Communications of the CCISA
Vol. 31 No. 2 May 2025

基於機器學習隨機森林演算法應用於 Apache Weblog 之入侵偵測系統

陳品蓉[1]、翁浩宇[2]、黃正達[3*]
[1, 3] 元智大學資訊學院英語學士班、[2] 中央大學資訊工程系、[3] 元智大學資訊管理系
[1]s1113505@mail.yzu.edu.tw、[2] pony71909@gmail.com、[3*]cthuang2020@saturn.yzu.edu.tw

摘要

隨著現今網路技術的快速發展，人們依賴網際網路的機率大幅增加，網站作為資訊傳遞與服務平台已成為人們日常生活中不可或缺的一環，然而其在給予人們便利的同時，也使其成為有心人士攻擊的主要標，導致資安事件的數量劇增。因此，為了有效辨識與偵測惡意的網路攻擊，本研究提出一套基於機器學習 (Machine Learning, ML) 的 Apache weblog 攻擊分類研究，建立四種類別均衡的訓練資料集。此外，本研究擷取八項潛在特徵，以 Gini 特徵重要性 (Gini Feature Importance) 作為指標，進行消融實驗 (Ablation Study) 探討各特徵組合不同的訓練結果。從實驗數據中，顯示了本研究的研究方法能夠達到 99.49%的預測準確率，且精確率 (Precision)、召回率 (Recall) 與 F1 分數 (F1-score) 均高於 99%，基於實驗結果得以說明，利用本論文方法能夠建立更加可靠的攻擊分類器，驗證了結合機器學習與日誌特徵分析於資安領域的應用潛力。

關鍵詞：入侵偵測系統、機器學習、Apache Web Log、隨機森林演算法、特徵重要性

* 通訊作者 (Corresponding author.)

Regular Paper

Communications of the CCISA
Vol. 31 No. 2 May 2025

A Machine Learning-Based Network Intrusion Detection System for Apache Weblogs Using the Random Forest Algorithm

PIN-RONG CHEN[1], HAO-YU WENG[2], CHENG-TA HUANG[3*]
[1, 3*]International Bachelor Program in Informatics, Yuan Ze University, Taiwan
[2]Department of Computer Science and Information Engineering, National Central University, Taiwan
[3*]Department of Information Management, Yuan Ze University, Taiwan
[1] s1113505@mail.yzu.edu.tw, [2] pony71909@gmail.com, [3*]cthuang2020@saturn.yzu.edu.tw

Abstract

With the rapid development of internet technologies these days, people start relying on the internet more and more. As a result, websites have become essential platforms for information dissemination and service delivery in daily life. However, while they bring convenience to people, they have also become primary targets for malicious actors which lead to a sharp increase in cybersecurity incidents. Therefore, this study proposes a machine learning-based approach to classify attacks in Apache weblogs in order to effectively identify and detect malicious network attacks. A balanced training dataset consisting of four attack categories was also constructed. Additionally, eight potential features were extracted, and Gini Feature Importance was used as an evaluation metric to conduct ablation studies on different feature combinations.

With regard to the experimental results, they demonstrate that the proposed method can reach a prediction accuracy up to 99.49% and with precision, recall, and F1-score all exceeding 99%. Based on the experiment results, the proposed approach can be used to build a more reliable attack classifier and verify the potential of combining machine learning with log feature analysis in the field of cybersecurity.

Keywords: Intrusion Detection System (IDS), Machine Learning (ML), Apache Web Log, Random Forest Algorithm, Feature Importance

壹、前言

在現今網路環境日益成熟的背景下，網站不僅是資訊展示的平台，更承載著許多機密資料，其中可能含有員工訊息、顧客資料、商業交易紀錄及其他內容。但這也使得網站成為駭客攻擊的主要目標，而現今的網路攻擊的型態日趨多樣化與複雜化，這些攻擊不僅可能導致資料外洩，更可能影響系統可用性與企業聲譽，對資訊安全造成嚴重威脅。

