAY	封包延遲
24.	PKT SEND TIME	封包傳送時間
25.	PKT RESEVED TIME	封包接收時間
26.	FIRST PKT SENT	傳送第一個封包
27.	LAST PKT RESEVED	接收最後一個封包

表 3：模擬實驗工具說明

實驗工具	實驗工具說明
Wireshark V2.6.0	Wireshark 是一個免費開源的網路封包分析軟體，可以擷取網路封包並詳細的顯示網路封包資料
TShark	TShark 是命令行工具，通過指令擷取網路封包並導出到文件
Python V3.6.5	Python，屬於通用型的高階程式語言並廣泛使用在不同領域
Weka V3.8.3	基於 JAVA 環境下所開發的免費資料探勘之機器學習軟體工具，可提供資料處理、特徵選擇、分類、分群、迴歸、關聯規則、視覺化等功能

二、關鍵績效指標定義與說明

透過模擬實驗將 HFADS 與 M.A.提出的方法進行比較與分析，提出精確率、召回率、總正確率、平均處理時間等 4 項關鍵績效指標（KPIs），其指標定義與說明如表 4 所示。此外，為表示分類精確度，運用機器學習之監督學習中常用的衡量模型指標之誤差矩陣表，其表達方法如表 5 所示。誤差矩陣表格中的 Positive 在此表示為「攻擊封包」，Negative 在此表示為「正常封包」。TP（true positive）表示在真實情況為攻擊封包且經隨機森林所做的判斷後，被正確分類為攻擊封包；TN（true negative）表示在真實情況為正常封包且經隨機森林所做的判斷後，被正確分類為正常封包；FP（false positive）表示在真實情況為正常封包但經隨機森林所做的判斷後，被錯誤分類為攻擊封包；FN（false negative）表示在真實情況是攻擊封包但經隨機森林所做的判斷後，被錯誤分類為正常封包。

表 4：關鍵績效指標說明

關鍵績效指標（KPIs）	關鍵績效指標說明
精確率（PR） （precision rate）	表示被分類為攻擊的封包中實際為攻擊的比率，此值越高越好，其計算如公式(3)所示。
召回率（RR） （recall rate）	在所有攻擊封包中，被正確識別為攻擊的比率，此值越高越好，其計算如公式(4)所示。
總正確率（TAR） （total accuracy rate）	機器學習模型的整體判斷正確率，此值越高越好，其計算如公式(5)所示。
平均處理時間（APT） （average processing time）	基於選擇後的特徵所採用的機器學習法之建立模型的時間，再加上運用此模型做測試時所花費的時間，此值越低越好，其計算如公式(6)所示。

表 5：衡量模型指標之誤差矩陣

真實狀況	經由隨機森林所做的判斷	
	Positive（攻擊封包）	Negative（正常封包）
True（攻擊封包）	TP	TN
False（正常封包）	FP	FN

本文所提出的精確率、召回率、總正確率與平均處理時間等 4 項關鍵績效指標（KPIs） 之計算方式如公式(3)至公式(6)所示。

$$PR=\frac{TP}{TP+FP} \tag{3}$$

$$RR=\frac{TR}{TP+TN} \tag{4}$$

$$TAR=\frac{TP+TN}{TP+FN+FP+TN} \tag{5}$$

$$APT=\text{time taken to build model + time taken to test model on training data} \tag{6}$$

三、模擬實驗與結果分析

本文使用 M.A.從 NIDS（network intrusion detection system）收集網路層及應用層的封包所產生的資料集，匯入資料探勘軟體工具 Weka 後分別執行三種不同的篩選方法。首先，先執行 Information Gain，執行後的結果挑選權重（Ranked attributes）為 0.5（含）以上的特徵。接著再執行 Gain ratio，執行後的結果也是挑選權重為 0.5（含）以上的特徵。然後再執行 Correlation-based Feature Selection，執行後的結果仍是挑選權重為 0.5（含）以上的特徵。最後，根據以上三種特徵方法選取後的編號選出所有可能呈現的特徵，並稱此特徵選擇方法為 CMFS （Combine Multiple Features Selection），其資料集的特徵編號彙整後，如表 6 所示。

表 6：特徵選擇方法及篩選後的特徵編號

特徵選擇方法	資料集篩選特徵編號
Information Gain	1, 2, 3, 7, 9, 10, 12, 13, 19, 20, 21, 22, 23, 24, 26, 27
Gain ratio	1, 2, 3, 6, 7, 9, 11, 19, 20, 21, 23, 26, 27
Corr. features sel.	2, 6, 7, 9, 10, 11, 12, 19, 21, 22, 23, 26, 27
CMFS	1, 2, 3, 6, 7, 9, 10, 11, 12, 13, 19, 20, 21, 22, 23, 24, 26, 27

