Communications of the CCISA
Vol. 29 No. 4 Nov. 2023

基於 Transformer 的入侵監測系統

蔡孟典[1]、曾國鈞[2*]
國立宜蘭大學資訊工程學系
[1]n1143009@niu.edu.tw、[2]kctseng@niu.edu.tw

摘要

數位時代的崛起深刻改變了我們社會的運作方式，公共和私營部門紛紛迎來了這場變革。隨著數位平台的普及，實現了無紙化的操作，這在日常工作中帶來了極大的便利。然而，隨之而來的是新的挑戰，數位科技的快速發展催生了各種威脅，包括駭客攻擊、惡意軟體、網路釣魚和 DDoS 攻擊等。在這樣的背景下，我們迎來了一個全新的挑戰——保障網路安全。惡意活動的不斷演進使得市場上需要更智慧、更靈活的方式來應對。為此，本研究提出了一種基於 Transformer 的 BERT multi-classification 預訓練模型的監測分類模型。這個模型不僅能夠實現即時線上分析網路流量，還能夠識別潛在的異常或可疑行為。主動預防和緩解惡意攻擊是我們的目標，而這需要我們透過網路分析深入了解攻擊的意圖。本研究基於 CIC-IDS-2017 資料集，該資料集包含各種攻擊和正常流量，為訓練模型提供了寶貴的訓練數據。本研究所的解決方案為建立在流的入侵偵測系統基礎之上，利用 Transformers 的機器深度學習技術建立一個能夠透過流量分析即時識別威脅的入侵偵測系統，為數位時代的安全保障提供一層堅實的防線。

關鍵詞：網路安全、流量分析、入侵檢測系統、深度學習、Transformer、BERT、CICIDS2017

* 通訊作者 (Corresponding author.)

Communications of the CCISA
Vol. 29 No. 4 Nov. 2023

Intrusion Detection System Based on Transformer

Meng-Tien Tsai [1], Kuo-Chun Tseng[2*]
Department of Computer Science and Information Engineering, National Ilan University
[1]n1143009@niu.edu.tw, [2]kctseng@niu.edu.tw

Abstract

The digital era has profoundly transformed the daily operations of both public and private sectors. They have embraced digital platforms across multiple devices, presenting diverse content to meet the challenges of everyday work. The shift to digital platforms has enabled paperless operations but has also brought new challenges. The rapid development of digital technology has given rise to malicious activities such as hacking, malware, phishing, and DDoS attacks. These threats aim to gather intelligence, steal sensitive data, or demand ransom, leading to a surge in cybersecurity incidents.

In this paper, we propose a monitoring classification model using a pre-trained Transformer-based BERT multi-classification model. This model can perform real-time online analysis of network traffic and identify potential anomalies or suspicious behavior. Proactive prevention and mitigation of malicious attacks involve understanding attack intent through network analysis. This study trains machine learning models on the CIC-IDS-2017 dataset, which includes various attacks and normal traffic. The proposed solution is based on a flow-based intrusion detection system that leverages machine deep learning with Transformers. The goal is to establish a real-time intrusion detection system capable of identifying threats through traffic analysis, providing a robust defense for the cybersecurity challenges of the digital era.

Keywords: Cybersecurity, flow analysis, Intrusion Detection System, Deep Learning, Transformer, BERT, and CICIDS2017

壹、前言

1.1 研究背景

在公部門和私部門的運作中，數位化時代持續並徹底地改變了我們的日常操作模式。這種數位化包括了各種不同樣式的內容呈現和多種載具設備，促使許多資料轉向了數位平台，實現了無紙化運作。然而，近年來隨著數位技術的快速發展，有心人士利用駭客、惡意軟件以及網路釣魚、DDoS 等手段，對系統進行攻擊，以搜集情報和嘗試竊取敏感資料。這導致了資安事件的頻繁發生。引述 CSR 天下的報導 “網路資安廠商 Fortinet 旗下 FortiGuard Labs 威脅情資中心 16 日公布《2023 上半年全球資安威脅報告》，報告顯示，台灣 2023 上半年遭受的惡意威脅數量急遽成長，叫去年同期大增逾八成，每秒有將近 1.5 萬次攻擊發生，高居亞太之冠“[1]。大量的攻擊行為已經無法由人簡單的判斷，即便將人的知識轉成規則，從大量日誌紀錄中去分析和辨識，也很難與時俱進，因為攻擊發展的速度太快了，甚至攻擊方式也演化為人工智能生成方式，以致於太過複雜，運用機器學習來建立偵測模型是個必然的方向。而透過大數據的學習，模型可以很快的從大量訓練資料中，學到新興的威脅並反應。另外，針對新的攻擊手法的防範措施，常常需要專家花費大量時間，透過知識轉換為規則的方式來應對。這樣的過程不僅耗時，還容易導致出現太冗長的空窗期，使得系統在這段時間內處於相對較弱的防護狀態。透過機器學習模型能夠自主地學習和識別，已更新防衛能力，不僅提高了系統對新型攻擊的應變能力，還減少了依賴人工維護的複雜性和延遲的空窗期，使得整體資安防禦更加強大而敏捷。

為了因應這些挑戰，入侵偵測系統（Intrusion Detection System, IDS）成為了一個關鍵的安全防禦工具，其主要透過持續性的網路流量監控，並進一步分析和識別是否存在潛在的異常或可疑行為。為了提前預防，避免受到惡意攻擊，我們透過網路偵測和分析來預先了解攻擊的意圖。市場上有不少偵測入侵的軟體，但要找到既能有效偵測又價格合理的選項確實有一定難度。一些商業軟體雖然提供較為強大的功能，但其價格昂貴，對於很多私部門，尤其是中小企業，其預算有限和 IT 資源不足的現實條件下，要安裝能夠偵測最新的攻擊手法，似乎不太可行。在此背景下，希望能致力發展一套能簡單部署於中小企業且有效的入侵偵測系統。

再者，深度學習（Deep Learning）在近代有了顯著的發展，帶來了許多突破和創新。首先在圖像辨識，透過卷積神經網絡（CNN）的模型，成功推動了圖像辨識的發展。圖型辨識模型如 ResNet（Residual Networks）的發展，能夠拓展更多層的模型架構。另外，也有像 YOLO 物體檢測（Object Detection）的深度學習模型，能夠在一張圖像中同時檢測多個物體，並為每個物體提供對應的邊界框（Bounding Box）和類別標籤。而自然語言處理（NLP）的革新也是這幾年快速發展的領域。模型如 BERT（Bidirectional Encoder Representations from Transformers）和 GPT（Generative Pre-

trained Transformer）在機器翻譯、情感分析、文本生成等任務上表現卓越。計算密集型任務產生了重大影響。這些重大的深度學習模型，都提供人工智慧在近年來蓬勃的發展。

此外，隨著圖形處理單元（GPU）的發展，計算機的計算能力取得了巨大的提升，這對於深度學習和其他計算密集型任務產生了重大影響。GPU 的主要特點是其強大的並行處理能力。對於深度學習中的大規模矩陣運算等並行計算來說，GPU 能夠充分發揮優勢。這也加速了深度學習訓練的時間。 因為 NLP 相關的深度學習模型，如 BERT 或 GPT 等等預訓練模型，訓練過程通常需要大量的運算。GPU 的並行處理能力使得訓練過程能夠更快速地完成，大大縮短了訓練時間。再加上近年來，大規模資料被提供做訓練，讓深度學習更容易在一般的實驗環境實現。