在資訊安全領域中，日誌資料完整記錄了用戶請求與伺服器回應等資訊，皆被視為攻擊分析與異常偵測的重要依據。然而，這些日誌資料通常具備非結構化、高噪音與資料量龐大的特性，加上攻擊樣本與正常流量樣本比例嚴重失衡，皆為進一步分析帶來挑戰。隨著機器學習技術的不斷進步，許多研究者嘗試將監督式學習與分類模型應用於入侵偵測系統 (Intrusion Detection System, IDS)，以提升偵測效率與準確性。不過在實務應用中，仍面臨如資料來源不平衡、特徵選擇困難等技術問題。

因此，本研究旨在建立一套基於網路日誌的網路攻擊分類系統，並結合隨機森林 (Random Forest, RF) 演算法進行訓練與預測。我們以 Kaggle 公開資料集為主，Apache Malicious Log Generator 與 Fake Apache Log Generator 等開源工具為輔，產出多類型的攻擊樣本與正常樣本，建立平衡的訓練資料集，改善資料不平衡問題所造成的模型誤差。此外，本研究亦從資料集中擷取八項潛在特徵，並透過 Gini 特徵重要性分析進行消融實驗 (ablation study)，探討不同特徵組合對模型分類效能的影響。

以下第二章節將介紹本研究參考的相關文獻及使用技術，第三章將說明本研究之研究方法，第四章則是實驗探討，後續第五章與第六章為實驗分析與結論。

貳、文獻探討

2.1 Hadda 等人所提出的 Apache weblog 攻擊分類器 [3]

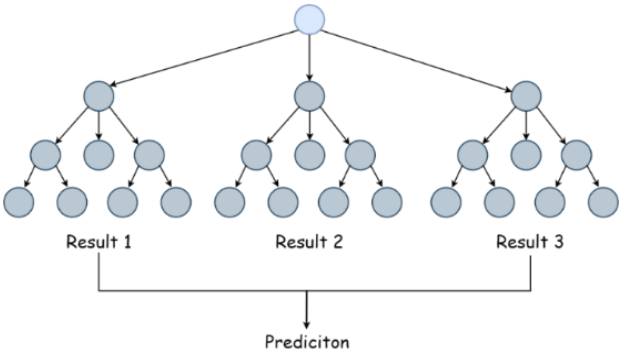
2022 年，Hadda 等人針對 Apache Web Server 的離線紀錄檔 (log) 進行分析，開發了一套能有效偵測惡意攻擊紀錄的機器學習的開源系統。研究以 Webhawk 工具 [11] 為基礎進行三大增強，其中包含資料集自動產生、進階正規表達式標記 (labeling) 機制，以及多種分類演算法的模型比較與精度選擇功能。該系統針對三種應用層攻擊 (SQL 注入 (SQLi)、跨站腳本攻擊 (XSS)、目錄遍歷 (DS)) 進行訓練與預測，他們利用多種模型相互比較，其中以隨機森林模型表現最佳，準確率為 98.8%，其他如決策樹 (Decision Tree, DT) 為 98.7%、極限隨機樹 (Extra Trees, ET) 為 98.5%，而多層感知機(Multi-Layer Perceptron, MLP) 及 K-近鄰演算法 (K-Nearest Neighbors, KN) 分別為 96.1%與 86.9%。該研究中也提及資料集在建構過程中存在資料不平衡 (data imbalance) 的問題，例如正常紀錄僅約 3,850 筆，相對於超過五萬筆的 SQL Injection 紀錄與 10 萬筆以上的 XSS 攻擊紀錄，模型學習可能會有誤差，需透過合適的預處理與模型評估方式加以彌補。

