結合語言模型與特徵機制之整合式網路入侵偵測告警系統

壹、 摘要

隨著網際網路的急速發展，我們現在能夠透過多種網路搜尋引擎（例如 Google [1]）與最近熱門的生成式 AI 聊天機器人（例如 ChatGPT [2]）來輕鬆地取得所需的資訊或回答，這無疑使我們的生活更加便利。然而，隨之而來的是各種安全威脅和攻擊，像是釣魚攻擊（Phishing）、分散式阻斷服務（Distributed Denial-of-Service, DDoS）、SQL 注入攻擊（SQL injection）等，這些都是我們在網路上時常會面臨的風險。只要我們使用網際網路，就會時刻面對這些風險與威脅。因此，在享受網路便利的同時，確保個人安全已成為一個極為重要的課題。為了解決這個問題，本提案計劃利用 CICIDS2017 資料集[3]來訓練一個自然語言處理 (Natural Language Processing, NLP) 模型。這個模型將能夠有效地分辨我們提供的資料是否包含惡意攻擊。同時，本提案欲結合 Wireshark [4]來進行定期監控骨幹網路中的封包，並將擷取到的資料經過一些處理後傳送給我們訓練出來的模型。透過這種方式，我們可以判斷流量中是否存在具惡意行為的封包。一旦檢測到惡意封包，系統將向管理者發送警告，告知其惡意流量的相關資訊，提醒管理者注意封包來源。這項研究成果有望對網路安全防護領域提供更有力的支援。

關鍵字: 網路安全、入侵檢測系統、機器學習、語言模型、CICIDS2017

貳、 研究動機與研究問題

隨著科技的迅速發展，網路攻擊手法變得更加複雜與隱匿。舉例來說，常見的網路釣魚攻擊，利用了電子郵件或簡訊等方式夾帶偽造的網站鏈結以誘導目標進行操作，試圖竊取敏感資訊或引導目標安裝惡意軟體。另一種常見例子是分散式阻斷服務攻擊，這種攻擊利用大量的殭屍機器向目標網站或伺服器發送大量請求封包，使其無法正常運作。除此之外，還有 SQL 注入攻擊。這是一種常見且具有破壞性的攻擊手法。在 SQL 注入攻擊中，攻擊者通過向應用程式的輸入字串中，試圖串接惡意的 SQL 指令，從而對資料庫進行非法存取或執行惡意操作。這種攻擊不但會導致敏感資料的洩露、資料庫的破壞，甚至是整個系統的控制權被攻擊者所掌握 [5]。

入侵檢測系統 (Intrusion Detection System, IDS) 是一種用於監控網路或系統活動的安全工具。它的主要功能是檢測並辨識潛在的惡意活動或安全事件，例如未經授權的存取、異常流量、惡意軟體、以及駭客入侵行為。 IDS 通常部署在骨幹網路邊界或主機系統上，並持續監控網路流量、日誌記錄以及系統和應用程式的行為。 IDS 的目標是在發生安全事件時及早發現並警告管理

員，以減輕損害並防止進一步的入侵。它是維護系統和網路安全的重要工具之一，與其他安全措施如防火牆、防毒軟體等常一起部署，以提供綜合的安全保護 [6]。

傳統的網路安全防護手段，例如防火牆、入侵檢測系統等，雖然可以對抗一些較簡單的攻擊，但對於新型態的威脅卻顯得力不從心。例如，近年來出現的隱蔽性高且難以檢測的零時差漏洞攻擊 (Zero-Day Vulnerability) [7]，這種攻擊利用系統或應用程式中未被發現的漏洞，從而進行未知的攻擊，傳統的防護手段往往難以及時發現和防禦。

本研究的動機在於提高網路入侵偵測的效能與精準度，有效地來保護使用者的網路安全。傳統的 IDS 往往受限於僅能捕捉靜態特徵，對於動態和隱匿性攻擊的偵測能力有所不足。因此，透過機器學習技術結合語言模型，我們能夠更全面地捕捉網路封包中的關鍵特徵，並對其進行有效分析。