而深度學習所建立的模型，在網路攻擊偵測方面帶來了多方面的優勢。此技術賦予了系統自動適應學習的能力，無需明確的先前規則。讓系統能夠靈活應對新型攻擊，並在不斷變化的威脅環境中調整。透過大量數據中學習攻擊和正常行為的模式，使得偵測更具彈性，能夠應對多種不同形式的攻擊。如果有更新的攻擊手法，相較於 rule-based 系統，利用深度學習的彈性更為高效。這種可以偵測一些看似無序，無規則性的攻擊行為，最適合透過深度學習所建立模型來偵測。

1.2 研究目的

隨著網路病毒和攻擊手法的不斷進化，保障公部門和私部門的資訊安全變得越來越有挑戰性。傳統上，尋找攻擊的方法是檢查每個封包的內容。然而，封包檢查在高速環境下難以輕易實現。因此本研究探索替代方法，例如基於流的入侵偵測。在這種方法中，分析的重點是數據在網路中的流動，而不是單個封包的內容。而過往已經有不少研究關於機器學習來做訓練和偵測，模型的選擇上，過往也有不同模型的採用，比方說 deep convolutional neural network (CNN) ensemble framework[2] 卷積神經網絡集成的模型，也有 SVM 支援向量機(Support Vector Machine)，隨機森林(Random Forest)，和 K 近鄰(K-Nearest Neighbors)的偵測方法[3]。

再者，即時大量的資料，也讓偵測和告警變得越來越困難，故透過流（flow）的研究，探索不同攻擊的 flow 型態，讓機器學習得以辨識。本研究將在 CIC-IDS-2017 數據集上進行機器學習訓練，以讓模型可以偵測攻擊。該數據集中包含的各種攻擊和正常流量提供了大量數據做訓練和檢測。固本研究基於這個數據集的訓練，而提出了基於流的入侵偵測系統。本研究專注於流(flow-based)的偵測，透過 Transformer 的機器深度學習的方法，致力於建立一套能夠即時依據流而做的識別入侵的檢測系統。

貳、文獻探討

S. KishorWagh 等人於 2013 年發表的論文「Survey on Intrusion Detection System using Machine Learning Techniques」中[4]，針對統計異常、知識和機器學習三種入侵偵測系統 (IDS) 技術做介紹，當中有提出這三種技術的優缺點。統計異常為基礎入侵偵測系統 是一種基於統計學的入侵偵測技術，它通過檢測數據中的異常來識別攻擊。知識為基礎的 IDS 是一種基於知識的入侵偵測技術，它使用關於已知攻擊的知識來識別攻擊。建立簽名和規則來作識別。機器學習方法所做的入侵偵測系統，是一種基於機器學習的入侵偵測技術，透過不同的模型建立來識別攻擊。該方法簡要出機器學習的技術，能夠準確並檢測出新的入侵攻擊。但需要大量人工標記資料做訓練。而 CIC-IDS2017 的資料集廣泛在標記資料中，運用於機器學習的訓練，所以我們從過往有做過在 CID-IDS2017 上做過的機器學習方面的研究來探索，了解目前在 IDS 入侵偵測相關研究的方向，並進一步的探索透過機器學習的可行性。也透過 NLP 領域上重要的模型 Transformer 在多類別分類上的任務，來研究運用在即時的 IDS 偵測的可行性和研究。以下會先提到過去做過的機器學習，然後在探討本研究相關的技術和使用 Transformer 預訓練 BERT 模型作微調的相關文獻。在訓練中，若遇到資料不平衡的問題，可採用文獻中 K. Md. Hasib 等人[9]提到的多類別資料分類所遇到資料不平衡的處理方式，來改善模型對少樣本的資料分類。

2.1 Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset[3]

2021 年，Z. K. Maseer 等人提出了一項研究[3]，針對 CICIDS2017 資料集使用不同的機器學習方法進行相關調查。該研究的目的是探討如何應用機器學習技術來提升對 CICIDS2017 資料集中的異常行為的檢測。研究中可能包含對不同機器學習算法的應用，例如人工神經網絡（ANN）、支持向量機（SVM）、決策樹（DT）、隨機森林（RF）和朴素貝葉斯（NB），以及非監督學習算法，如 k 均值(K-means)、自組織映射（SOM）和期望最大化（EM）算法。研究還探討了這些方法在檢測網絡攻擊方面的效能，以及對性能進行評估的指標和方法。這項研究的重要性在評估不同機器學習方法在處理 CICIDS2017 資料集時的效能，以及對新型攻擊的檢測能力。在 ANN 的表現方面，Z. K. Maseer 等人使用的 testing dataset 主要測試在四種標籤上，分別為 Benign，Bruce Force, XSS, SQL Injection。Z. K. Maseer 等人的實驗結果顯示，透過 ANN 的方式，可以得到平均 99.31%的準確率。但可惜的是這份的測試集只有四個分類，也沒有實際模擬攻擊情況，故本研究希望將測試集的 14 種攻擊分類都做測試，並且提升準確率(Accuracy)和精確率(Precision)。與此同時，能夠結合自然語言處理(Natural Language Processing)的技術 Transformer 來做訓練和部署。此技術在 14 中攻擊分類中，都有卓越

的表現，而總體的準確率達到 99.9%。再者，透過 Transformer 的方式，在資料前處理更為簡單，只需要把數字資料轉換為文字即可。

2.2 Attention is All You Need[5]

Transformer 模型是在 2017 年由 A. Vaswani 等人在題為「Attention is All You Need」[5]的論文中提出的。這個模型基於了編碼器-解碼器結構，但增加了一個重要的自注意力層，這使得模型能夠更好地處理複雜的語言任務。而 Transformer 模型主要是想要改善 RNN 在語言訓練方面的任務而創建的，RNN 在訓練過程中，會產生梯度消失或梯度爆炸的問題。另外，也產生出平行運算方法，已改善序列運算的效率。

Transformer 模型採用編碼器-解碼器結構。 編碼器和解碼器均由 6 層堆疊組成(如圖一)。 每個編碼器-解碼器結構的開頭是一個嵌入層，它映射了序列中單字的位置。每個編碼器-解碼器結構還有一個多頭自注意力層（如圖二），它重現序列中的所有單字之間的關係。 然後輸出通過前饋網路(Feed forward network)，進一步處理以獲得最終輸出。 解碼器層與編碼器層類似，解碼器用於輸出預測結果。共有三層：多頭注意力（Multi-head attention）、遮蔽多頭注意力和全連接前饋。該模型能提供並行處理資料的能力，並且能夠減輕梯度消失/爆炸的問題。而多頭注意力讓每一個頭擁有自己的權重矩陣，這也讓每個頭能夠學習不同的表示，有助於模型更全面捕捉輸入序列的訊息，也提升輸入序列中的不同部分表達能力。