2.2 Saleem 等人所提出以機器學習為基礎的網頁伺服器攻擊偵測方法 [9]

Saleem 等人於 2020 年發表針對網頁伺服器常見攻擊 (SQL 注入、拒絕服務 (Denial of Service, DOS) 及跨站腳本攻擊) 提出一套基於機器學習的入侵偵測方法。該研究首先於私有網路環境中架設 Apache WAMP 伺服器，並利用 DVWA (Damn Vulnerable Web Application) 作為攻擊目標，透過實際發動攻擊來收集並建立包含正常流量與三種攻擊類型的日誌資料集。在特徵工程方面，該論文分別採用兩種方法，第一種為簡單分類技術，從 URL 查詢字串中手動萃取 26 項關鍵字特徵，以 0/1 表示特徵是否出現，並分別以決策樹、支持向量機 (SVM)、K-近鄰演算法及自適應增強 (AdaBoost) 等分類器進行訓練與測試。第二種為文字分類技術，將 URL payload 以 TF-IDF 方式轉換為特徵向量，並應用決策樹、支持向量機、自適應增強及多項式貝氏分類器 (Multinomial Naive Bayes) 等文字分類器進行多類別攻擊偵測。在實驗設計中，該論文將資料集分為訓練、測試與驗證集，並針對模型進行參數調整與 k-fold 交叉驗證。結果顯示，文字分頻技術下的決策樹與自適應增強模型表現最佳，於多項指標均約 0.98 以上，且在多類別分類上均有穩定且優異的表現。該研究亦指出，文字分類技術因考慮特徵數量較多，能有效提升攻擊偵測的靈敏度與準確性。

2.3 隨機森林演算法 (Random Forest Algorithm)

在模型建立階段，本研究使用隨機森林演算法來構建網頁攻擊分類器。隨機森林是一種集成學習演算法，常用於分類與回歸任務，通過多棵決策樹的組合來實現。正如圖一所示，隨機森林通過投票等方法構建多棵不同的決策樹，並生成或選擇最佳結果。在此框架下，我們採用來自 *sklearn.ensemble* 模組的隨機森林演算法來提升分類效能。



圖一：隨機森林演算法架構

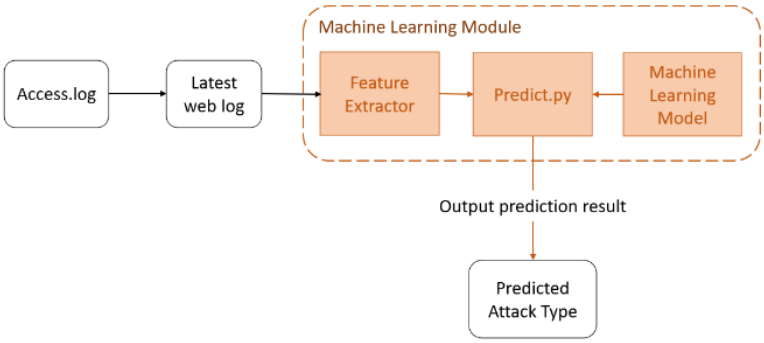
訓練過程中，我們首先從 CSV 檔案中載入標註資料集，其中特徵與標籤分開存放。為確保評估的公正性，資料集被劃分為 80%的訓練資料和 20%的測試資料。在使用訓練

資料集對隨機森林模型進行訓練後，我們應用準確度評分函數來評估其性能。隨後，訓練完成的模型被用於預測測試資料集中的攻擊類型，從而提供模型準確性與有效性的洞察。

為進一步提升模型的可解釋性，我們應用了混淆矩陣 (Confusion Matrix) 並生成分類報告。混淆矩陣通過比較實際標籤與預測標籤，詳細展示了模型的預測表現。這使得我們能夠清楚了解模型在識別特定攻擊類型 (如 SQL 注入 (SQLi)、跨站腳本攻擊 (XSS)、目錄遍歷 (DS)) 與正常流量方面的表現。另一方面，分類報告則提供了各類別性能的總結，詳細分析將在後續章節中進一步闡述。

參、提出的研究方法

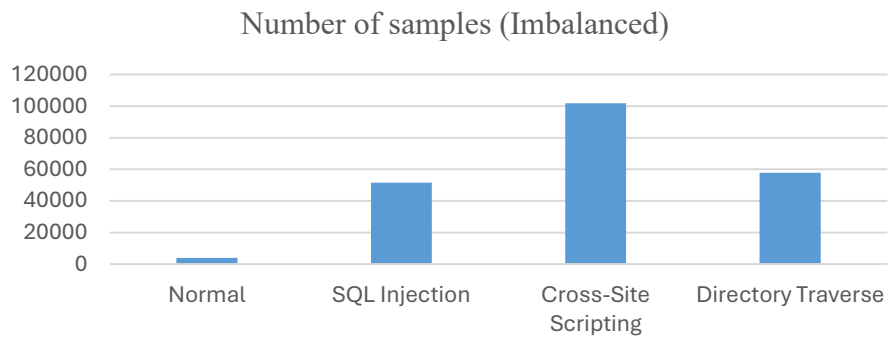
本節將詳細介紹我們的研究方法，呈現了整體的流程架構，如圖二。起初，請求的資料會被記錄在 Access.log 中，後續由腳本擷取最新的日誌內容，並將其傳入機器學習模組中，該模組包含特徵擷取器、預測腳本與預訓練模型，用以判斷攻擊類型。最後，模組預測的攻擊類型將做為系統的輸出結果，後續章節會進一步介紹細節。



