國立清華大學

智慧化企業整合
Intelligent Integration of Enterprise

Project 3

Web page Phishing Detection

網絡釣魚網頁檢測系統之模型建立

指導教授： 邱銘傳 教授

學生：110034402 黃彥蓉

目錄

一、 背景介紹

（一）背景與動機

在這個互聯網發達的世代，我們越來越依賴互聯網來進行及處理我們大部分日常的事務，這為網絡犯罪分子提供了發起有針對性的網絡釣魚攻擊的完美環境。因此網絡釣魚成為了網絡犯罪分子欺騙網絡使用者最成功和最有效的方式之一，他們籍著釣魚網站竊取我們的個人及財務資訊。

現今發生的網絡釣魚攻擊非常複雜，而且越來越令人難以發現。有研究發現，97% 的安全專家無法從真實電子郵件中識別出網絡釣魚電子郵件。

（二）研究目的

由於網絡釣魚攻擊非常複雜，一般人難以發現，因此本研究透過建立模型辨識釣魚網頁，期望能提供所有網絡使用者受到網絡釣魚攻擊疑惑時可自行進行檢測的系統，以降低網絡犯罪分子利用網絡釣魚手法竊取我們的個人和財務信息之機會。

（三）問題描述

表 1　5W1H

What	網絡釣魚攻擊非常複雜一般人難以發現，有機會被網絡犯罪分子竊取我們的個人及財務資訊。
Why	透過深度學習的模型，建立網絡釣魚網頁檢測系統的模型，提供所有網絡使用者自行進行檢測，降低受到網絡釣魚攻擊的機會。
Where	任何地方
When	網絡使用者對於網頁有網絡釣魚攻擊疑惑時
Who	所有網絡使用者
How	MLP/ SVC/ LR/ NB

（四）資料集介紹

本研究使用 Kaggle 網站中網路釣魚網站辨識的公開資料集。資料集中包含 11430 筆 URL (網頁位址)。資料集中包含 87 個提取的特徵。特徵來自三個不同的類別：56 個從 URL 的結構和語法中提取；24 個從其對應頁面的內容中提取；7 個通過查詢外部服務提取。

數據集中包含 50% 的網絡釣魚和 50%的合法網頁的 URL。

二、 研究方法

(一) 方法選擇

本研究使用神經網路 MLP 建立模型，套用網頁位址資料集進行網路釣魚網頁檢測辨識，再調整參數達至 MLP 模型最佳精確度。

最後再以 SVC、LR 及 NB 三種機器學習方法對比四種不同模型的結果進行最終比較與選擇。

(1) 多層感知器（MLP）

MLP 是深度神經網路(DNN)的一種 special case。MLP 是一種前向傳遞類神經網路，至少包含三層結構(層感知輸入層、隱藏層和輸出層)，並且利用到「倒傳遞」的技術達到 model learning 的監督式學習。

MLP 神經網路利用 gradient descent 找最佳參數解，最後帶入 MLP 內的前向傳遞即可得到最後的預測值。

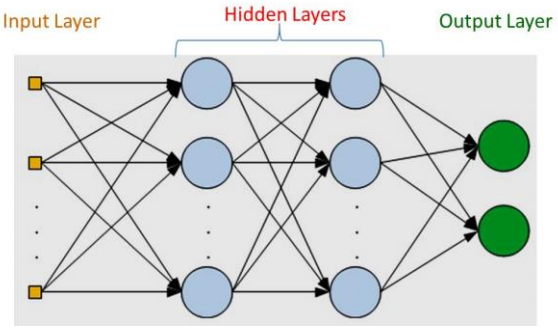

圖 1 MLP 架構圖

(2) Support Vector Classification（SVC）

SVM 是一種監督式的學習方法，用統計風險最小化的原則來估計一個分類的超平面。基礎概念是找到一個決策邊界(decision boundary)讓兩類之間的邊界(margins)最大化，使其可以完美區隔開來。

(3) Logistic Regression（邏輯迴歸）

Logistic Regression 是迴歸分析的一種，與一般線性迴歸不同，Logistic Regression 的依變項(Y)是類別變數，類別只有兩個為二元的邏輯式迴歸(Binary logistic regression) ，類別超過三個以上為 Polytomous logistic regression。通常使用「最大概似函數估計法」對參數做估計。

(4) Naive Bayes（樸素貝葉斯）

樸素貝葉斯演算法是應用最為廣泛的分類演算法之一。它是基於貝葉斯定義和特徵條件獨立假設的分類器方法。簡單貝氏模型直接假設所有的隨機變數之間具有條件獨立的情況，因此可以直接利用條件機率相乘的方法，計算出聯合機率分布。