然而，這一研究面臨著多重挑戰。首先，如何**建構一個有效的流量分析模型**，以捕捉網路封包中的重要特徵是一個關鍵問題。傳統文獻的特徵提取方法無法應對動態攻擊的變化，需要探索新的建模和優化技術。其次，如何**整合語言模型**，提高偵測系統對新型態攻擊的適應能力也是一個重要挑戰。新型攻擊的快速演變需要我們不斷更新和優化偵測模型，以確保其準確性和效能。此外，如何**設計特徵提取機制**，以更細緻地分析封包內容，區分不同類型的攻擊也是一個關鍵問題。攻擊者通常會採取各種手段來隱匿其行蹤，因此我們需要設計更加靈活和有效的特徵提取和分析方法。最後，如何**實作一個能夠即時偵測並發出警告的入侵偵測告警系統**，讓使用者能夠在攻擊發生時迅速做出反應與處理是我們所追求的目標。

總而言之，本計畫旨在利用機器學習技術結合語言模型，開發或增強一個更為有效且靈活的網路入侵偵測告警系統，以應對不斷變化的網路攻擊威脅。然而，實現這一目標面臨著多重挑戰，包括建構有效的流量分析語言模型、整合語言模型提高偵測系統對新型態攻擊的適應能力、設計特徵機制更細緻地分析封包內容，以及實作即時偵測並發出警告的入侵偵測告警系統。透過克服這些挑戰，我們將提升網路安全防禦的能力，為使用者提供更加安全可靠的網路環境。

參、 文獻回顧與探討

為深入了解語言模型與特徵機制之整合式 IDS 研究現況與關聯影響，翻閱許多文獻和研究。本計畫使用網路安全、入侵檢測系統、機器學習、語言模型、 CICIDS2017 等不同關鍵字組合，搜尋與本計畫相關之論文，並挑選重要的期刊論文進行研讀。

本計畫將使用網路入侵偵測系統的標竿資料集 CIC-IDS2017 (Canadian Institute for Cybersecurity Intrusion Detection Systems 2017) [8]。 CIC-IDS2017 資料集是由加拿大網路安全研究院 (Canadian Institute for Cybersecurity) 於 2017 年釋出，用於評估入侵檢測系統 (IDS) 和入侵防禦系統 (IPS) 效能。它解決了現有資料集缺乏流量多樣性和即時性的問題，提供了包括良性和最新攻擊在內的真實世界流量。該資料集的生成過程包括了使用 B-Profile 系統生成自然的良性背景流量，以及在特定時間段內施行不同類型的攻擊，如 Brute Force FTP、Brute Force SSH、DoS、Heartbleed、Web Attack 等。資料集還包括了攻擊和良性流量的標記資料，以及從產生的網路流量中提取的 80 多個網路流量特徵。這些特徵可作為訓練入侵檢測系統的依據，幫助提高系統的準確性和效能[3]。該資料集的建立符合可靠的基準，包括完整的網路設定、完整流量捕獲、標記資料集、完整的互動和攻擊多樣性等要求[9]。

Ahmad 等人[10] 於 2020 年回顧了基於 Machine Learning, ML 和 Deep Learning, DL 方法的網路入侵檢測機制。研究顯示，大約 80% 的提出的解決方案都是基於 DL 方法，其中 Autoencoder 和 Deep Neural Network 是最常用的演算法，不過基於 ML 的解決方案較簡易、需要較少的運算資源。研究亦提出強調了訓練時應使用 CSE-CIC-IDS2018 等較新的資料集，最後，本文強調了在實現低頻攻擊的模型性能和減少提出模型的複雜性方面的研究差距，並表示未來研究的方向可能為使用較不複雜的 DL 演算法、並盡量確保模型的有效性。 Dini 等人[11] 於 2023 年對機器學習在入侵檢測系統中的應用做了深度的研究，並著重於資料集、演算法和性能。結果顯示，使用決策樹和隨機森林選定資料集時，無論是二元分類還是多類分類，始終表現優於其他模型。此外，研究探索了深度學習模型的潛力，強調了其在各種任務中的有效性，並且對於高流量網路極其重要，也可以整合進嵌入式系統中。 Kumar 等人[12] 在 2010 年討論了人工智慧技術在入侵檢測系統中的應用及優點和限制，並提出需要被解決的問題包含自動調整高效率的辨識技術、特徵縮減技術、高速處理即時分析、低強度攻擊識別。 Dr. Tiberiu-Marian Georgescu [13] 在 2020 年討論了關於自然語言處理在自動分析與資安相關文件上的應用，包含了如何利用自然語言處理技術來分析與資安相關的文件，例如安全漏洞報告、威脅情報、安全策略等。 Benyamin Ghojogh [14] 等人在 2019 年發布的論文中回顧了不同的常見特徵選擇和提取方法的理論和動機，並介紹了它們的一些應用，以及展示了這些方法的一些數值實現。最後，將幾種方法進行了比較。