圖二：架構流程圖

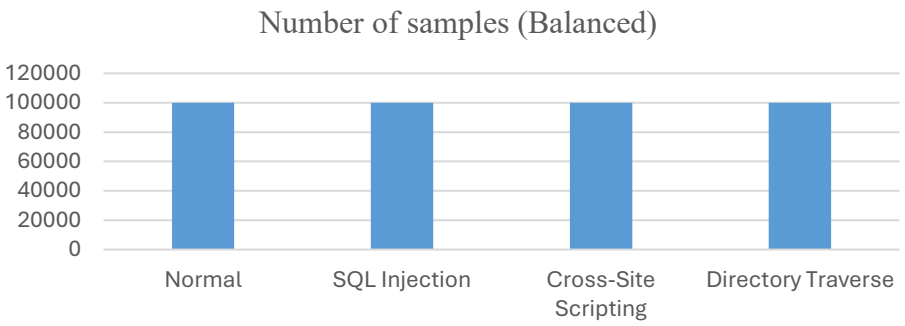
3.1 資料預處理 (Data Preprocessing)

本研究選擇了來自 Kaggle 的資料集 [5]，我們觀察到該資料集存在資料不平衡的問題。根據表一跟圖三可以明顯看出四個類別的樣本數有顯著的差異，且此類資料不平衡的狀態可能導致機器學習模型產生誤差的預測結果，因此，針對該問題進行調整對於提升分類準確性而言至關重要。



圖三：未經平衡的資料集

在資料處理的過程中，本研究透過使用 Apache-Malicious-Log-Generator [7]與 Fake-Apache-Log-Generator [6]產生模擬的網頁日誌資料，以建立一組平衡的資料集，最終生成之結果如圖四所示。



圖四：平衡後的資料集

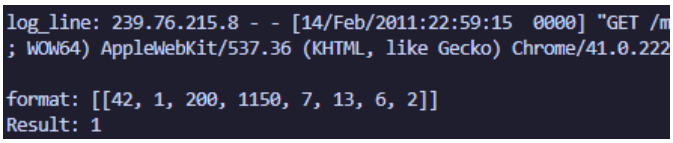
3.2 特徵擷取 (Feature Extraction)

為進行有效的攻擊偵測，本研究從網頁日誌資料中擷取下列表一陳述的八項潛在特徵，這些特徵皆來自於 HTTP 請求與回應內容，透過這些特徵，執行特徵擷取以判別資料中是否存在辨識價值，表三也呈現各潛在特徵的數量，以下附上一則經過處理後、由終端機輸出之日誌紀錄作為範例。終端機中顯示的輸出結果，如圖五。

239.76.215.8 - - [14/Feb/2011:22:59:15 0000] "GET /main9.php?name1=") or "1"="1 HTTP/1.1" 200 1150 "http://www.jones.info/mainpage/" "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2225.0 Safari/537.36"

表一: 八項潛在特徵之各項數值

特徵	數量
Length of the specific request	42
Number of parameters	1
Return code	200
The size of the request	1150
Number of upper cases	7
Number of lower cases	13
Number of special characters	6
The depth of directory for accessed resource	2
攻擊類型 (結果)	1 (SQLi)