三、 個案研究

(一) 資料介紹

本研究使用 Kaggle 網站中網路釣魚網站辨識的公開資料集。資料集中包含 11430 筆 URL (網頁位址)。資料集中包含 87 個提取的特徵。特徵來自三個不同的類別：56 個從 URL 的結構和語法中提取；24 個從其對應頁面的內容中提取；7 個通過查詢外部服務提取。

其中原數據集包含 50%的網絡釣魚 URL 和 50% 的合法網頁的 URL。

(二) 資料前處理

Step 1: 資料狀態轉碼

首先把每筆資料中的網頁地址(url)及其狀態(status)提出，明確分出各網址分別為合法網頁或釣魚網頁，並把其狀態 encode 為 1,0，合法網頁為 1；釣魚網頁為 0。

```
df_data['target'] = pd.get_dummies(df_data['status'])['legitimate'].astype('int')
df_data.drop('status',axis = 1, inplace=True)
df_data[['url','target']].head(5)
```

	url	target
0	http://www.crestonwood.com/router.php	1
1	http://shadetreetechnology.com/V4/validation/a...	0
2	https://support-appleId.com.secureupdate.duila...	0
3	http://rgipt.ac.in	1
4	http://www.iracing.com/tracks/gateway-motorspo...	1

Step 2: 資料檢查

檢查資料集中是否有 Missing Value

```
tmp = df_data.isnull().sum().reset_index(name='missing_val')
tmp[tmp['missing_val']!= 0]
```

index	missing_val

Step 3: 提取特徵

使用 urlparse 方法將其分解為有用的部分，從 URL 中提取有用的特徵

```
def parse_url(url: str) -> Optional[Dict[str, str]]:
    try:
        no_scheme = not url.startswith('https://') and not url.startswith('http://')
        if no_scheme:
            parsed_url = urlparse(f'http://{url}')
            return {
                'scheme': None, # not established a value for this
                'netloc': parsed_url.netloc,
                'path': parsed_url.path,
                'params': parsed_url.params,
                'query': parsed_url.query,
                'fragment': parsed_url.fragment,
            }
        else:
            parsed_url = urlparse(url)
            return {
                'scheme': parsed_url.scheme,
                'netloc': parsed_url.netloc,
                'path': parsed_url.path,
                'params': parsed_url.params,
                'query': parsed_url.query,
                'fragment': parsed_url.fragment,
            }
    except:
        return None
```

	url	status	scheme	netloc	path	params	query	fragment
0	http://00324il.moonfruit.com	phishing	http	00324il.moonfruit.com				
1	http://02dfbe20-6956-4be6-ba22-36ae0e0d3053.ht...	phishing	http	02dfbe20-6956-4be6-ba22-36ae0e0d3053.htmlcompo...	/get_draft		id=99ea0e_8eddaa8b0b3b6316a653bbd03eb5b48f.html	
2	http://03418f6.netsolhost.com/FF7AADF203DF6C7A...	phishing	http	03418f6.netsolhost.com	/FF7AADF203DF6C7A0B7C8A74B8164E55/			
3	http://03418f6.netsolhost.com/FF7AADF203DF6C7A...	phishing	http	03418f6.netsolhost.com	/FF7AADF203DF6C7A0B7C8A74B8164E55/		sec=Milka%20Gostovic	
4	http://03418f6.netsolhost.com/FF7AADF203DF6C7A...	phishing	http	03418f6.netsolhost.com	/FF7AADF203DF6C7A0B7C8A74B8164E55/		sec=Puc%20Kotsis	
...	...	...	...	...	...	...	...	...
11424	https://zmail221.appspot.com	phishing	https	zmail221.appspot.com				
11425	https://zonasegura1.bn.com.multiservicioswebth...	phishing	https	zonasegura1.bn.com.multiservicioswebther.com	/BNIWeb/Inicio/logins.do			
11426	https://zoomic.io/wp-includes/neworder/bizmail...	phishing	https	zoomic.io	/wp-includes/neworder/bizmail.php		email=&_rand=13vqcr8bp0gud&lc=1033&amp...	
11427	https://zoryanvk.wordpress.com/	legitimate	https	zoryanvk.wordpress.com	/			
11428	https://zrq2y.weblium.site/	phishing	https	zrq2y.weblium.site	/			