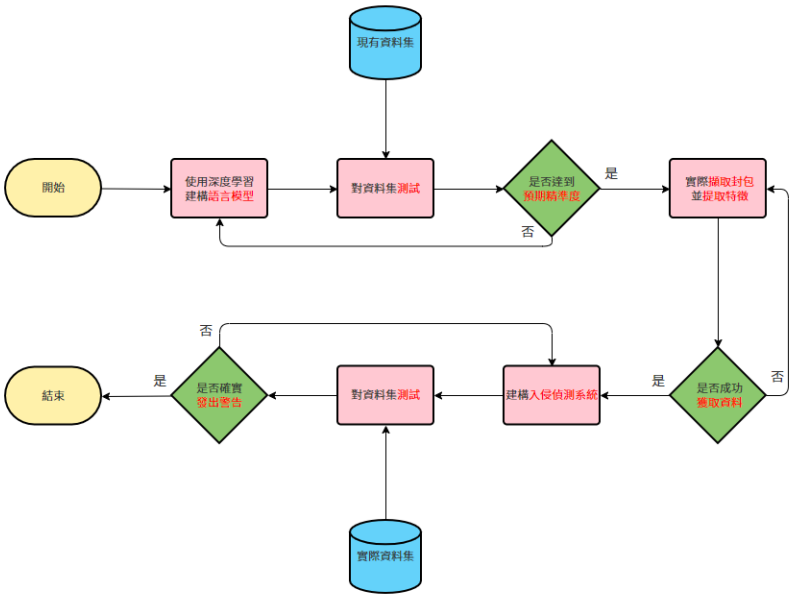
透過查閱上述論文得知， IDS 系統在當今的網路安全中扮演著至關重要的角色。了解到訓練資料集的選擇至關重要，因此我們以 CICIDS2017 為核心、佐以其他資料集，以提高模型的準確性和普遍性。未來研究方向包括高速即時分析、

使用更簡單的演算法及輕量的模型，並且本次研究將嘗試模仿出一套類似的系統。

肆、 研究方法及步驟

一、研究流程

本研究旨在利用機器學習技術，結合特徵機制與語言模型，開發一個能夠有效偵測網路中惡意流量並傳送警告的系統。圖一為本研究流程圖。



圖一:研究流程圖

首要任務是建構一個高效的流量分析語言模型，通過對現有資料集的測試，確保模型達到預期的準確度水準。在此階段，我們採用了 Python 語言中的 SimpleTransformers 套件，並評估不同的語言模型，如 BERT、RoBERTa、ALBERT 等，觀察它們在相同參數和條件下，針對同一份資料集進行訓練的效果是否存在明顯差異，並比較其優缺點。所使用的資料集為 CICIDS2017，該資料集提供了包括良性和最新攻擊在內的真實世界流量，可有效評估各模型在訓練方面的成效。在評估過程中，我們將主要使用混淆矩陣、精確率、召回率、ROC 曲線、F1 score 等指標，以評估各模型的性能表現，從而更全面地了解各模型的數據，最後決定最適合本研究計畫的語言模型，此語言模型必須有 0.99 以上的 F1 score。隨後，將實際擷取的封包進行內容分析，以提取多種特徵。獲取資料後，將這些特徵與語言模型相結合，對封包進行深入分析。在此階段，我們首先透過 Wireshark 進行定期的排程監控，每隔 3 分鐘捕獲一次封包。接下來，使用 SplitCap 工具將封包進行處理和轉換，以便後續的使用與分析。接著，利用 CICFlowMeter 提取特徵和數據，並以 CSV 格式輸出，並對資料進行預處理，包括處理缺失值、無限大值以及特徵縮放，以確保資料的完整性。最後，建構一個 IDS 系統，當系統檢測到疑似惡意流量時，將立即向

使用者發出警告，以加強網路安全防禦的效果。在此階段，我們利用 Python 的 Tkinter 套件在 Windows 系統上建立彈出式視窗，在彈出式視窗中，我們將包含有關檢測到的惡意流量的相關內容，包括時間戳記、攻擊手法、來源 IP 地址、目標連接埠、通訊協議等等。通過呈現這些信息，幫助使用者及時應對潛在的安全風險。