根據表 6 所做的篩選後的資料集特徵，在資料探勘軟體工具 Weka 的分類頁面，選擇隨機森林演算法進行分類，再分別執行二種不同的測試選項：(1)Use training set、(2)Cross-validation with 10 folds，此模擬實驗在 use training set 的測試選項中，執行的結果以誤差矩陣表格，如表 7 所示。在 Cross-validation with 10 folds 的測試選項中，執行的結果以誤差矩陣表格，如表 8 所示。最後的結果，在此二種測試選項中，以 Use training set 的測試選項在偵測網路層通訊協定 UDP、ICMP 和應用層通訊協定 HTTP（使用 TCP）的混合式泛濫攻擊所獲得的數據較為精確。

表 7：Use training set 之誤差矩陣表

真實狀況	經由隨機森林所做的判斷				
	UDP-Flooding	ICMP-Flooding	HTTP-Flooding	Normal	無法判斷
UDP-Flooding	97519	0	0	2	0
ICMP-Flooding	0	6208	0	3	0
HTTP-Flooding	0	0	1997	0	0
Normal	0	0	0	25109	0

表 8：Cross-validation with 10 folds 之誤差矩陣表

真實狀況	經由隨機森林所做的判斷				
	UDP-Flooding	ICMP-Flooding	HTTP-Flooding	Normal	無法判斷
UDP-Flooding	90055	822	0	6606	38
ICMP-Flooding	937	2376	27	2693	178
HTTP-Flooding	0	3	1888	0	106
Normal	5745	2162	1	17111	90

根據表 7 執行 Use training set 所得到的數據，針對本文所提出的精確率、召回率、總正確率以及平均處理時間等關鍵指標進行詳細說明與實驗結果分析。

（一）精確率（precision rate）

在資料集裡被分類為攻擊的封包，其實際為攻擊封包的比率，此精確率值越高越好。被除數為在真實情況為攻擊封包且經由隨機森林所做的判斷後，被正確分類為攻擊封包所得到的值，除數則是上述情況再加上在真實情況為正常封包但經由隨機森林所做的判斷後，被錯誤分類為攻擊封包，最後所得到的百分比。UDP 的精確率為 100%，比 M.A.的 95.19%提升了 4.8%；ICMP 的精確率為

100%，比 M.A.的 52.92%提升了 47.1%；HTTP 的精確率為 100%，比 M.A.的 98.60%提升了 1.4%，如圖 12 所示。

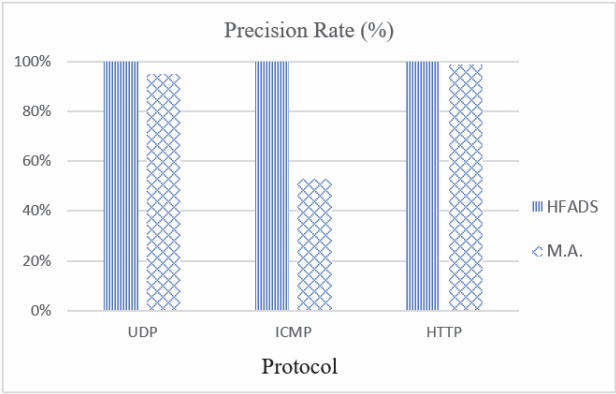

圖 12：精確率之模擬結果

（二）召回率（recall rate）

在資料集裡，所有的攻擊封包中，被正確識別為攻擊的比率，此值越高越好。被除數為在真實情況為攻擊封包且經由隨機森林所做的判斷後，被正確分類為攻擊封包所得到的值，除數則是上述情況再加上在真實情況是攻擊封包但經由隨機森林所做的判斷後，被錯誤分類為正常封包，最後所得到的百分比。UDP 的召回率為 99.99%，比 M.A.的 90.18%提升了 9.8%；ICMP 的召回率為 99.95%，比 M.A.的 33.57%提升了 66.4% ；HTTP 的召回率為 100%，比 M.A.的 99.41%提升了 0.6%，如圖 13 所示。