圖五：輸出結果

3.2.1 Gini 特徵重要性 (Gini Feature Importance)

如表四，我們可透過上述的特徵擷取篩選出較有價值的資訊，因此，本研究採用 Gini 特徵重要性作為分類的評估依據。決策樹中的 Gini 指數 (Gini Index) 定義如下，s_t 為當前節點上的資料集，C 為類別總數，而 p_i 為樣本屬於第 i 類的機率。

$$Gini(s_t) = 1 - \sum_{i=1}^{C} p_i^2 \tag{1}$$

此外，為了在訓練過程中尋找每個節點的最佳切分點，Gini 不純度 (Gini impurity) 被用作提升最大化純度，其定義如下，t 代表當前節點，t_L 與 t_R 分別代表切分後的左子節點與右子節點， $p_L = \frac{n_{t_L}}{n_t}$ 與 $p_R = \frac{n_{t_R}}{n_t}$ 則為各子節點的樣本比例。

$$\nabla i(s_t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{2}$$

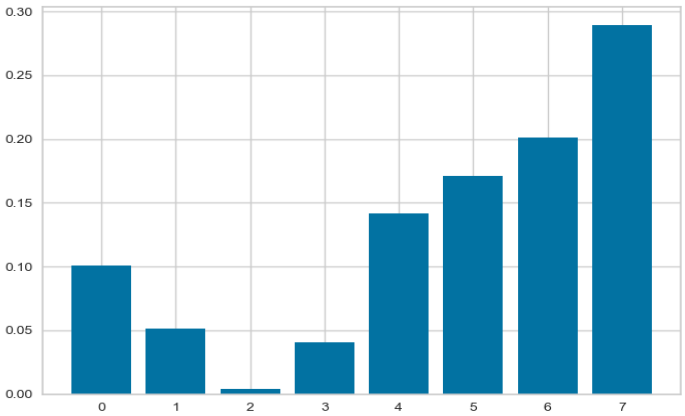
基於上述公式，我們可計算出 X_m 的 Gini 特徵重要性，其為整個隨機森林中所有決策樹加權後的不純度減少總和，其中 B 表示森林中樹的總數，$p(t)$ 為到達節點 t 的樣本比例， $v(s_t)$ 則為節點 (s_t) 使用的切分特徵。

$$Imp(X_m) = \frac{1}{B}\sum_{b=1}^{B}\sum_{t\in b:v(s_t)=X_m} p(t)\nabla i(s,t) \quad (3)$$

透過上述計算，我們可得出各特徵的重要性排序，進而輔助後續的特徵選擇與模型優化。我們將優先保留對模型影響力較高的特徵，以降低雜訊、提升分類準確率。表二呈現了各特徵重要性的分數，而圖六亦以長條圖展示了各特徵的 Gini 重要性，可作為判斷分類貢獻程度的重要依據。

表二: 各特徵重要性分數

No.	特徵	重要性分數
0	*Length*	0.10084
1	*param_number*	0.05141
2	*return_code*	0.00368
3	*size*	0.04070
4	*upper cases*	0.14166
5	*lower cases*	0.17132
6	*special chars*	0.20125
7	*depth*	0.28914



圖六：各特徵之 Gini 重要性分佈圖

根據計算結果可明顯觀察到，param_number、return_code 以及 size 三項特徵的重要性分數顯著低於平均水準。換言之，這些特徵對於模型分類的貢獻較低，故可於後續實驗中予以移除。如此將有助於降低計算複雜度、提升模型效能，並解決過擬合 (overfitting) 的問題。

肆、實驗方法

4.1 使用資料集

本研究所採用的資料集為 Apache Web Log Dataset [5]，源自於 Kaggle 平台。該資料集主要收錄自 Apache HTTP 伺服器的存取日誌，總計包含約 200,000 筆網頁請求紀錄。每一筆紀錄皆對應一個用戶端對伺服器的請求，詳細記錄了請求的相關資訊。資料集主要欄位說明如下：