二、流量占比不平衡之預處理

CIC-IDS2017 資料集涵蓋了各種不同類型的網路活動，旨在模擬現實世界中的網路攻擊和正常流量情境。其中包括正常流量以及 14 種不同種類的攻擊，相對應的資料與筆數如表一所示。然而，由於正常流量占比較多，而一些惡意流量的數量相對較少，導致實驗結果未達預期效果。

為解決這種不平衡的情況，我們對資料集進行了預處理，將資料按標籤分為 15 個子資料集。接著，從 14 種不同攻擊類型的資料集中分別選取 80% 作為訓練集， 20% 作為測試集，同時調整正常流量和惡意流量的比例為 1:1 ，相對應的資料與筆數如表二所示。最後，在每次執行時，對資料進行打亂處理，確保了流量占比和資料分布的平衡性。這樣的處理方式有助於提升實驗結果的準確性和可信度。

表一: CICIDS 2017 攻擊資料集的資料分布情形

標籤 (Label)	樣本數 (Samples)	佔比 (Composition)
BENIGN	2273097	80.301%
DoS Hulk	231073	8.163%
PortScan	158930	5.615%
DDoS	128027	4.523%
DoS GoldenEye	10293	0.364%
FTP-Patator	7938	0.281%
SSH-Patator	5897	0.209%
DoS slowloris	5796	0.205%
DoS Slowhttptest	5499	0.195%
Bot	1966	0.07%
Web Attack-Brute Force	1507	0.054%
Web Attack - XSS	652	0.024%
Infiltration	36	0.002%
Web Attack- Sql Injection	21	0.001%
Heartbleed	11	0.001%
Total	2830743	100%

表二:調整後 CICIDS 2017 攻擊資料集的資料分布情形

標籤 (Label)	樣本數 (Samples)	佔比 (Composition)
BENIGN	557649	50%
DoS Hulk	231073	20.719%
PortScan	158930	14.25%
DDoS	128027	11.479%
DoS GoldenEye	10293	0.923%
FTP-Patator	7938	0.712%
SSH-Patator	5897	0.529%
DoS slowloris	5796	0.52%
DoS Slowhttptest	5499	0.493%
Bot	1966	0.176%
Web Attack-Brute Force	1507	0.135%
Web Attack - XSS	652	0.058%
Infiltration	36	0.003%
Web Attack- Sql Injection	21	0.002%
Heartbleed	11	0.001%
Total	1115295	100%

三、建構流量分析語言模型

在建構語言模型的過程中，我們採用了 Python 語言中的 SimpleTransformers 套件[15]。 SimpleTransformers 是一個基於 Transformers 和 Hugging Face 庫的工具，專為解決自然語言處理 (NLP) 任務而設計。它提供了一個簡單使用的介面，讓使用者能夠輕鬆地建立、訓練和部署各種 NLP 模型，如文本分類、情感分析、命名實體識別等。這個工具具有幾個主要特點。首先，它的易用性非常高，使用者可以通過簡單而直觀的 API 快速開始建立和訓練模型，無需深入了解 Transformers 和機器學習的細節。其次，它支援多種 NLP 任務，涵蓋了文本處理的各個方面。它基於 Hugging Face 的 Transformers 庫，使用預訓練的 Transformer 模型，如 BERT、RoBERTa、ALBERT 等，從而具有強大的性能和擴展性。同時， SimpleTransformers 支援在 GPU 和 TPU 上進行模型訓練，能夠加速訓練過程，提高效率。最後，它提供了豐富的示例代碼和詳細的文檔，幫助使用者快速上手並解決問題。

在模型的選擇上，我們預計優先嘗試 BERT 和 RoBERTa 等模型，因為這兩個模型在語言模型領域具有較大的影響力和廣泛的應用。此外，我們也考慮嘗試 ALBERT 模型，因為我們在後續的實作方面需要一個更輕量級、更小巧，同時依舊保持較高性能水平的語言模型，以提高整體效率。最後，使用相同的參數設置，包括 Epoch、Batch size、Iteration 等等，觀察不同模型所訓練