Step 4: URL 中提取資訊&標籤

在 URL 中提取有機會辨識釣魚網頁的有用的資訊，如 url 長度、tld (.com)、是否 ip 位置、標點符號等資訊，並加上標籤並把 URL 列刪除。

```
df_grp_y = df_grp['status'] #It was df_grp_
df_grp.drop('status', axis=1, inplace=True) #
df_grp.drop('url', axis=1, inplace=True)
df_grp.drop('scheme', axis=1, inplace=True)
df_grp.drop('netloc', axis=1, inplace=True)
df_grp.drop('path', axis=1, inplace=True)
df_grp.drop('params', axis=1, inplace=True)
df_grp.drop('query', axis=1, inplace=True)
df_grp.drop('fragment', axis=1, inplace=True)
df_grp
```

	length	tld	is_ip	domain_hyphens	domain_underscores	path_hyphens	path_underscores	slashes	full_stops	num_subdomains	domain_tokens	path_tokens
0	28	com	False	0	0	0	0	0	0	1	il moonfruit	
1	126	com	False	4	0	0	1	1	10	1	dfbe be ba ae e d htmlcomponentservice	get draft
2	63	com	False	0	0	0	0	2	34	1	f netsolhost	FF AADF DF C A B C A B E
3	84	com	False	0	0	0	0	2	34	1	f netsolhost	FF AADF DF C A B C A B E
4	80	com	False	0	0	0	0	2	34	1	f netsolhost	FF AADF DF C A B C A B E
...	...	...	...	...	...	...	...	...	...	...	...	...
11424	28	com	False	0	0	0	0	0	0	1	zmail appspot	
11425	76	com	False	0	0	0	0	3	24	3	zonasegura bn com multiservicioswebther	BNIWeb Inicio logins do
11426	145	io	False	0	0	1	0	3	33	0	zoomic	wp includes neworder bizmail php
11427	31	com	False	0	0	0	0	1	1	1	zoryanvk wordpress	
11428	27	site	False	0	0	0	0	1	1	1	zrq y weblium	

Step 5: Label 轉碼

把分類特徵(TLD 及 IP 位置)使用 OneHot 編碼轉換為 1,0，使 TLD 及 IP 位置在特徵重要性部分變得明顯。

```
categorical_features = ['tld', 'is_ip']
categorical_transformer = Pipeline(steps=[
        ('onehot', OneHotEncoder(handle_unknown='ignore'))])
```

Step 6: 資料切分資料分割 (訓練與驗證集)

將資料以 8：2 的比例切分成訓練集及測試集

```
X_train, X_test, y_train, y_test = train_test_split(df_grp, df_grp_y, test_size=0.2)
```

(三) 模型建立與訓練

建立 MLP 模型，架構為 Input Layer ,，兩個隱藏層，及 Output Layer。各層 Layer 節點數，Input 為 87；隱藏層一為 300；隱藏層二為 100；Output 為 1。Loss function 使用 Binary Cross Entropy。各 Layer 間應用 ReLU 函數，而 Output Layer 則應用 Sigmoid 函數以作最後的二元分類。其中加入 Dropout 層，以減少訓練的時間及避免過擬合，最後使用 Adam Optimizer 自動調整學習率。

```
ChurnModel(
  (layer_1): Linear(in_features=87, out_features=300, bias=True)
  (layer_2): Linear(in_features=300, out_features=100, bias=True)
  (layer_out): Linear(in_features=100, out_features=1, bias=True)
  (relu): ReLU()
  (sigmoid): Sigmoid()
  (dropout): Dropout(p=0.1, inplace=False)
  (batchnorm1): BatchNorm1d(300, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (batchnorm2): BatchNorm1d(100, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
```

圖 2 MLP 模型架構圖

表 1 MLP 參數設定

	Hyperparameter
NN MLP	(87,300,100,1)
Loss computation function	Binary Cross Entropy
Optimizer	Adam
Dropout	0.1
Activation function	ReLu, Sigmoid

(四) 參數優化

本研究利用實驗設計中的田口方法，我們選擇了四項主要參數：Dropout、Learning Rate、Epochs 及 Batch Size，並使用四因子三水準方法進行實驗設計，以 L9 直交表進行九次實驗來幫助參數優化，有效減少調整參數的總次數並獲得接近相同的結果。

表 2 實驗因子及水準

	水準 1	水準 2	水準 3
Dropout	0.1	0.3	0.6
Learning Rate	0.01	0.005	0.001
Epochs	50	70	100
Batch Size	16	32	64

表 3　L9 實驗設計參數組合

實驗	Dropout	Learning Rate	Epochs	Batch Size
1	0.1	0.01	50	16
2	0.1	0.005	70	32
3	0.1	0.001	100	64
4	0.3	0.01	70	64
5	0.3	0.005	100	16
6	0.3	0.001	50	32
7	0.6	0.01	100	32
8	0.6	0.005	50	64
9	0.6	0.001	70	16