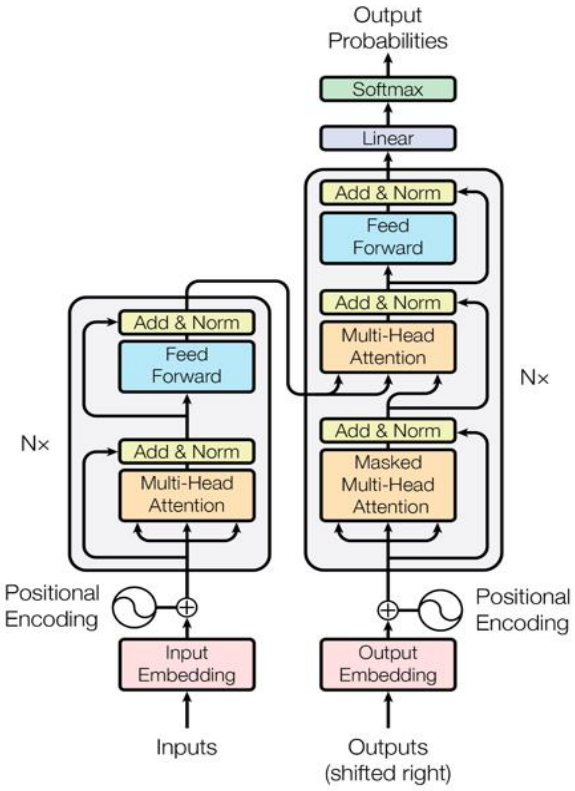
2.3 BERT 用於文本 multi-class 多類別的分類任務

Bert 的多類別文本分類可應用於各個領域，比方說情感分析相關的產品評論、股票預測、電影評論、新聞文章或政治辯論等。情感分析旨在從文本中提取和理解作者的情感和態度，透過 BERT 的文本分類，分析文本中的情感態度在哪個分類中[6]。在商品評價方面，企業可以利用情感分析來了解客戶對其產品或服務的滿意度，從而針對性地進行改進。在股票預測領域，情感分析能夠掌握市場情緒，有助於預測股價的走勢。在政治辯論中，情感分析則能夠追蹤公眾對政治話題的情感變化，有助於政治策略的制定。另外還有假新聞的辨識，有害新聞的分類有害程度[7]，這些都可以透過 BERT 的文本多類別的分類模型做微調 fine-tune，而成為特定任務的分類器。

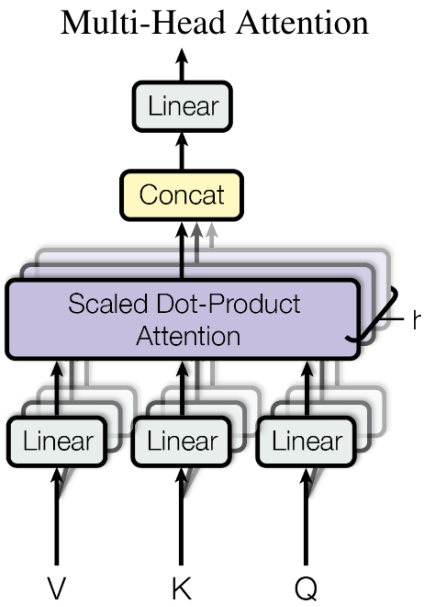
BERT-base 模型包含一個具有 12 個 Transformer 塊、12 個自注意力頭和隱藏大小為 768 的編碼器。BERT 接受一個不超過 512 個標記的序列輸入，並輸出該序列的表示。序列中包含一個或兩個段落，其中序列的第一個標記始終是[CLS]，包含特殊的分類嵌入，另一個特殊標記[SEP]用於分隔段落[8]。 在文本分類任務中，BERT 將序列的整體表示視為第一個標記[CLS]的最終隱藏狀態 h。為了預測標籤 c 的概率(1)：

$$p(c|h) = \text{softmax}(W\ h) \quad (1)$$

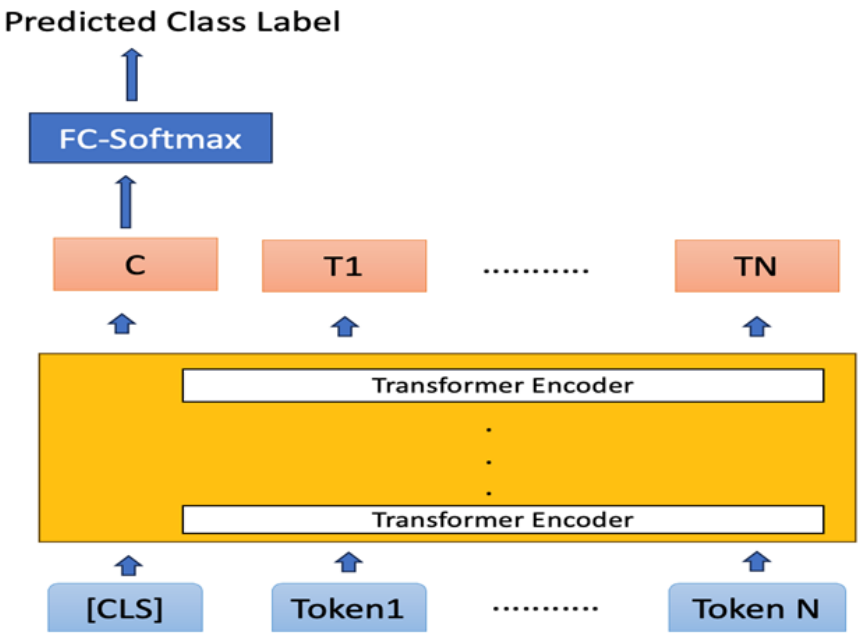
BERT 的頂部添加了一個簡單的 softmax 分類器(如圖三)。其中 W 是特定任務的參數矩陣。我們通過最大化正確標籤的對數概率，聯合微調 BERT 的所有參數以及 W。



圖一：Transformer 模型



圖二：Multi-Head Attention



圖三：BERT-base Multi-class classification

2.4 A Survey of Methods of Managing the Classification and Solution of Data Imbalance Problem[9]

在這篇文獻K. Md. Hasib 等人[9]提到的多類別資料分類所遇到資料不平衡的問題，在資料預處理技術中，採樣的技術被應用於資料集，可以透過新增或刪除樣本來進行，將新樣本加入現有樣本的過程稱為過採(oversample)，而刪除樣本的過程稱為欠採樣(undersample) 。在分類問題中，演算法通常更加重權重於分類為多數類的樣本。在許多資料不平衡的訓練情況下，會將罕見的分類類別，視為多數類的標籤，這可能會導致更嚴重的問題。文獻中提到，在機器學習過程中，資料集不平等分布的類別可能會影響效能。這表示在學習過程中，類別對於少數類提供了極小的特定性，而在多數類中提供了極高的準確性。在數據集中的類別不平衡可能會嚴重扭曲在多數-少數分類問題中分類器的性能。

綜合以上，在機器學習領域中，應用於入侵偵測系統（IDS）技術的相關研究已經展現出比基於知識或統計的偵測方式更為優越的表現。特別是透過深度學習，已經能夠達到超過 99.5%的準確率。但缺點是未來要實際偵測的時候，產生的 flow 都需要做正規化，不是非常直覺的做法。更遑論其他的機器學習比方說決策樹(DT)，需要做特徵的挑選和尺度的縮放等等前置作業，這對非機器學習領域的資訊人員來說，如果要再次訓練新的攻擊類型以產生模型，就需要相關的資料前處理知識。有賴於今日 GPU 的快速方展和強大演算例，透過自然語言的領域來做複雜的多類別分類，已變得可行。只要把這些特徵結合起來當做一句機器說的語言，透過 Transformer 的 BERT 可以深度學習後做多類別的分類，而準確率可以達到 99.9%。我們也發現，這個訓練方法，可以將數量少的類別也能正確的辨別出來。