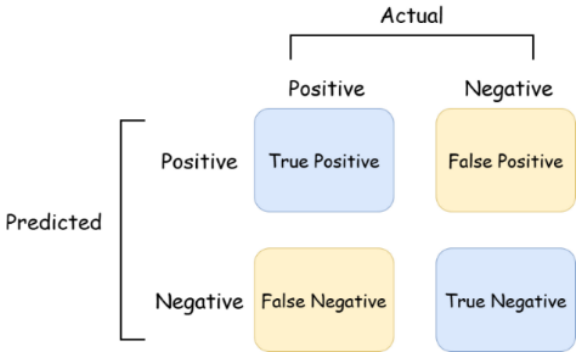
1. IP Address：發出請求的用戶端 IP 位址，可用於分析用戶來源及行為模式。
2. Timestamp：請求發生的日期與時間，有助於進行時間序列分析或流量高峰時段判斷。
3. Request Method：HTTP 請求方法，用於區分不同類型的操作。
4. URL：被請求的資源路徑，可分析熱門頁面或資源。
5. Status Code：伺服器回應的 HTTP 狀態碼，可用於判斷請求是否成功或發生錯誤。
6. User Agent：用戶端的瀏覽器或裝置資訊，有助於分析用戶設備分布。

本研究選用此資料集，主要因其具備豐富且多元的網頁存取紀錄，能夠模擬實際網站運作情境。透過分析這些日誌資料，可以有效進行網站流量分析、用戶行為模式挖掘，以及異常偵測 (如惡意攻擊、爬蟲行為等)。此外，資料集結構清晰便於進行資料前處理與特徵工程，適合用於機器學習與深度學習相關實驗。

4.2 評估方法

本節將說明本研究提出方法的實驗結果，如上節所述，我們採用混淆矩陣來視覺化模型在分類任務中的正確與錯誤預測結果。而混淆矩陣有四個分類指標，見圖七與解釋如下：

1. 真陽性 (True Positive, TP) 為模型正確預測實際為正類的樣本數。
2. 真陰性 (True Negative, TN) 指模型正確預測實際為負類的樣本數。
3. 假陽性 (False Positive, FP) 為模型誤將負類樣本預測為正類的樣本數。
4. 假陰性 (False Negative, FN) 是模型誤將正類樣本預測為負類的樣本數。



圖七：混淆矩陣

此外，本研究採用分類報告 (Classification Report) 用以呈現關鍵的評估指標，全面評估分類模型的效能表現。

1. 準確率 (Accuracy) 用以衡量模型正確預測的比例，公式如下：

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

2. 精確率 (Precision) 評估模型在預測為正類、實際為正類的比例，即模型避免誤判的能力，其公式為：

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

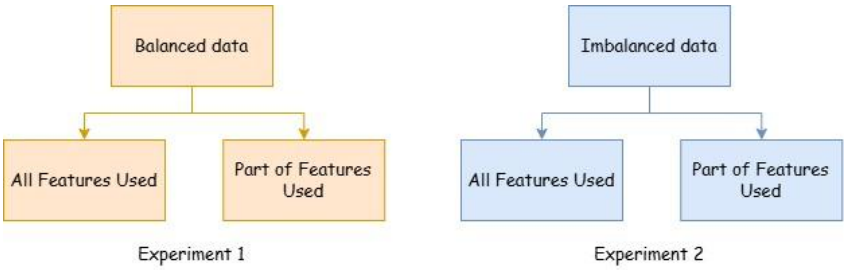
3. 召回率 (Recall)，又稱敏感度 (Sensitivity)，反映模型正確識別出實際正類樣本的能力，公式如下：

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

4. F1 分數 (F1-score) 為精確率與召回率的調和平均數，用以兼顧兩者的權衡，公式如下：

$$F1\ score = \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

在本研究的實驗設計中，我們將實驗分為兩組，旨在以不同的資料訓練不同的特徵組合。起初，我們選取了八個特徵作為初始輸入，隨後透過 Gini 重要性指標篩選出影響力最小的三個特徵，將其分批移除。整體實驗流程如下圖八所述：



圖八：實驗流程架構圖

本研究分別採用不同資料分布進行兩組實驗的比較。第一組為使用平衡資料集 (balanced dataset) 訓練的實驗組；第二組則是採用不平衡資料集 (imbalanced dataset) 的對照組。針對每一組資料集，我們分別在兩種情境下評估模型效能，一種是使用全部擷

