國立政治大學

網路安全的資料科學期末報告

基於機器學習和深度學習的惡意網址辨識

Detection of Malicious Websites Using Machine Learning and Deep Learning Techniques

授課教授：蕭舜文 博士

學生：黃柏翔 撰

中　華　民　國　一　一　一　年　六　月

目錄

第一章　緒論

在網際網路如此發達的年代，網路攻擊儼然成為我們最需要防範的事項之一，如果不慎遭受攻擊，可能導致個人的如信用卡號碼、帳號密碼等機密資訊外流，其中一種攻擊手段即為社交工程(social engineering)，常見的手法偽裝成政府機構或是宅配業者等，在郵件或簡訊中放入惡意網址引誘他人點選，而這些惡意網址便會透過各種形式來欺騙使用者，以下列舉幾種常見的詐騙形式:

圖: 偽裝成銀行的簡訊詐騙 (來源:whoscall)

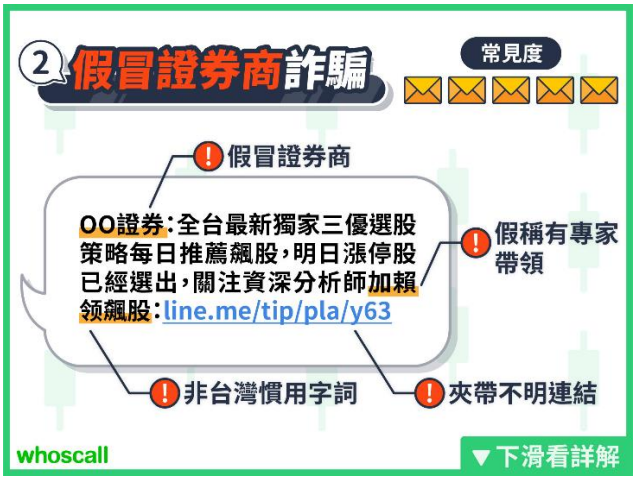

圖: 投資簡訊詐騙 (來源:whoscall)

圖：偽裝成政府的簡訊詐騙 (來源:whoscall)

圖：我實際收到的詐騙簡訊

有鑑於近期實在收到太多這種包含惡意網址的詐騙簡訊，本次的期末報告期望可以透過人工智慧的技術來辨別惡意網址，一方面可以實踐課程所學的技術，另一方面也可以使我更加了解資料科學於資訊安全領域的應用。

第二章　資料介紹

本文選用 Siddhartha, M. (2021)於 Kaggle 所刊登的資料集，此為作者蒐集不同惡意 URL 相關的資料集整合而成，該資料集並無缺失值，共包含兩個欄位: URL 和 label，其中 label 為標記該網站為良性(benign)、竄改(defacement)、釣魚(phishing)、惡意(malware)，特徵的建立上我參考施淳譯(2020)、Rasymas, T.& Dovydaitis, L.(2020)、Ozcan, A., Catal, C., Donmez, E. et al.(2021)產生了共創造了以下 26 個變數，變數的說明、分布圖、相關係數圖如下:

變數名稱	**說明**
and	The number of "&" in the URL
backslash	The number of "\" in the URL
comma	The number of "," in the URL
dash	The number of "-" in the URL
digit	The number of digits in the URL
dot	The number of "." in the URL
exclamation	The number of "!" in the URL
hashtag	The number of "#" in the URL
http	The number of "http" in the URL
https	The number of "https" in the URL
ip	Does an ip address in the URL?
known_tld	Does the url contain a known TLD ("com","org","net","edu","gov","int","mil","us","ca","cn","fr","au","de","jp","nl","uk","mx","no","ru","br","se","es","co","tw") ?
letter	The number of letters in the URL
mouse	The number of "@" in the URL
path_len	Total length of the path
percent	The number of "%" in the URL
plus	The number of "+" in the URL
question	The number of "?" in the URL
semicolon	The number of ";" in the URL
shorten	Does the URL contain a shortening service?

slash	The number of "/" in the URL
tilde	The number of "~" in the URL
tld_len	Total length of the TLD
underline	The number of "_" in the url
url_len	Total length of the url
www	The number of "www" in the URL