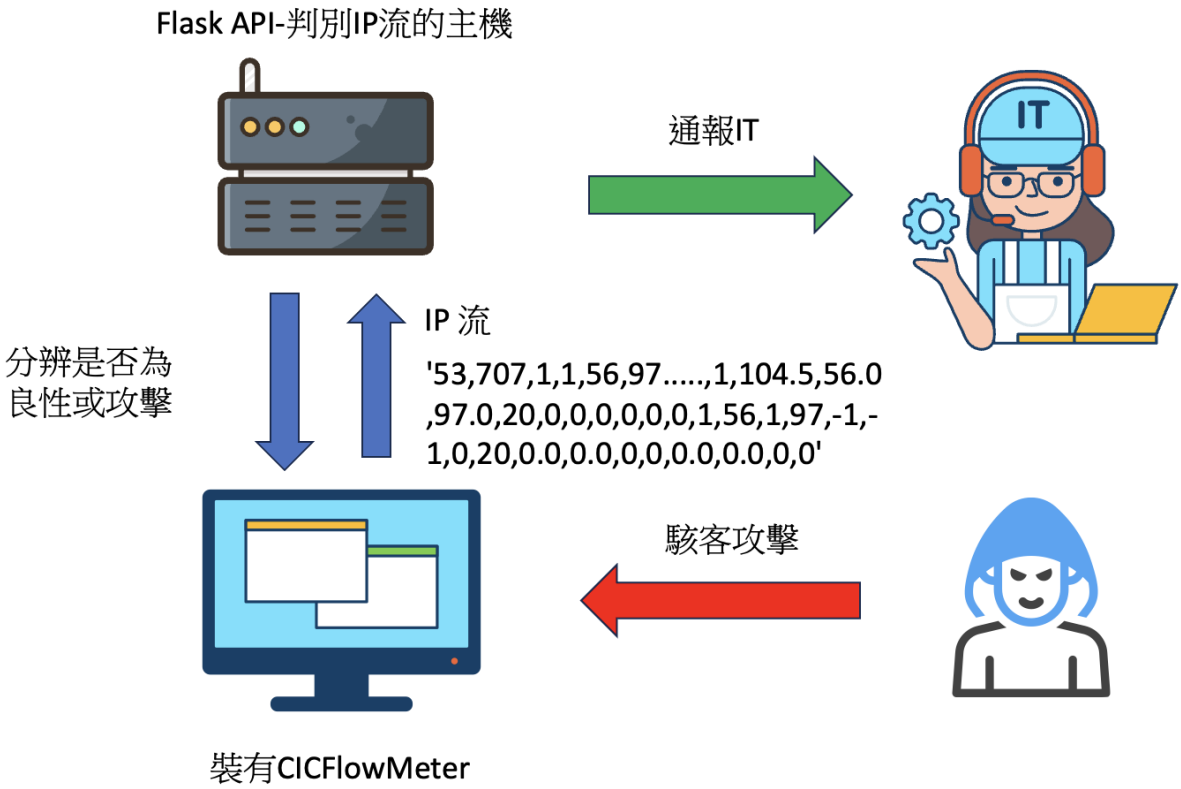
參、方法

透過語言模型做分類的目前有 BERT，RoBERTa，GPT，ALBERT 等較為熟知的預訓練模型，都是基於 Transformer 模型或是輕量化 BERT 而生成的預訓練語言模型。而本次利用 Transformer 的 BERT 語言模型來訓練，將資料型態轉換為文字型態，就能做分類訓練。將資料集裡面的數字，看似無意義的數字轉換為文字做訓練，並且建構多類別分類模型。承如上面 2.1 點提到 Z. K. Maseer 等人 [3]，在 CID-IDS2017 資料集做過的不同種類的機器學習模型訓練，而當時並沒有使用到 Transformer 的相關機器學習做訓練。故本研究將使用此資料集作模型訓練。一方面證實語言模型運用在即時偵測系統的可行性，一方面也提升準確率。在訓練好的模型完成後，會嵌入一個 Flask API 來回應送來的流是否為惡意攻擊和攻擊的分類。在實際模擬攻擊環境中，我們透過 CICFlowMeter 安裝在被攻擊主機上產生流，也即時將產生的流送到 Flask API 來進行判

斷，再將判斷反饋給網管人員。以下就整個研究方法的系統架構，微調使用的 NLP 套件 Simple Transformers，和產生的工具 CICFlowMeter 做介紹。而模型訓練將使用 BERT 預訓練來微調，是希望透過機器學習中，能學習到 flow 的文本表示。將無法人工判斷的 flow，轉為文字型態，讓模型來理解並進行分類。同時，也讓訓練出來的分類器，能在 CIC-IDS2017 的 15 種分類的表現上，都能夠有不錯的準確率。

3.1 系統架構圖

實驗架構如圖四，在靶機（攻擊目標）上裝載 CICFlowMeter 以產生流。另在實驗環境中安裝一臺 Kali 當做攻擊者，進行相關的攻擊。CICFlowMeter 將封包訊息轉成 flow，透過 post 方式，傳送到遠端的 Flask API 做判別。當偵測到 flow 有可能是攻擊行為，會判定類別並通知。



圖四：系統架構圖

3.2 訓練模型之工具 Simple Transformers

訓練工具上，我們使用 Simple Transformers 的 BERT 來訓練 multi-classification 的多類別分類任務[10]。Simple Transformers 是一個自然語言處理 (NLP) 函式庫，它簡化 Transformer 模型的使用。Simple Transformers 分為常見的 NLP 任務，例如文字分類、

問答和語言建模。 每個任務都有自己特定於任務的 Simple Transformers 模型，所有特定於任務的模型都保持一致的使用模式（初始化、訓練、評估、預測）[10]。使用的預訓練為 bert-base-uncased 這個預訓練模型。 「uncased」意味著模型不區分大小寫，這意味著它將「Hello」和「hello」視為同一個單字。

模型的訓練係採用 CICIDS2017 數據集，此數據集包含良性和最新的常見攻擊，類似於真實世界數據 （PCAP）。透過 Simple Transformer 的 multi-class classification, 將模型訓練為能夠分辨 15 種類型的模型，其中包含 14 種攻擊行為，和一種良性行為。

3.3 CICFlowMeter 4.0

此元件會在實際模擬攻擊的環境中，安裝在靶機，根據此元件產生流(flow)來做分析。我們使用 CICFlowMeter 4.0 來產生流(flow)[11]。CICFlowMeter 是一款開源的網路流量生成器和分析器。它可以產生雙向流量，其中第一個封包確定了正向（源到目的地）和反向（目的地到源）的方向。並且分別計算正向和反向方向上的 80 多個網路流量統計特徵，如持續時間、封包數量、位元組數量、封包長度等。 其他功能包括從現有功能列表中選擇功能、新增新功能以及控制流程超時的時間。輸出是以 CSV 格式的檔案，其中每個流量（流量 ID、來源 IP、目的地 IP、來源埠、目的地埠和協定）被標記為六列，包括 80 多個網路流量分析特徵。TCP 流通常在連接斷開時終止（透過 FIN 封包），而 UDP 流則由流程超時終止。流程超時值可以根據具體情況任意指定，例如 TCP 和 UDP 的超時值可以設定為 600 秒。

肆、實驗過程暨結果

實驗的過程，首先我們會先透過資料集做訓練。然後用測試集測試模型的準確率和精準度。確認訓練好的模型在測試集上有不錯的表現後，我們將模型裝在 Flask API 上，以即時接收來自靶機的流（flow）。模擬攻擊時，我們在攻擊機上嘗試不同的攻擊方法，並看模型可否即時測出惡意的攻擊行為。這也驗證是否此模型可以在實際的網路環境中，擔任偵測的任務。