取特徵，另一種使用部分篩選後的特徵。根據上述的說明，我們觀察到 *param_number*、*return_code* 與 *size* 等特徵的重要性分數相對較低。因此本研究依據其影響程度，分階段移除這三個特徵，並於不同資料集下進行測試，以評估特徵選擇對模型表現的影響。此外，本研究平衡資料集中的每一類別均包含 100,000 筆資料，而不平衡資料集的各類別分別包含 3,805 筆、51,528 筆、101,845 筆與 57,864 筆資料。表三顯示了平衡與不平衡資料集在樣本分布上的差異，後續我們將以上資料集作為訓練基礎，進行機器學習模型的訓練以預測不同的攻擊類型。

表三: 平衡與不平衡資料集的樣本分布差異

類別	不平衡樣本數	平衡樣本數
Normal	3805	100000
SQLi	51528	100000
XSS	101845	100000
DS	57864	100000

伍、實驗結果與分析

本節我們將呈現本研究的實驗結果，將一併展示針對平衡與不平衡資料集在不同特徵選擇條件下所產生的混淆矩陣與分類報告。此外，我們也將分別說明模型在各情境下的準確率、召回率、精確率與 F1 分數。根據前述的分析，我們將特別關注表四分數最低的三個關鍵特徵作為模型效能評估的依據。

表四: 重要性分數最低的三個特徵

No.	特徵	重要性分數
(1)	*param_number*	0.05141
(2)	*return_code*	0.00368
(3)	*size*	0.04070

根據本研究方法的實驗結果，我們進一步進行了消融研究 (ablation study) 的分析。無論資料集為平衡或不平衡，相較於 Hadda 等人所提出之系統，本研究透過資料平衡處理與特徵重要性分析，有效提升模型效能，皆在多項指標上優於該研究。在表五的第一組使用平衡資料集的實驗中，我們發現當移除特徵 (3)，即 size 特徵的模型表現最佳，其準確率達到 0.9949，是所有實驗設定中最高者，顯示該特徵在模型中的影響可能較為干擾。另一方面，在表六的第二組使用不平衡資料集的實驗中，可明顯觀察到整體訓練效能略低於平衡資料集。儘管如此，幾乎所有設定的準確率仍穩定落在約 0.98 左右。而即使在不平衡的資料條件下，移除特徵 (3) 仍為該組表現最佳的設定，其準確率達 0.9906，再次驗證該特徵的移除對提升模型效能具有一致性效果。此外，由表七能直觀地觀察到本研究對參考文獻之研究改良後，無論在準確率、精確率、召回率又或是 F1 分數的結果皆有成長。

表五: 平衡資料集之效能分析

	使用所有特徵	移除 (1, 2, 3)	移除 (1)	移除 (2)	移除 (3)	移除 (1,2)	移除 (2, 3)	移除 (1, 3)
準確率	0.99378	0.9923	0.9906	0.9939	**0.9949**	0.9909	0.9926	0.9945
精確率	0.9975	0.99225	0.993	0.99375	0.99475	0.99075	0.9945	0.99225
召回率	0.99375	0.99225	0.9905	0.994	0.995	0.99075	0.9945	0.99275
F1 分數	0.99375	0.99225	0.9905	0.99375	0.99475	0.91575	0.9945	0.9925

表六: 不平衡資料集之效能分析

	使用所有特徵	移除 (1, 2, 3)	移除 (1)	移除 (2)	移除 (3)	移除 (1,2)	移除 (2, 3)	移除 (1, 3)
準確率	0.9886	0.9882	0.9857	0.9887	**0.9906**	0.9857	0.9903	0.9883
精確率	0.98025	0.98025	0.978	0.98075	0.9815	0.97775	0.98225	0.98
召回率	0.98975	0.98725	0.987	0.99	0.991	0.987	0.98875	0.9895
F1 分數	0.985	0.9835	0.98275	0.9855	0.98625	0.9825	0.9855	0.98475