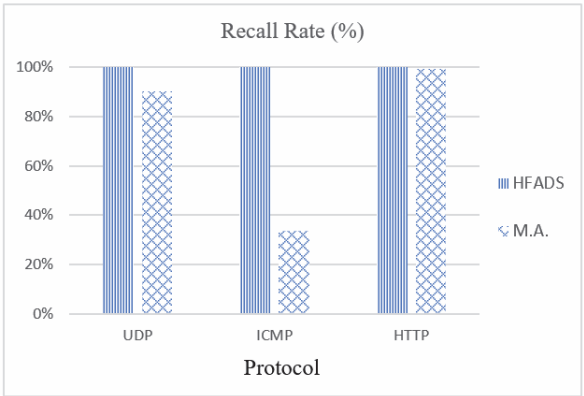

圖 13：召回率之模擬結果

（三）總正確率（total accuracy rate）

被除數為在真實情況為攻擊封包且經由隨機森林所做的判斷後，被正確分類為攻擊封包所得到的值，再加上在真實情況為正常封包且經由隨機森林所做的判斷後，被正確分類為正常封包所得到的值，除數則是真實情況所有的封包，在資料探勘軟體工具 Weka 執行三種不同的測試選項後，整體的偵測總正確率為 99.98%，比 M.A.的 98.02%提升了 2%，如圖 14 所示。

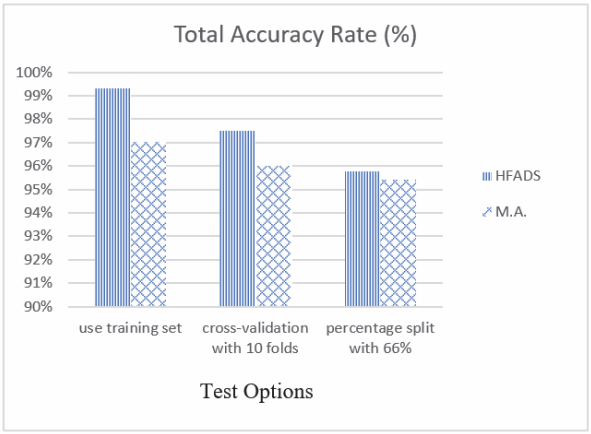

圖 14：總正確率之模擬結果

（四）平均處理時間（average processing time）

欲縮短*處理時間*，關鍵即在於掌握多少攻擊特徵，此模擬實驗花費在建立模型與測試的時間，平均是 65.34 秒比 M.A.的 68.88 秒改善了 5.14%，如圖 15 所示。

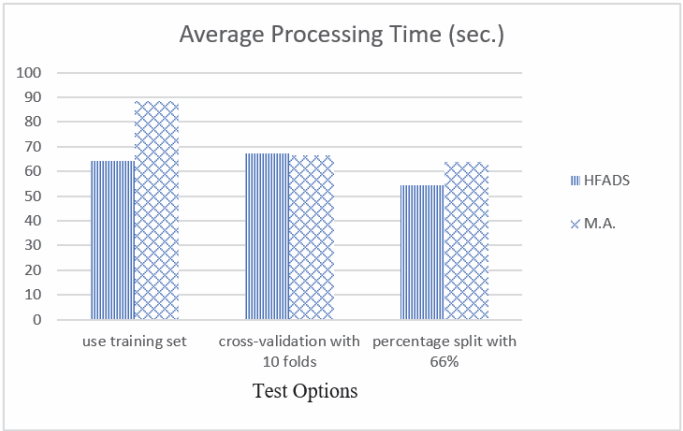

圖 15：平均處理時間之模擬結果

本論文不但以 M.A.所使用的方法為基礎，同時運用該作者在 NIDS 產生的資料集做為模擬實驗的依據而提出 HFADS 偵測混合式泛濫攻擊的機制。此機制在執行隨機森林分類前，以特徵篩選的方法過濾了多餘且不必要的特徵後再執行模擬實驗，所獲得的結果，可看出由於 M.A.只有執行隨機森林機器學習，以致於在精確率和召回率的關鍵指標以 ICMP 的數據較不理想，然而受限於資料集的樣本數及特徵項目的擷取，以致於 HFADS 都可以維持在 99%以上，但在真實環境下有太多的變數及特徵難以預料及掌控，運用本機制以達到較佳的偵測結果。以上模擬實驗結果之數據彙總，如表 9 所示。

表 9：模擬結果之數據彙總表

關鍵績效指標 / 方法		HFADS	Alkasassbeh et al.（M.A.）	改善比率
精確率（單位：%）	UDP	100%	95.19%	4.8%
	ICMP	100%	52.92%	47.1%
	HTTP	100%	98.60%	1.4%
召回率（單位：%）	UDP	99.99%	90.18%	9.8%
	ICMP	99.95%	33.57%	66.4%
	HTTP	100%	99.41%	0.6%
總正確率（單位：%）		99.98%	98.02%	2%
平均處理時間（單位：Sec.）		65.34	68.88	5.14%