4.1 模型訓練使用的資料集

本模型的訓練，將採用在 CIC-IDS2017[12]資料集進行模型訓練。CIC-IDS2017 資料集包含良性且最新的常見攻擊，類似於真實世界收集到的資料（PCAP）。 此資料集採用透過使用 CICFlowMeter 進行網路流量分析的結果，基於時間戳記、來源和目標 IP、來源和目標連接埠、協定和攻擊，並根據這些網路分析資料做標記。 對於這個資

料集，基於 HTTP、HTTPS、FTP、SSH 和電子郵件協定建構了 25 個使用者的抽象行為。資料收集於 2017 年 7 月 3 日星期一上午 9 點開始，於 2017 年 7 月 7 日下午 5 點結束，共 5 天。 星期一是正常的一整天，僅包括良性使用行為。 實施的攻擊包括暴力 FTP、暴力 SSH、DoS、Heartbleed、Web 攻擊、滲透、殭屍網路和 DDoS，這些都在其餘四天執行攻擊。

4.2 資料不平衡的前處理

在這個資料集中，除了標籤為 "BENIGN" 的封包是正常的以外，其他所有封包都涉及攻擊行為。我們可以看到 "BENIGN" 類別的樣本佔了整個資料集的 80.3004%，而部分攻擊類別的樣本數量非常少，比如 "Heartbleed" 和 "Infiltration"，原始資料類別分佈如表一所示。因此，我們需要先處理資料不平衡的問題。作法是將標籤屬於 BENIGN 的樣本進行隨機的 Undersampling 欠採樣，採樣的數量為 2 萬筆。而得到新的 BENIGN 的資料數量在跟其他攻擊的資料結合。另外，Infiltration，Web Attack Sql Injection，Heartbleed 這三種類別資料過於稀少，透過 Oversampling 過採樣，將資料增加 50 倍，總資料數目為調整之後的資料比重如表二所示。

表一：原始 CIC-IDS2017 資料集之資料分布

標籤	資料數量	佔總資料百本比%
BENIGN	2,273,097	80.3004%
DoS Hulk	231,073	8.1630%
PortScan	158,930	5.6144%
DDoS	128,027	4.5227%
DoS GoldenEye	10,293	0.3636%
FTP-Patator	7,938	0.2804%
SSH-Patator	5,897	0.2083%
DoS slowloris	5,796	0.2048%
DoS Slowhttptest	5,499	0.1943%
Bot	1,966	0.0695%
Web Attack Brute Force	1,507	0.0532%
Web Attack XSS	652	0.0230%
Infiltration	36	0.0013%
Web Attack Sql Injection	21	0.0007%
Heartbleed	11	0.0004%
Total	2,830,743	100%

表二：調整後的資料集分佈情形

標籤	資料數量	佔總資料百本比%
BENIGN	200,000	26.2796%
DoS Hulk	231,073	30.3626%
PortScan	158,930	20.8831%
DDoS	128,027	16.8225%
DoS GoldenEye	10,293	1.3525%
FTP-Patator	7,938	1.0430%
SSH-Patator	5,897	0.7749%
DoS slowloris	5,796	0.7616%
DoS Slowhttptest	5,499	0.7226%
Bot	1,966	0.2583%
Web Attack Brute Force	1,507	0.1980%
Web Attack XSS	652	0.0857%
Infiltration	1,836	0.2412%
Web Attack Sql Injection	1,071	0.1407%
Heartbleed	561	0.0737%
Total	761,046	100%

4.3 資料轉換前處理

資料的前處理 (Data preprocessing)的第二步驟接下來，將特徵轉變文字型態，將 78 種數字特徵，轉為文字並結合再一起。把 78 個數字特徵當作一段文字（如圖五）。

然後將資料集最後一欄位為標籤，獨立為學習的目標值 y，並進行編碼。總共 15 個分類項將其編碼為 0-14 。最後，我們透過隨機分層抽樣，把資料集分成 80%的訓練集，20%的測試集。

‘53,707,1,1,56,97,56,56,56.0,0.0,97,97,97.0,0.0,216407.355,2828.854314,707.0,0.0,707,707,0,0.0,0.0,0,0,0,0,0.0,0.0,0,0,0,0,0,0,0,0,20,20,1414.427157,1414.427157,56,97,69.66666667,23.67136104,560.3333333,0,0,0,0,0,0,0,0,1,104.5,56.0,97.0,20,0,0,0,0,0,0,1,56,1,97,-1,-1,0,20,0.0,0.0,0,0,0,0.0,0.0,0,0’

圖五：將數值資料轉換為文字

4.4 設備暨環境

因為大量的資料需經過 Transformer 的模型運算，這需要強大的運算系統來協助處理，以下表三是訓練暨預測的運算環境。

表三：設備暨環境

Equipment	Description
CPU	13th Gen Intel(R) Core(TM) i9-13900K 3.00 GHz
GPU	NVIDIA GeForce RTX 4080
Operation System	Window 11 22H2
Environment	Jupyter-lab 4.0.7-1, transformers==4.24.0, simpletransformers==0.63.11, PyTorch 2.0.1

4.5 測試集的實驗數據

針對 transformer 訓練後的模型進行實驗結果評估時，我們使用混淆矩陣 (Confusion Matrix) 如表四， 以及兩種評估深度學習模型效能的指標：準確度 (Accuracy) 和精確度 (Precision)。混淆矩陣如表四所示，其中 True Positive (TP) 代表實際為該類別且預測為該類別，即真陽性；False Positive (FP) 代表實際非該類別但預測為該類別，即偽陽性；False Negative (FN) 代表實際為該類別但預測非該類別，即偽陰性；True Negative (TN) 代表實際非該類別且預測為非該類別，即真陰性。

表四：混淆矩陣(Confusion Matrix)

	Actual Class (正確的分類)		
		Positive	Negative
Predicated Class (預測的分類)	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

表四中呈現了本研究方法在測試資透過這四項指標，我們可以計算模型的準確度 (Accuracy) 和精確度 (Precision)，其計算公式如以下所示的(2)和(3)。