(五) 實驗設計結果

表 4 實驗設計結果

實驗	Dropout	Learning Rate	Epochs	Batch Size	Test Accuracy
1	0.1	0.01	50	16	0.957
2	0.1	0.005	70	32	0.953
3	0.1	0.001	100	64	0.969
4	**0.3**	**0.01**	**70**	**64**	**0.975**
5	0.3	0.005	100	16	0.965
6	0.3	0.001	50	32	0.970
7	0.6	0.01	100	32	0.960
8	0.6	0.005	50	64	0.971
9	0.6	0.001	70	16	0.971

表 4 為 MLP 模型的實驗組合及結果，準確度最高的為實驗 4，準確度達到了 0.975，將這 9 次實驗結果透過 Minitab 統計分析後可以得到各個因子對於準確率的影響程度。結果如下圖 3：

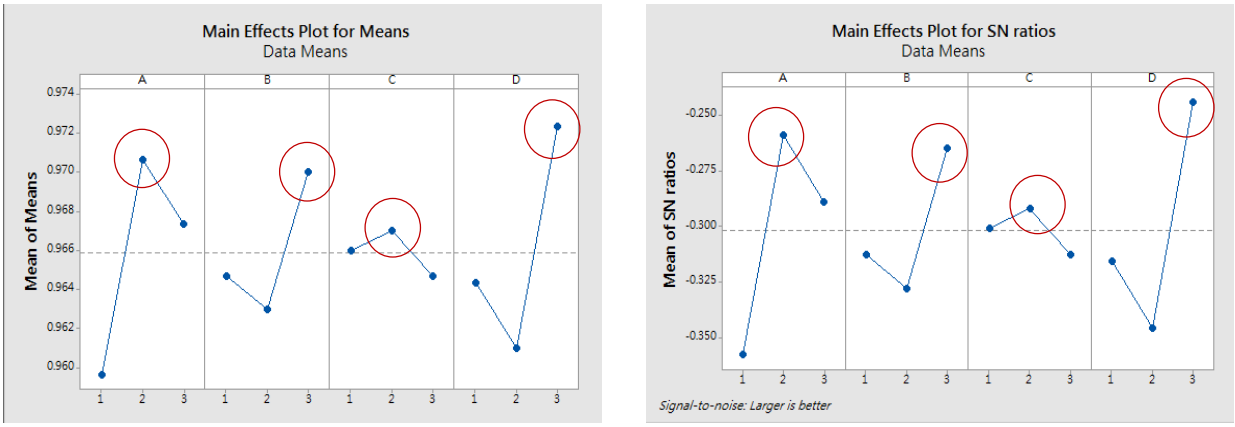

圖 3　Minitab 分析結果

最後從分析結果中得出各個因子最佳水準組合為 A2 B3 C2 D3，以最佳水準組合再做測試，希望可以得到更好的結果。最佳組合為表 5，準確度為 0.974。

表 5 最佳組合

	Dropout	Learning Rate	Epochs	Batch Size	Test Accuracy
最佳組合	0.3	0.001	70	64	0.976

(六) 四種不同模型結果比較

最後，本研究建立 SVC、LR 及 NB 三種機器學習演算法，對網頁位址進行釣魚網頁檢測辨識，以評估 MLP 深度學習模型之效度並選出最佳模型。從表 6 結果可得出，MLP 神經網絡模型比 SVC、LR 及 NB 三個機器學習模型在準確率上表現都較佳。

表 6　四種模型結果

Method	MLP	SVC	LR	NB
Precision	0.976	0.901	0.897	0.916
Recall	0.923	0.9	0.9	0.92
F1 score	0.949	0.9	0.9	0.92

四、結論與未來展望

本次研究中建立了釣魚網頁檢測系統模型，讓所有網絡使用者能在任何時候對於網頁有釣魚攻擊疑惑時能自行進行檢測，令複雜的網絡釣魚攻擊無所遁形，大大降低網絡犯罪分子利用網絡釣魚手法竊取我們的個人和財務信息之機會。

由於本研究模型精確度高，因此網絡使用者使用檢測系統後能放心相信結果
未來期望可以透過增加資料集數據，使模型往更精準結果優化

五、參考資料

https://chih-sheng-huang821.medium.com/機器學習-神經網路-多層感知機-multilayer-perceptron-mlp-含詳細推導-ee4f3d5d1b41
https://chih-sheng-huang821.medium.com/機器學習-支撐向量機-support-vector-machine-svm-詳細推導-c320098a3d2e
https://cchia.kmu.edu.tw/images/文章/2-Logistic_Regression_介紹.pdf
https://iter01.com/573828.html