出來的結果，並決定最適合我們需求的訓練模型。

BERT (Bidirectional Encoder Representations from Transformers) 模型其特色為一種雙向的預訓練語言模型，通過雙向遞歸神經網路 (Bidirectional Transformer) 從大量無標籤文本中學習詞彙的表徵。 BERT 使用了 Transformer 的多層編碼器，這些編碼器能夠將輸入序列轉換為一系列隱藏表示，同時保留了序列中每個位置的上下文信息。而 BERT 進一步將這些表示稱為上下文敏感的「Token Embedding」，並通過將隱藏表示與固定的位置嵌入結合，以保持位置信息。這樣的設計改變了傳統的從左到右或從右到左的單向模型，使 BERT 模型能夠理解整個文本序列的上下文和關聯性，並在處理 NLP 任務時表現出色。 BERT 模型通常通過預訓練和微調兩個階段來使用。在預訓練階段，模型通過大量未標記的文本數據進行自我監督訓練，從而學習到語言的通用表示。在微調階段，模型通過在特定任務的標記數據上進行微調，以適應該任務的特定要求。

RoBERTa (Robustly optimized BERT approach) 是一種基於自注意力機制的預訓練語言表示模型。它是基於 BERT 模型進行優化和擴展的，旨在解決 BERT 中存在的一些問題並提高性能。 RoBERTa 通過使用更長的訓練時間、更大的訓練數據集、移除下一句預測任務、動態調整訓練參數等方式來改進 BERT 。

ALBERT (A Lite BERT) 是一種基於 BERT 模型的輕量級版本。它針對 BERT 存在的一些問題進行了優化，旨在提高模型的效率和性能。 ALBERT 通常具有比原始的 BERT 模型更小的模型尺寸和更高的效率，同時在各種自然語言處理任務中表現出色。 ALBERT 通常會對 BERT 模型的一些關鍵組件進行改進，如詞彙表大小、嵌入維度、隱藏層大小等，以實現更好的性能和更高的效率。

在觀察不同模型所訓練出來的結果時，我們主要透過混淆矩陣 (Confusion Matrix) 、精確率 (Precision) 、召回率 (Recall) 、 ROC 曲線以及最重要的 F1 score，判斷各種模型的效率，最後選出效率最高的模型。混淆矩陣(如圖二)用於比較模型的預測結果與真實情況之間的差異。其中，混淆矩陣的行表示真實的類別，列表示模型預測的類別。總共分為四種結果，分別是真陽性 (True Positive, TP) 、真陰性 (True Negative, TN) 、偽陽性 (False Positive, FP) 、偽陰性 (False Negative, FN)。透過這些結果，能夠計算出多項機器學習模型的評估指標。例如，Precision (如公式 1)表示在預測正向的情況下，正確預測的機率；而 Recall (如公式 2) 則表示在實際為正向的情況下，被正確預測的機率。最後，兩者的調和平均數 (harmonic mean) 也稱為 F1 score (如公式 4)，則被視為是該二指標的綜合

指標，能更全面地評斷模型的表現。

另外，ROC 曲線 (Receiver Operating Characteristic curve) 是用於評估分類模型效能的圖形工具。其中，橫軸表示偽陽性率 (False Positive Rate, FPR) ，縱軸表示真陽性率 (True Positive Rate, TPR) 。真陽性率也就是上述所提過的 Recall ，所以計算公式也跟公式 2 一致，表示在實際為正向的情況下，被正確預測的機率；偽陽性率則表示在實際為負向的情況下，被錯誤預測為正向的機率，計算公式為公式 3。 ROC 曲線下方的面積 (Area Under the Curve, AUC) 則是評估分類器性能的重要指標， AUC 越接近 1 ，表示模型性能越佳(詳見圖四)。大致可分為四種結果，分別是準確性較高、中等、較低、不良。透過這些結果，能夠展現模型的性能表現，利於觀察。

真實 / 預測	實際正向	實際負向
預測正向	True Positive (TP)	False Positive (FP)
預測負向	False Negative (FN)	True Negative (TN)

圖二:混淆矩陣 (Confusion Matrix)