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

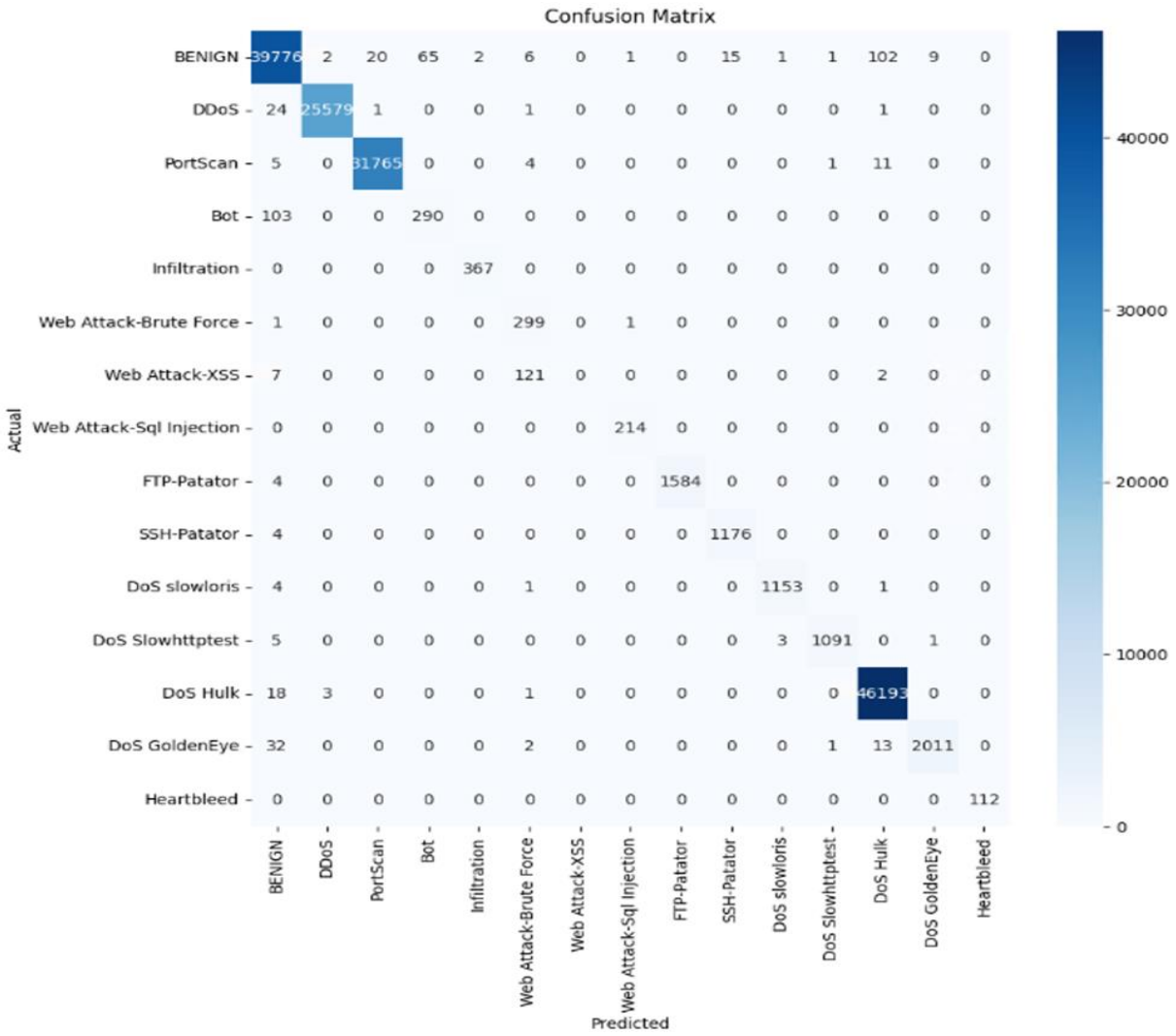
$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

測試集總共包含了 151,529 筆資料，並列出了各種攻擊類型的真陽性（TP）、假陽性（FP）、真陰性（TN）和假陰性（FN）的數據筆數。根據實驗數據，我們可以看出本文提出的 Transformer 模型能夠實現 99.95%的準確率以及 99.6%的精確率（如表五）。另外，上述提到樣本數稀少的一些攻擊類型，例如 Infiltration, Heartbleed 和 WebAttack Sql Injection 等，其樣本數量佔總資料量太少，這在初期不使用 oversample 的情況下，會導致了精確度為零的狀況。再透過 oversample 方式來補充這些樣本數較少的資料類型。經過訓練和測試後，這些稀少樣本都提高了準確率。此外，本文提出的 Transformer 模型在應對分散式阻斷服務攻擊（DDoS）和阻斷服務攻擊（DoS）方面表現出很高的預測準確率和精確率。

表五：測試集之測試結果

Label	**TP**	**FP**	**TN**	**FN**	**Accuracy**	**Precision**
BENIGN	399,776	207	112,003	224	0.9992	0.999
DDoS	25,579	5	126,599	27	0.9998	1.000
PortScan	31,765	21	120,403	21	0.9997	0.999
Bot	290	65	151,752	103	0.9989	0.817
Infiltration	367	2	151,841	0	1.0000	0.995
Web Attack Brute Force	299	136	151,773	2	0.9991	0.687
Web Attack XSS	0	0	152,080	130	0.9991	0.000
Web Attack Sql Injection	214	2	151,994	0	1.0000	0.991
FTP-Patator	1,584	0	150,622	4	1.0000	1.000
SSH-Patator	1,176	15	151,015	4	0.9999	0.987
DoS slowloris	1,153	4	151,047	6	0.9999	0.997
DoS Slowhttptest	1,091	3	151,107	9	0.9999	0.997
DoS Hulk	46,193	130	105,865	22	0.9990	0.997
DoS GoldenEye	2,011	10	150,141	48	0.9996	0.995
Heartbleed	112	0	152,098	0	1.0000	1.000
Total	**511,610**	**600**	**2,130,340**	**600**	**0.9995**	**0.999**

圖六進一步顯示在測試資料集中，各種分類被誤判後，被歸類為哪一類。這可以供未來工作上，如果要改進罕見攻擊類型在資料集內的表現，作為進一步分析和資料擴增的依據。



圖六：測試集各類型攻擊分類

4.6 模擬攻擊的前處理

本研究要將訓練好的模型，透過模擬攻擊來了解模型是否能即時檢測攻擊行為。模擬攻擊前，先在被攻擊的機器上安裝 CICFlowMeter 4.0[13]，此工具在偵測的時候，會產生並輸出被攻擊主機的 flow 資料，並轉換為 csv 檔案。 為了更同步的將 flow 資料送到偵測主機，我們在 CICFlowMeter 的 src 檔案夾內，修改 flow_session.py 檔案，增加 post request 的模組。此舉可以讓每次被攻擊主機所產生的 flow，都能立即傳送到偵測主機做判讀。同時，在傳送 flow 之前，我們也需要前處理該 flow 資料，將 src IP，dest IP，Timestamp，des_mac，src_mac，src_port, protocol 從 flow 中移除，並且把剩下的資料結合，轉換為字串。 但在每次偵測之後，由偵測主機回傳判別的種類後，我們需要將 src IP 加回該筆資料。 這樣方便我們判別是怎樣的攻擊，從哪個 src IP 而來。

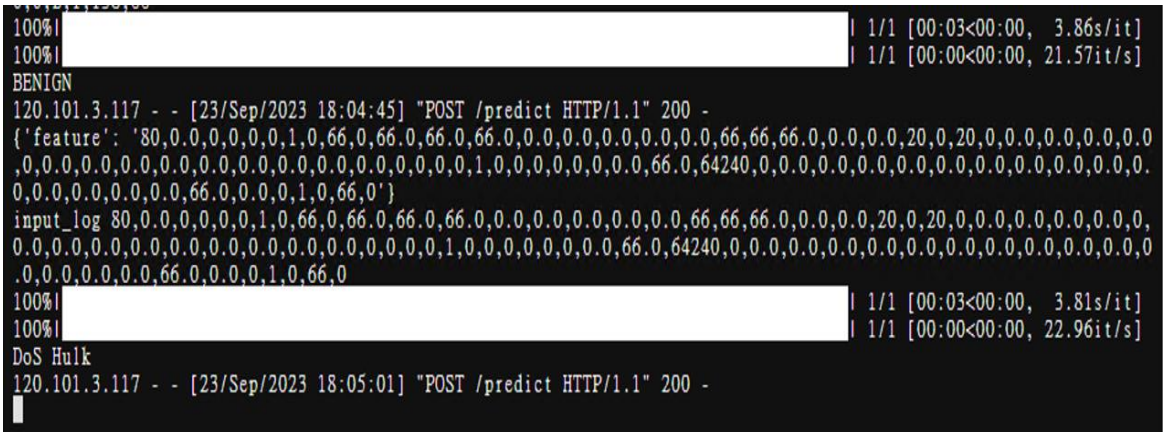
4.7 模擬攻擊預測

在前面實驗章節所做的模型訓練，我們想不僅是透過測試集做準確率的偵測，還進一步想要將此模型掛載在 Flask API 上，並實際透過模擬攻擊，來查看是否真的能即時的偵測攻擊。故以下我們透過 Slowloris，slowhttptest，nmap 這些工具，來做模擬攻擊，並查看偵測模型是否能成功偵測。