伍、結論

雲端虛擬化技術的運用不但讓政府或企業的營運成本大幅減少，更讓決策執行效率更有競爭力，但安全防護的挑戰也隨之而來。雲端虛擬化之共享計算資源的特性，讓惡意的使用者有新的攻擊管道，多租戶的架構，辨識攻擊者的途徑更加困難。因此，本文基於特徵篩選並結合隨機森林（random forest）分類模型偵測 UDP、ICMP 和 HTTP（使用 TCP）的混合式泛濫攻擊而提出之防禦機制，稱為 HFADS，此防禦機制有資源監控、資料特徵篩選和機器學習評估等三個功能模組，並且使用資料探勘軟體工具 Weka 免費開放軟體的三個特徵篩選工具：Information Gain、Gain ratio、Corr. features sel.進行特徵屬性的篩選，運用隨機森林機器學習分類模型對篩選後的特徵進行偵測，實驗結果顯示隨機森林所建構的模型其整體偵測的總正確率為 99.98%、平均處理時間為 65.34 秒，並與 M.A.所提出的隨機森林演算法以偵測混合式泛濫攻擊進行模擬比較，除有較佳的精確率與召回率外，在總正確率指標提升了 2%，平均處理時間改善約 5.14%。由此可

見，本文所提出的 HFADS 防禦機制在混合式泛濫攻擊的辨識上有較佳的精確率與召回率、較高的總正確率、以及較短的平均處理時間。整體而言，本文之主要研究成果說明如下：

1. 泛濫攻擊是 DDoS 演進的攻擊方式之一，針對此一新型式的攻擊，本文提出結合特徵的篩選機制和隨機森林分類模型的混合式泛濫攻擊防禦機制 HFADS，以偵測不同的通訊協定為 UDP、ICMP、HTTP 等。
2. 透過監測 CPU 的使用率是否低於門檻值現象，以偵測是否有泛濫攻擊跡象進而觸發命令行工具 TShark 中執行程序，以擷取並過濾封包後而產生資料集，以做為防禦攻擊的基礎。
3. 本論文所提出的 HFADS 防禦機制，由於在執行隨機森林分類前，資料集已透過特徵篩選機制，過濾了不必要和不相關的特徵，所以在運作成本上不需使用太多的計算資源及負荷（overhead），便可以提高偵測混合式泛濫攻擊的總正確率及縮短平均處理時間。
4. 將 HFADS 與使用隨機森林偵測之 M.A.等方法進行模擬實驗，並提出精確率、召回率、總正確率和平均處理時間四種關鍵績效指標，實驗結果顯示本文所提出的 HFADS 機制比 M.A.方法在偵測網路層 UDP、ICMP 和應用層 HTTP（使用 TCP）等通訊協定上，均有較佳的精確率與召回率外，以及其總正確率為 99.98%提升了 2%和平均處理時間亦改善了 5.14%。

藉由模擬實驗後的結果分析，證明了本論文所提出的 HFADS 機制與 M.A.比較後，因為比 M.A.在運用機器學習訓練之前先執行了特徵篩選，所以在防禦混合式泛濫攻擊的總正確率有顯著的提升。本論文的主要貢獻在於所提出的 HFADS 防禦機制在運作成本上不但不會消耗大量計算資源及負荷，而且在特徵篩選及執行隨機森林分類的過程中，總正確率及平均處理時間都能夠穩定的維持在一定的水準，同時也證明本文所提出 HFADS 的混合式泛濫攻擊防禦機制比只有執行機器學習的方法更加完整。然而，受限於模擬實驗所使用由 M.A.所提供的資料集的樣本數及特徵項目的擷取，以致於 HFADS 皆可維持在 99.9%以上，但在真實環境下有太多的變數及特徵難以預料及掌控，運用本機制也許沒有預期的結果。然而，未來我們不但可依此設計原理，強化特徵篩選功能與善加運用機器學習演算法不斷的訓練和測試及自我學習的機制來提升偵測攻擊的總正確率，以因應真實環境的變化。此外，更可依此機制的結構應用在其他雲端上常見的攻擊之防禦機制上，如：反射攻擊（reflection attack）、側通道攻擊（side channel attack）、殭屍網路攻擊（botnet attack）、惡意程式攻擊（malware-injection attack）等。整體而言，在任何攻擊行為產生之前，此機制可先行過濾及篩選，並且提供建議解決方案給雲端供應商，如此才能料敵機先，決戰於千里之外。