表: 變數名稱與說明

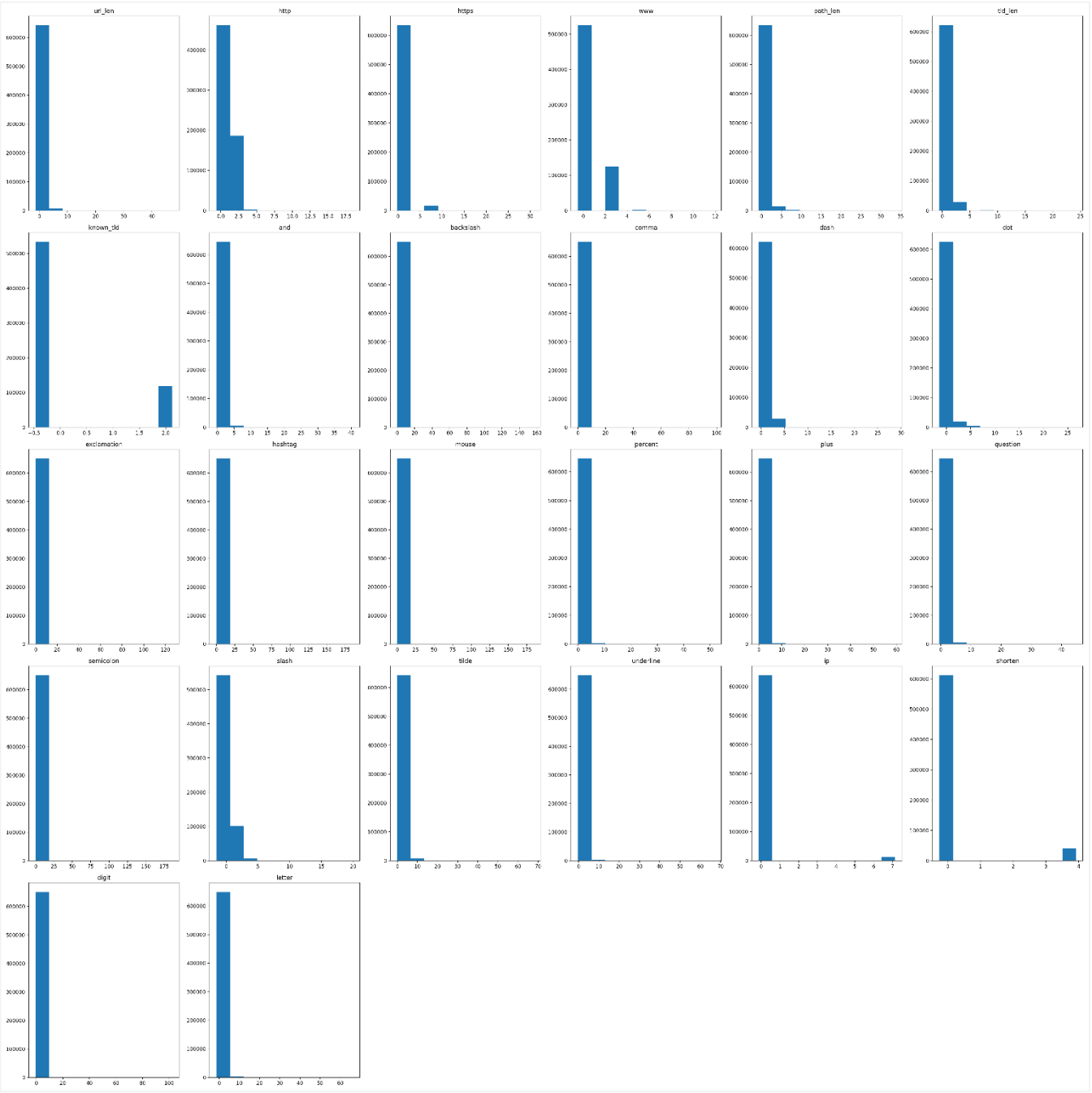

圖: 變數分配直方圖

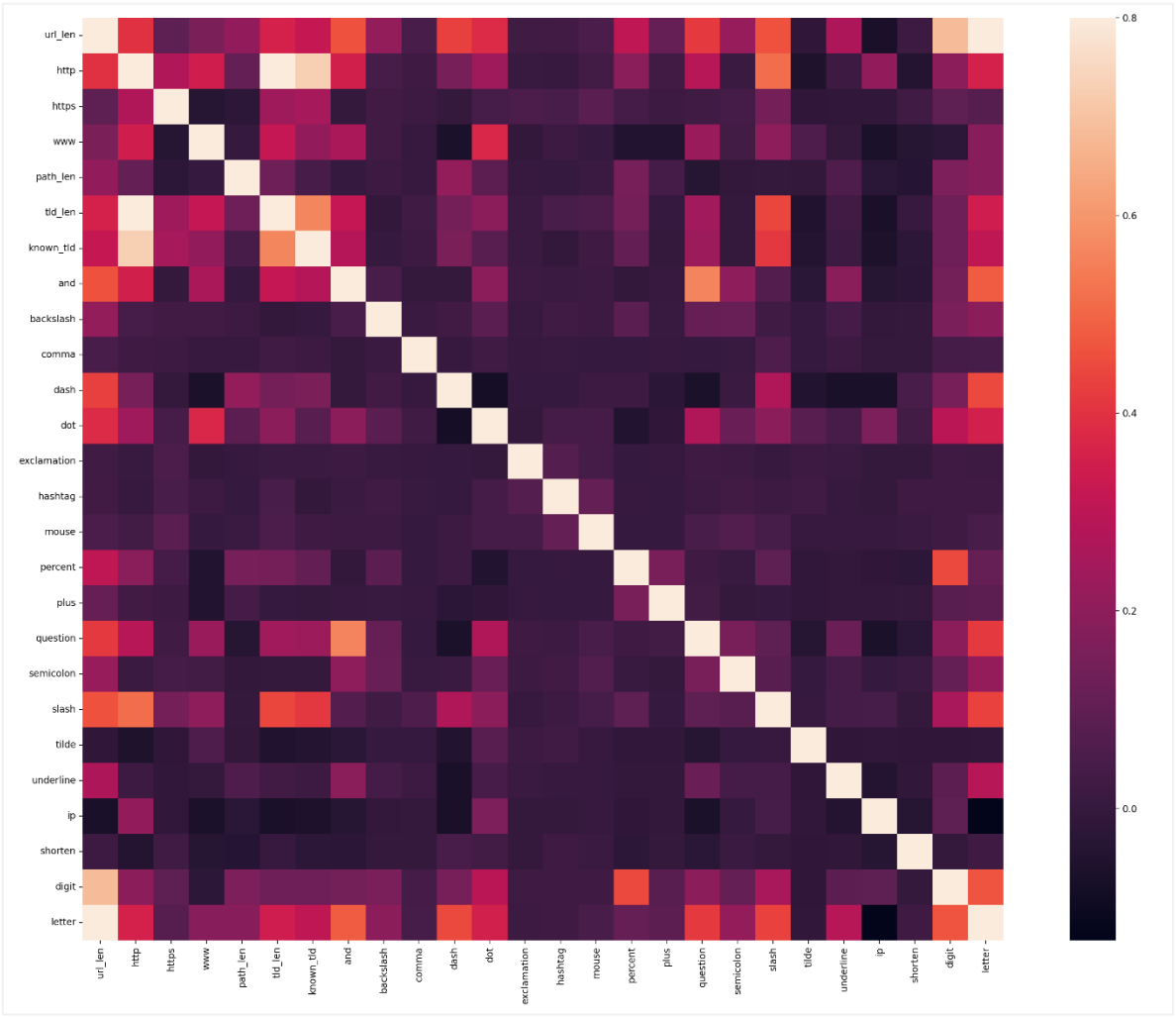

圖: 變數相關係數圖

在 label 的處理上我將其轉為數值型，其中 benign 的 URL 為 0，而 defacement、phishing、malware 的 URL 我全部視為惡意的 URL 並標記為 1:

標籤	說明
0	Benign or safe URL
1	Defacement, phishing, or malware URL

表: 標籤說明

資料集中共有 66%為良性的 URL、34%為惡意的 URL，下圖為原始資料集 label 分布的情形:

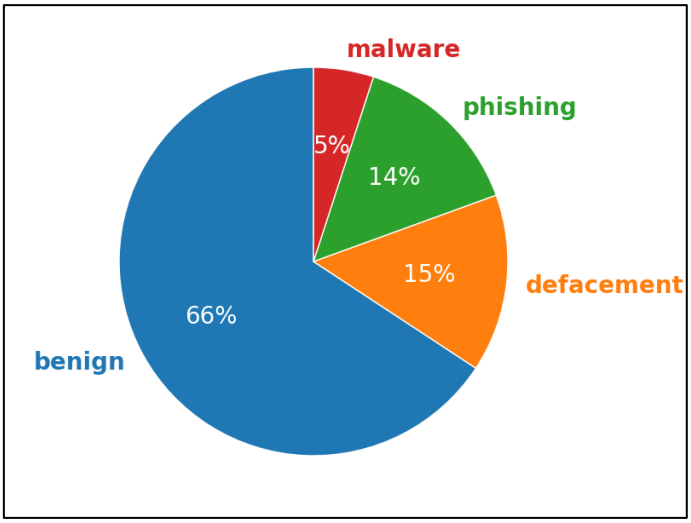

圖: label 分布

第三章　研究方法

本章研究方法共分為四小節，第一節敘述特徵工程，第二節介紹本次所採用的機器學習模型，第三節進一步引進深度學習模型，第四節說明評估模型績效所採用的指標。

第一節　特徵工程

在資料的特徵工程，我僅有將解釋變數進行標準化，原先有嘗試過 PCA，然而結果是需要 20 個主成分才可以解釋 90%的總變異，因此後來我便決定就使用全部的變數不做 PCA，而後我將 2/3 的資料作為訓練集、1/3 的資料做為測試集，在後續隨機種子的選用我統一設定為我的學號 110352028。

第二節　機器學習模型

由於資料集帶有標籤，適合採用監督式學習演算法，本次選取的模型如下:

模型	參數設定(如空白則表示使用預設之參數)
Naïve Bayes	
Logistic Regression	• random_state=110352028
Decision Tree	• criterion='entropy' • random_state=110352028 • max_depth=3

Random Forest	• n_estimators=100 • max_depth=3 • min_samples_split=10 • max_features=10 • n_jobs=-1 • warm_start=True • random_state=110352028
XGBoost	• n_estimators=100 • max_depth=3 • random_state=110352028

表: 機器學習模型參數選擇

由於資料筆數較多，使用我的筆記型電腦訓練模型可能需要比較長的時間，因此在參數的選用上，我選擇的標準是模型可以在 2 分鐘內訓練完成且訓練準確率達到 90%為基準。

第二節　深度學習模型

本次使用的資料是網址，而網址的文字彼此之間並沒有序列的關係，我認為如果使用 NLP 的模型可能沒有那麼合適，同樣地，我也認為 CNN、RNN 兩個模型較不適用於此次的資料集，因此本次選用的模型是一般的神經網路，進一步去探討說新增了 early stopping 和 dropout layer 是否對於模型的預測準確度有不同的影響。

模型	參數設定(如空白則表示使用預設之參數)
Neural Network 1	• Dense (100, activation='relu') + Dense (50, activation='relu') + Dense (1, activation='sigmoid') • model.compile(optimizer='adam', loss='BinaryCrossentropy') • model.fit(x = X_train, y = y_train, batch_size = 128, epochs=10, validation_split = 0.3)
Neural Network 2 - early stopping	同上，增加 early stopping • callbacks = [keras.callbacks.EarlyStopping(monitor="val_loss", min_delta=1e-2, patience=5, verbose=1)] • model.fit (x = X_train, y = y_train, batch_size = 128, epochs=10, callbacks = callbacks, validation_split = 0.3)
Neural Network 3 - early stopping - dropout layer	同上，神經網路架構增加 dropout layer • Dense (100, activation='relu') + Dropout (0.2, seed=110352028) + Dense (50, activation='relu') + Dropout (0.2, seed=110352028) + Dense (1, activation='sigmoid')