圖三: ROC 曲線

ROC曲線下面積	判斷結果
0.9以上	準確性較高
0.70～0.90	準確性中等
0.50～0.70	準確性較低
0.5以下	準確性不良

圖四: ROC 曲線下面積的判斷標準

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

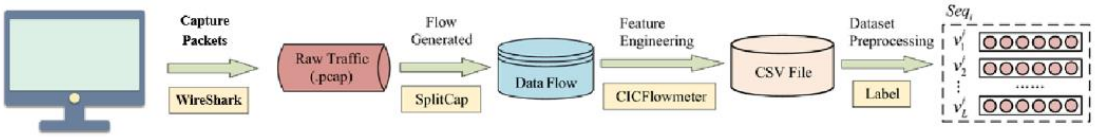
$$FPR = \frac{FP}{FP + TN} \tag{3}$$

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

四、實際擷取封包與提取特徵

在此階段中，我們預計使用到 Wireshark、SplitCap 和 CICFlowMeter (Canadian Institute for Cybersecurity Flow Meter) ，流程步驟如圖五所示，細節如下：

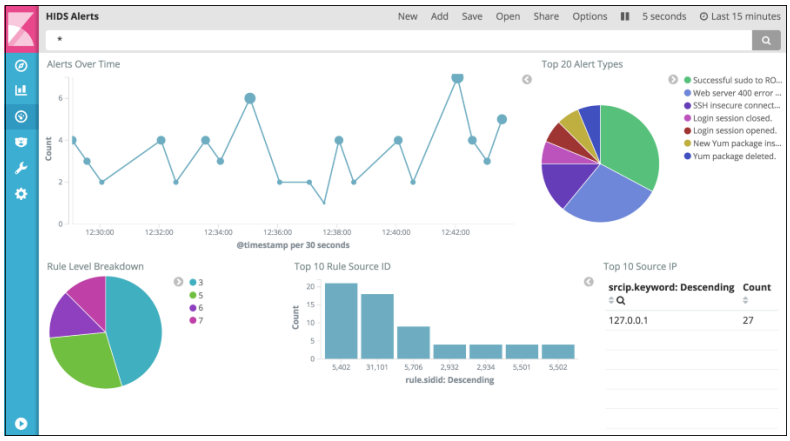
首先，我們將使用 Wireshark 工具，它是一款開源的網路封包分析軟體，它可以捕獲和分析經過計算機網路的數據封包。我們用它進行定期的排程監控，每隔 3 分鐘捕獲一次封包，並將捕獲的封包保存為 pcap 檔案。第二步，我們將使用 SplitCap 工具，它是用來分割 pcap 檔案的工具，對檔案進行處理或轉換，提高 pcap 檔案的可用性。因為 pcap 檔案可能非常大，難以處理或傳輸。SplitCap 可以幫助將大的 pcap 檔案分割成多個較小的文件，以便更輕鬆地處理和分析。第三步，我們使用 CICFlowMeter 工具，從這些 pcap 檔案中提取封包的特徵訊息。 CICFlowMeter 是一種專用於分析網路流量的工具，其功能包括監控和捕獲經過網路設備的封包，並提取各種有用的特徵和數據。這些特徵包括了 80 多個維度，並將其以 CSV 表格的形式進行導出。其具體工作原理是從 pcap 檔案中逐一讀取封包，將每個未完成的 TCP 和 UDP 流量封包添加到對應的流量中。因此，擷取的訊息是以 TCP flow 或 UDP flow 作為單位。 TCP flow 以 FIN 標誌來判斷結束，而 UDP flow 則根據設置的 flowtimeout 進行結束的判斷。最後，我們將進行資料的前處理。我們的主要任務是處理缺失值或無限大值，以及對類別數據進行處理和特徵縮放。在我們獲得的 CSV 檔案中，已經處理了類別和特徵相關的問題，因此我們主要處理缺失值或無限大值，確保每個字串都沒有缺失值或無限大值。由於在處理這部分常見的手法就是丟棄或是補值，但在我們的實作中要確保每一筆流量都能被正確紀錄，因此我們選擇的是補值。根據其標籤來進行常數填補 (補 0) 或前向、後向填補，確保其值是符合我們需求的格式以保障判斷的正確性。



圖五:實際擷取封包與提取特徵之流程圖

五、實作入侵檢測系統 (IDS)

在此階段中，我們利用 Python 的 Tkinter 套件在 Windows 系統上建立彈出式視窗，作為我們入侵檢測系統的使用者介面，圖六為示意圖。 Tkinter 作為 Python 的標準 GUI 工具包，提供了豐富的元件和功能，使我們能夠輕鬆地設計和實現使用者友好的圖形使用者介面。



圖六：IDS 使用者介面示意圖

在彈出式視窗中，我們將包含有關檢測到的惡意流量的相關內容。這些內容可能包括時間戳記、攻擊手法、來源 IP 地址、目標連接埠、通訊協議等等。通過呈現這些信息，使用者可以快速了解到系統檢測到的潛在安全威脅，從而採取適當的措施加強網路安全防護。