表七: 不平衡資料集之效能分析

		The proposed method	Hadda et al.'s method [3]	Saleem et al.'s method [9]
準確率		99.49%	98.8%	98%
精確率	Normal	99.8%	95.6%	97%
	SQL Injection (SQLi)	99.3%	98.1%	
	Cross-Site Scripting (XSS)	99.1%	99.1%	
	Directory Traverse (DS)	99.7%	99.3%	
召回率	Normal	99.9%	99%	98%
	SQL Injection (SQLi)	99.3%	98.6%	
	Cross-Site Scripting (XSS)	99.2%	98.9%	
	Directory Traverse (DS)	99.6%	99.1%	
F1 分數	Normal	99.9%	97.3%	98%
	SQL Injection (SQLi)	99.3%	98.3%	
	Cross-Site Scripting (XSS)	99.1%	99%	
	Directory Traverse (DS)	99.6%	99.2%	

陸、結論

本研究提出了一套利用隨機森林演算法並以 Kaggle 公開資料集進行訓練與評估的網路攻擊分類系統。為了解決資料不平衡問題，本研究使用惡意 Apache 日誌產生器來合成額外的記錄藉此平衡整體資料集。隨後，從產生的日誌資料中擷取八個特徵，並採用 Gini 特徵重要性指標進行評估，也在最後進行了消融實驗，以探討在不同特徵選擇條件下模型的效能變化。就實驗結果而言，本研究所提出之模型達到最佳效能，其準確率為 0.9949，精確率為 0.99475，召回率為 0.995，F1 分數為 0.99475。而這些結果顯示這不僅提升了分類效能，亦展現出應用於網路入侵偵測場景的潛在實用性。

參考文獻

[1] E. E. Abdallah, W. Eleisah, and A. F. Otoom, “Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey,” *Procedia Computer Science*, vol. 201, pp. 205-212, 2022.

[2] E. M. Abu Hadda, H. M. Abu Sada, and E. S. Sallouha, “Offline Log Analysis f or Apache Web Servers Using Machine Learning,” Dept. of Information Security Engineering, University of Gaza, 2022.
Available: https://drive.google.com/file/d/1cexbOLLV-fggL6pYRBOqUjwFO9snJrSd/vi ew.

[3] K. Basu, “Fake-Apache-Log-Generator,” *GitHub*, Mar 16, 2018.
Available: https://github.com/kiritbasu/Fake-Apache-log-Generator.

[4] W. DABOUBI, “Intrusion-and-anomaly-detection-with-machine-learning,” GitHub.
Available: https://github.com/slrbl/Intrusion-and-anomaly-detection-with-machine-learn ing

[5] T. Gaber, J. B. Awotunde, M. Torky, S. A. Ajagbe, M. Hammoudeh, and W. Li, “Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoT networks,” *Internet of Things*, vol. 24, 2023, 100977.

[6] J. M. Kim, “Apache Web Log," *Kaggle*,” 2022.
Available: https://www.kaggle.com/datasets/kimjmin/apache-web-log.

[7] S. G. McLabraid, “Apache-Malicious-Log-Generator," *GitHub*. May 17, 2021.
Available: https://github.com/McLabraid/Apache-Malicious-Log-Generator.

[8] S. Neupane et al., “Explainable Intrusion Detection Systems (X-IDS): A Survey o f Current Methods, Challenges, and Opportunities,” in *IEEE Access*, vol. 10, pp. 112392-112415, 2022, doi: 10.1109/ACCESS.2022.3216617.

[9] S. Saleem, M. Sheeraz, M. Hanif and U. Farooq, "Web Server Attack Detection using Machine Learning," *2020 International Conference on Cyber Warfare and S ecurity (ICCWS)*, Islamabad, Pakistan, pp. 1-7, 2020, doi: 10.1109/ICCWS48432.20 20.9292393.

[10] A. K. Silivery, R. M. R. Kovvur, R. Solleti, L. K. S. Kumar, and B. Madhu, "A model for multi-attack classification to improve intrusion detection performance us ing deep learning approaches," *Measurement: Sensors*, vol. 30, 2023, 100924.

[11] H. Teymourlouei and V. E. Harris, "Preventing Data Breaches: Utilizing Log Anal ysis and Machine Learning for Insider Attack Detection," *2022 International Conf erence on Computational Science and Computational Intelligence (CSCI)*, Las Veg as, NV, USA, pp. 1022-1027, 2022, doi: 10.1109/CSCI58124.2022.00181.