4.7.1 Slowloris 攻擊

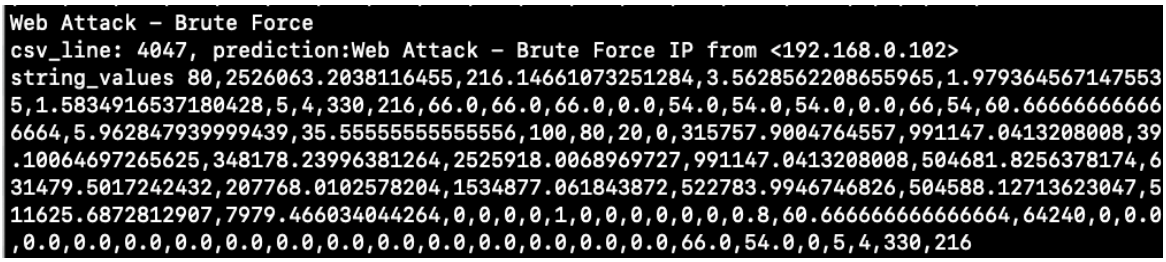
在 Kali（模擬攻擊者）的機器上，我們執行 Slowloris 的攻擊。Slowloris 是一種簡單而有效的網路攻擊工具，用於對目標網站進行拒絕服務（DoS）。Slowloris 攻擊利用了 HTTP 協議的一個特性，即在客戶端發送 HTTP 請求後，伺服器會保持與客戶端的連接打開一段時間，直到該請求完成或超時。Slowloris 利用這一點，通過在連接上發送大量的部分請求但不完整的請求，使得伺服器保持這些連接打開，最終消耗伺服器的資源，導致其無法正常處理其他合法的請求。Slowloris 攻擊的特點是它不需要大量的資源或大量的傳輸流量，但可以有效地使目標網站變得不可用。這種攻擊方式特別適用於那些無法負擔大量流量的網站，因為它主要通過保持連接來達到攻擊的目的。下圖是由 Kali 執行 Slowloris 來攻擊 mac 的時候，在裝載 transformer model 上的偵測主機，偵測到的即時畫面，顯示此次的攻擊為 Dos Hulk 如下圖七和圖八。

>python slowloris.py 192.168.0.100 -s 5000



圖七：使用 slowloris 攻擊之模型即時判斷

另外執行兩次攻擊指令“slowloris -c 1000”，此時偵測模型判別為 WebAttack-Brute Force 如圖八：



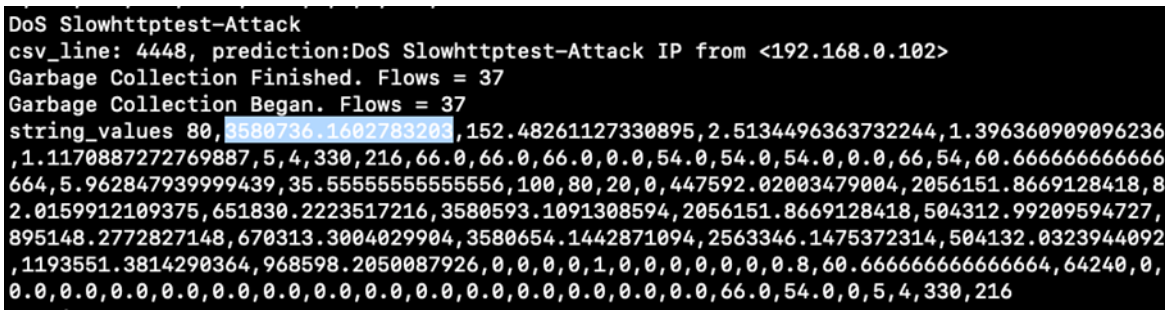
圖八：使用 slowloris 攻擊之模型即時判斷

4.7.2 slowhttptest 攻擊

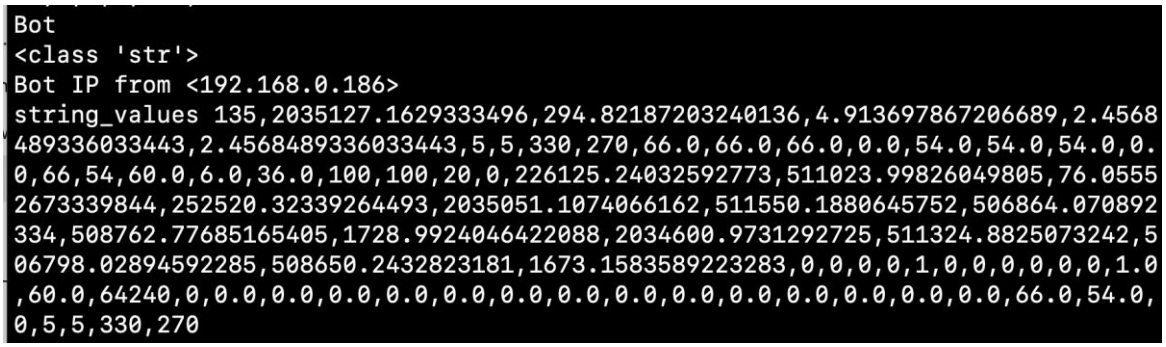
另外，在攻擊機 Kali 上，也做了 slowhttptest 的攻擊行為。slowhttptest [14]是一種用於模擬慢速 HTTP 攻擊的工具。慢速 HTTP 攻擊是一種利用 HTTP 協議中的一些漏洞或設計缺陷，通過以非常緩慢的速度傳輸HTTP請求來消耗目標服務器的資源的攻擊方式。slowhttptest 可以通過以不同的方式傳輸 HTTP 請求，如緩慢的請求體、緩慢的標頭、緩慢的請求行等，來模擬這種攻擊，以便測試目標服務器的抗擊能力。以下是在 Kali 攻擊機上，執行的攻擊指令:

>sudo slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 200 -t GET -u http://192.168.0.100 -x 24 -p 3