實作上，我們將利用 Tkinter 提供的元件和佈局管理器 (Layout Managers) 來設計視窗的外觀和佈局。我們將透過 Python 的編程能力來處理彈出視窗的行為，例如何時彈出視窗、視窗的大小和位置等。同時，我們還會使用 Python 來處理和顯示從入侵檢測系統中獲取的惡意流量數據，確保彈出的視窗能夠即時且準確地顯示相關信息，幫助使用者及時應對潛在的安全風險。

伍、 預期結果

我們的目標成果分成以下幾個面向：

- 在準確度方面，訓練出一個高度精確的異常檢測模型，並保證在基於 CICIDS2017 資料集的沙盒測試中達到 f1 score 大於或等於 0.99 。
- 在模型量級方面，逐步調整訓練模型的參數，盡量減少模型收斂的時間、加速訓練流程。
- 在佈署與應用方面，開發定時監控功能，並提供告警系統。可根據使用者需求訂製異常條件，以提供更靈活的安全設置。

陸、 需要指導教授指導內容

- 研究方向與專業領域探索。
- 相關機器學習套件的使用與教學理論推導。
- 成果評估與問題解決
- 會議或是論文的撰寫方法與修改論文。

柒、 參考文獻

[1] “What is Google?” Computer Hope, 2024. [Online]. Available: https://www.computerhope.com/jargon/g/google.htm. [Accessed: 16 2 2024].

[2] “Introducing ChatGPT” OpenAI, 2022. [Online]. Available: https://openai.com/blog/chatgpt. [Accessed: 16 2 2024].

[3] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018

[4] “What Is Wireshark and How Is It Used?” CompTIA, Inc., n. d. [Online]. Available: https://www.comptia.org/content/articles/what-is-wireshark-and-how-to-use-it. [Accessed: 16 2 2024].

[5] “Common cyberattacks to look out for” Zoho Corporation Pvt. Ltd., 2023. [Online]. Available: https://www.manageengine.com/log-management/cyber-security-attacks/common-types-of-cyber-attacks.html. [Accessed: 7 2 2024].

[6] Harley Kozushko, “Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems,” vol. 11, 2003.

[7] Leyla Bilge and Tudor Dumitraş. 2012. Before we knew it: an empirical study of zero-day attacks in the real world. In Proceedings of the 2012 ACM conference on Computer and communications security (CCS '12). Association for Computing Machinery,NewYork,NY,USA,833-844.DOI:https://doi.org/10.1145/2382196.2382284

[8] Salah, K., Kahtani, A.: Performance evaluation comparison of Snort NIDS under Linux and Windows Server. J. Netw. Comput. Appl. 33(1), 6–15 (2010). https://doi.org/10.1016/j.jnca.2009.07.005. (ISSN: 1084-8045)

[9] A. Gharib, I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "An Evaluation Framework for Intrusion Detection Dataset," 2016 International Conference on Information Science and Security (ICISS), Pattaya, Thailand, 2016, pp. 1-6, doi: 10.1109/ICISSEC.2016.7885840.

[10] Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerging Tel Tech.2021;32:e4150. https://doi.org/10.1002/ett.4150

[11] Dini P, Elhanashi A, Begni A, Saponara S, Zheng Q, Gasmi K. Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity. Applied Sciences. 2023; 13(13):7507. https://doi.org/10.3390/app13137507

[12] Kumar, G., Kumar, K. & Sachdeva, M. The use of artificial intelligence based techniques for intrusion detection: a review. Artif Intell Rev 34, 369–387 (2010). https://doi.org/10.1007/s10462-010-9179-5

[13] Georgescu T-M. Natural Language Processing Model for Automatic Analysis of Cybersecurity-Related Documents. Symmetry. 2020; 12(3):354. https://doi.org/10.3390/sym12030354

[14] Benyamin Ghojogh, Maria N. Samad, Sayema Asif Mashhadi, Tania Kapoor, Wahab Ali, Fakhri Karray, & Mark Crowley. (2019). Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review. https://doi.org/10.48550/arXiv.1905.02845

[15] “About - Simple Transformers” Thilina Rajapakse, 2020. [Online]. Available: https://simpletransformers.ai/about/. [Accessed: 17 2 2024].