未來的研究將納入在偵測傳輸層透過 TCP 為主的通訊協定，如：FTP、

SMTP 及 Telnet 等的泛濫式攻擊，並且採用其他的機器學習偵測及使用更多的關鍵指標以獲取更為準確的數據。為因應未來任何的攻擊型態，更可依此架構設以計自動偵測與辨識封包的功能，快速的擷取、過濾及篩選再加上機器學習不斷訓練與測試的機制並整合網路防護設備，搭配完善的雲端服務管理政策，讓此防護系統能更加的完整。以本文所設計的 HFADS 防禦機制為基礎，整合主機網路防護設備或雲端清洗中心，進而提供更安全的雲端運算環境，讓使用者能有效率地與安心地使用雲端環境的計算資源以提高整體處理績效並提昇其雲端服務品質。

誌謝

本論文承蒙兩位匿名審查委員之寶貴意見與悉心指正，使得本論文得以修正並更加完善，謹此致謝。

參考文獻

于鵬程、戚湧、李千目（2017），『基於隨機森林分類模型的 DDoS 攻擊檢測方法』，*電腦應用研究*，第三十四卷，第十期，頁 3068-3072。

Al-Hawawreh, M.S. (2017), 'SYN flood attack detection in cloud environment based on TCP/IP header statistical features, *Proceedings of the 8th IEEE International Conference on Information Technology (ICIT 2017)*, Al-Zaytoonah University of Jordan, Jordan, May 17, pp. 236-243.

Alkasassbeh, M., Hassanat, A.B.A, A1-Naymat, G. and Almseidin, M. (2016), 'Detecting distributed denial of service attacks using data mining techniques', *International Journal of Advanced Computer Science and Application*, Vol. 7, No. 1, pp. 436-445.

Alkasassbeh, M., Hassanat, A.B.A, A1-Naymat, G. and Almseidin, M. (2016), 'Dataset-detecting distributed denial of service attacks using data mining techniques', https://www.researchgate.net/publication/292967044_Dataset-Detecting_Distributed_Denial_of_Service_Attacks_Using_Data_Mining_Techniques

Brownlee, J. (2014), 'Feature selection to improve accuracy and decrease training time', https://machinelearningmastery.com/feature-selection-to-improve-accuracy-and-decrease-training-time/

Hussain, K., Hussain, S.J., Dillshad V., Nafees, M. and Azeem, M.A. (2016), 'An adaptive SYN flooding attack mitigation in DDOS environment', *International Journal of Computer Science and Network Security*, Vol. 16, No. 7, pp. 27-33.

JavaPipe (2018), '35 types of DDoS attacks explained', https://javapipe. com

/ddos/blog/ddos-types/

Kshirsagar, D., Sawant, S., Rathod, A. and Wathore, S. (2016), 'CPU load analysis & minimization for TCP SYN flood detection, *Procedia Computer Science*, Vol. 85, pp. 626-633.

Mell, P. and Grance, T. (2011), 'The NIST definition of cloud computing', https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

Namvarasl, S. and Ahmadzadeh, M. (2014), 'A dynamic flooding attack detection system based on different classification techniques and using SNMP MIB data', *International Journal of Computer Networks and Communications Security*, Vol. 2, No. 9, pp. 279-284.

Sahi, A., Lai, D., Li, Y. and Diykh M. (2017), 'An efficient DDoS TCP flood attack detection and prevention system in a cloud environment', *IEEE Access*, Vol. 5, pp. 6036-6048.

Somani, G., Gaur, M.S., Sanghi, D., Conti, M. and Buyya, R. (2017), 'DDoS attacks in cloud computing: Issues, taxonomy, and future directions', *Computer Communications*, Vol. 107, pp. 30-48.

TechRepublic (2018), 'GitHub hit with massive 1.35 Tbps DDoS attack, could be world's largest', https://www.techrepublic.com/article/github-hit-with –massive-1-35-tbps-ddos-attack-could-be-worlds-largest/

Wikipedia contributors (2016), 'Random forest', https://en.wikipedia.org/wiki/ Random_forest

Wikipedia contributors (2018), 'Feature selection', https://en.wikipedia.org/wiki/ Feature_selection

Wikipedia contributors (2018), 'DDoS', https://zh.wikipedia.org/wiki/%E9%98%BB%E6%96%B7%E6%9C%8D%E5%8B%99%E6%94%BB%E6%93%8A

Zargar, S.T., Joshi, J. and Tipper D. (2013), 'A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks', *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 4, pp. 2046-2069.