表: 深度學習模型參數選擇

第三節　績效評估指標

本次選擇的預測績效評估指標如下:

- Accuracy
- Precision
- Recall
- F1-Score

第四章　實證結果

本章敘述模型用於測試資料集的績效評估，以下為各模型用於測試資料集的混淆矩陣以及預測績效:

圖: 模型混淆矩陣

	Naïve Bayes	Logistic Regression	Decision Tree	Random Forest	XGBoost	Neural Network 1	Neural Network 2	Neural Network 3
Accuracy	86.30%	89.86%	91.24%	92.59%	96.46%	96.67%	96.59%	96.32%
Precision	85.03%	90.20%	91.22%	92.14%	96.50%	96.78%	96.61%	96.52%
Recall	84.22%	87.02%	89.11%	91.26%	95.61%	95.79%	95.78%	95.28%
F1-Score	84.60%	89.86%	90.02%	91.67%	96.03%	96.26%	96.17%	95.86%

表: 模型預測績效

整體而言，深度學習模型的表現皆優於機器學習模型。在機器學習模型中，XGBoost 表現最為優異，準確率已與深度學習模型十分靠近，且本次為了縮短訓練的時間，有對 max_depth 做限制，如後續再針對參數進行最佳化，機器學習模型可能可以有更好的表現。在深度學習模型中，原先我在 callbacks 的參數中設定 patience=3，這會造成訓練第 3 次就觸發了 early stopping，此時模型的預測準確度只有約 50%，因此我將最小訓練的次數設定為 5 次，NN2 和 NN3 於第 5 次就觸發了 ealry stopping，就結果而言，NN1 後續訓練的第 6 至 10 次對於準確率並沒有太多貢獻；增加了 dropout layer 的 NN3 預測準確率並沒有表現更好，或許日後選擇防止 over-fitting 的方法中可以在 early stopping 和增加 dropout layer 之間擇一即可。

第五章　結論與未來展望

本文嘗試利用網路安全的資料科學課程所學之技術，實際應用於惡意網站的資料集上，透過機器學習和深度學習的技術來辨識惡意網站，在特徵工程上由於沒有辦法使用較少的主成分解釋整體資料的變異，並沒有執行 PCA，另外考量網址彼此文字之間較沒有序列關係的特性，後續模型選擇沒有選用 NLP、CNN、RNN 等模型。

整體而言深度學習的模型皆優於機器學習的模型，然而為了訓練時間的縮減我有針對機器學習模型設定參數，若有更長的訓練時間機器學習模型可能可以有更好的表現。比較三個神經網路，可以發現三者的預測準確度並無太大的差異，我推論可能在防止 over-fitting 的方法中，可能在 early stopping 或是增加 dropout layer 擇一即可。

受限於資訊相關知識的不足，我認為我在特徵選用上有可以改進的地方，例如增加常見的 subdomain，抑或是針對 URL 的特性做更細膩的處理。此外在模型建立的部分，與朋友討論後他有提出可以實際連上網站抓取網頁上的訊息，如此便可以使用 NLP 模型來建立特徵，後續要繼續進行相關的研究，我認為可以從上述這兩點來著手。

參考文獻

1. 施淳譯（2020）。基於類神經網路之釣魚網站辨識系統，國立中興大學資訊管理學系所碩士論文，台灣台中。
2. Ozcan, A., Catal, C., Donmez, E., & Senturk, B. (2021). A hybrid DNN–LSTM model for detecting phishing URLs. *Neural Computing & Applications*, 1–17.
3. Rasymas, T. & Dovydaitis, L. (2020). Detection of phishing URLs by using deep learning approach and multiple features combinations. *Baltic Journal of Modern Computing*, 8(3), 471–483.
4. Siddhartha, M. (2021) Malicious URLs dataset. Retrieved from https://www.kaggle.com/datasets/sid321axn/malicious-URLs-dataset on May 25, 2022.