在偵測主機上，偵測到的訊息 DoS Slowhttpest-Attack 和 Bot 的判別如圖九和圖十：



圖九：使用 slowhttptest 攻擊之即時判斷



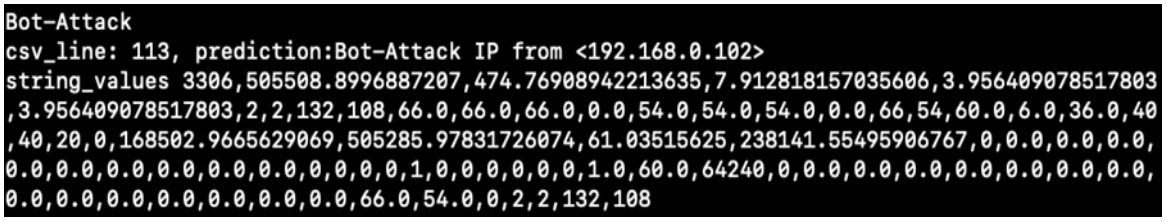
圖十：使用 slowhttptest 攻擊之即時判斷

4.7.3 Nmap 端口掃描

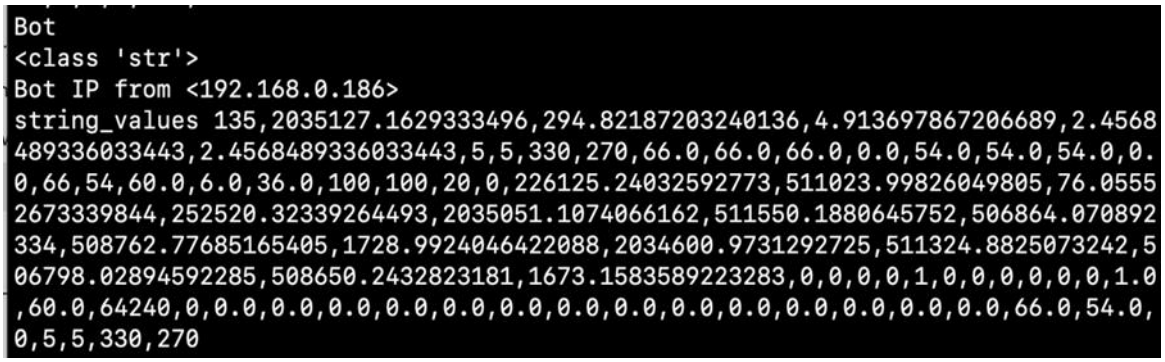
在攻擊機 Kali 上面，以超級用戶權限（sudo）使用 nmap 工具。對 IP 地址為 192.168.0.202 的主機進行廣泛的端口掃描，以了解哪些端口是開放的，可以接受連接。指令如下：

>sudo nmap -p- 192.168.0.202

這執行過程中，偵測主機回應為 Bot 的攻擊，如下面圖十一和圖十二所示：

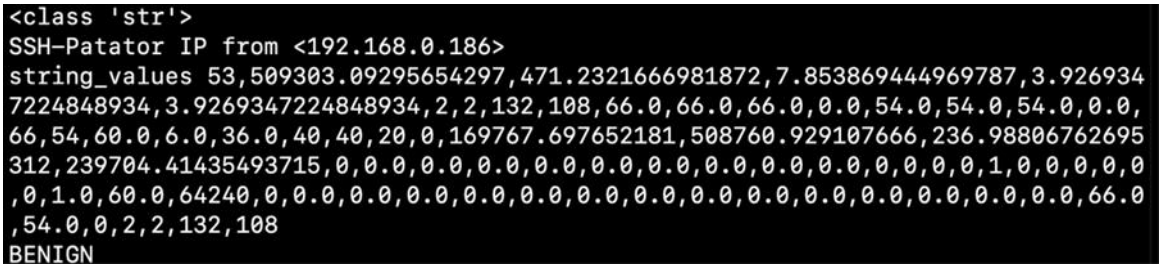


圖十一：使用 Nmap 掃描之即時判斷



圖十二：使用 Nmap 掃描之即時判斷

此外，當我們做 sudo nmap -p- 的掃描時候，偵測主機也會反映出 SSH-Patator 和 Bot 的攻擊行為如圖十三。當使用 nmap 掃描一個主機的所有端口時，會將這種掃描行為或模式，解釋為可能的攻擊行為。在這種情況下，回覆這樣的 flow 為 SSH-Patator 的攻擊。SSH-Patator 是一種針對 SSH 服務的暴力破解攻擊，嘗試通過反覆嘗試不同的用戶名和密碼來破解 SSH 登錄。當 nmap 掃描所有端口時，可能會檢測到這種攻擊行為。



圖十三 使用 Nmap 掃描之即時判斷

伍、結論

綜合以上所述，本論文提出了一種基於 Transformer 神經網路的分類模型，用於偵測和判別網路攻擊。研究方法顯示，本文所提出的模型在預測準確率上可達到 99.9%。模型的訓練過程大約耗時2小時，因此未來若有新的資料集或自行模擬攻擊的資料集，都可以加入該模型進行訓練。在本研究的結論中，我們強調了該模型在實際網路安全應用中的可行性和高效性。此外，透過 CICFlowMeter 的即時流量判別，有效地提高了網路攻擊偵測的即時性，為網路安全防護提供了有別於封包檢測的偵測方案。 再者，因模型能夠短時間內訓練，展示了其在應對未來新資料集或模擬攻擊資料集時的可拓展性。 未來的工作上，近一步能夠透過集成的方式，結合其他預測分類模型，增加未知攻擊的判斷性，並且能在偵測攻擊的報告中，提供相對應的防範措施，更能提高企業網路安全。

參考文獻

[1] “上半年台灣平均每秒遭駭客攻擊近1.5萬次 企業面臨產線中斷、匿名勒贖風險,” CSR@天下. Accessed: Nov. 20, 2023. [Online]. Available: https://csr.cw.com.tw/article/43300

[2] S. Haider et al., “A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks,” IEEE Access, vol. 8, pp. 53972–53983, 2020, doi: 10.1109/ACCESS.2020.2976908.

[3] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, “Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset,” IEEE Access, vol. 9, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614.

[4] S. KishorWagh, V. K. Pachghare, and S. R. Kolhe, “Survey on Intrusion Detection System using Machine Learning Techniques,” IJCA, vol. 78, no. 16, pp. 30–37, Sep. 2013, doi: 10.5120/13608-1412.

[5] A. Vaswani et al., “Attention Is All You Need.” arXiv, Aug. 01, 2023. Accessed: Nov. 04, 2023. [Online]. Available: http://arxiv.org/abs/1706.03762

[6] S.-Y. Lin, Y.-C. Kung, and F.-Y. Leu, “Predictive intelligence in harmful news identification by BERT-based ensemble learning model with text sentiment analysis,” Information Processing & Management, vol. 59, no. 2, p. 102872, Mar. 2022, doi: 10.1016/j.ipm.2022.102872.

[7] E. Shushkevich, M. Alexandrov, and J. Cardiff, “Improving Multiclass Classification of Fake News Using BERT-Based Models and ChatGPT-Augmented Data,” Inventions, vol. 8, no. 5, p. 112, Sep. 2023, doi: 10.3390/inventions8050112.

[8] C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to Fine-Tune BERT for Text Classification?” arXiv, Feb. 05, 2020. Accessed: Nov. 27, 2023. [Online]. Available: http://arxiv.org/abs/1905.05583

[9] K. Md. Hasib et al., “A Survey of Methods for Managing the Classification and Solution of Data Imbalance Problem,” Journal of Computer Science, vol. 16, no. 11, pp. 1546–1557, Nov. 2020, doi: 10.3844/jcssp.2020.1546.1557.

[10] T. Rajapakse, “Multi-Class Classification,” Simple Transformers. Accessed: Nov. 06, 2023. [Online]. Available: https://simpletransformers.ai/docs/multi-class-classification/

[11] “Applications | Research | Canadian Institute for Cybersecurity | UNB.” Accessed: Nov. 06, 2023. [Online]. Available: https://www.unb.ca/cic/research/applications.html

[12] “IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” Accessed: Nov. 26, 2023. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[13] datthinh1801, “Python CICFlowMeter.” Nov. 08, 2023. Accessed: Nov. 10, 2023. [Online]. Available: https://github.com/datthinh1801/cicflowmeter

[14] “slowhttptest | Kali Linux Tools,” Kali Linux. Accessed: Nov. 06, 2023. [Online]. Available: https://www.kali.org/tools/slowhttptest/