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

Outline

01
背景介紹

02
研究方法

03
模型訓練與績效

04
結果與未來展望

背景介紹

背景介紹

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

研究目的

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
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。

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 透過建立網絡釣魚網頁檢測系統模型，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。

When 網絡使用者對於網頁有網絡釣魚攻擊疑惑時

Where 網絡世界

Who 所有網絡使用者

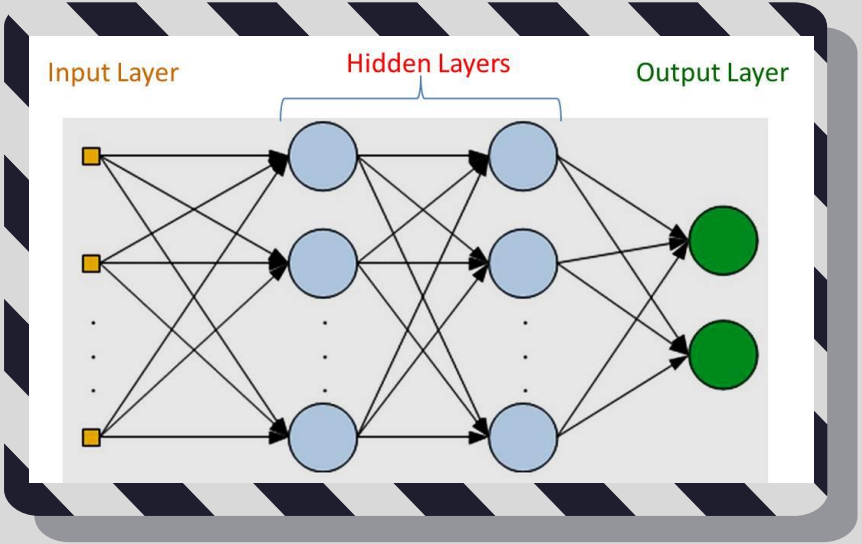
How MLP/ SVC/ LR/ NB

研究方法

多層感知器（MLP）

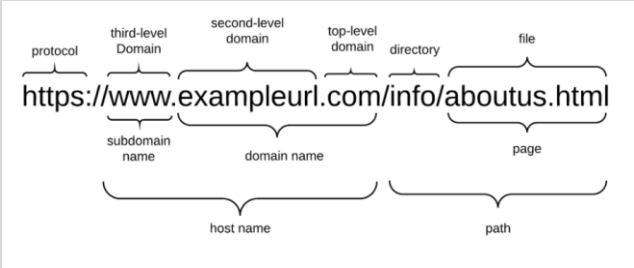
- MLP 是深度神經網路(DNN)的一種special case。
- MLP 是一種**前向傳遞類神經網路**，
- 至少包含三層結構(層感知輸入層、隱藏層和輸出層)，
- 並且利用到「**倒傳遞**」的技術達到model learning的監督式學習
- MLP神經網路利用gradient descent找最佳參數解

最後帶入MLP內的前向傳遞即可得到最後的預測值。



模型訓練與績效

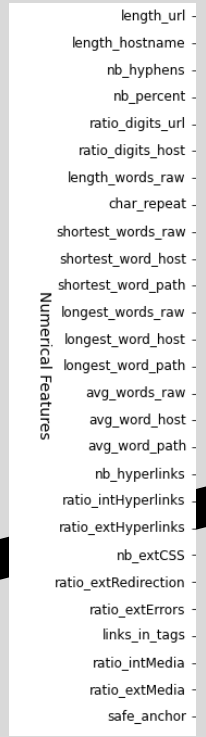
資料介紹



資料來源

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)，
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



資料前處理

Step 1

資料狀態轉碼

每筆資料中的網頁地址(url)及其狀態(status)
Encode 為1,0

- 合法網頁為1
- 釣魚網頁為0

	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

資料前處理

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

	url	status	scheme	netloc	path	params	query	fragment
0	http://00324ll.moonfruit.com	phishing	http	00324ll.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	/			

```python
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
```

提取有用特徵

資料前處理

Step 4

URL中提取資訊&標籤

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、
標點符號等資訊加上標籤，並把URL列刪除

```
df_grp_y = df_grp['status'] #It was df_grp_y
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
0	28	com	False	0	0	0	0	0	0	1
1	126	com	False	4	0	0	1	1	10	1
2	63	com	False	0	0	0	0	2	34	1
3	84	com	False	0	0	0	0	2	34	1
4	80	com	False	0	0	0	0	2	34	1
...	...	...	...	...	...	...	...	...	...	...
11424	28	com	False	0	0	0	0	0	0	1
11425	76	com	False	0	0	0	0	3	24	3
11426	145	io	False	0	0	1	0	3	33	0
11427	31	com	False	0	0	0	0	1	1	1
11428	27	site	False	0	0	0	0	1	1	1

從URL中提取長度/TLD/IP等資訊並標記

資料前處理

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

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

label轉碼

Step 6

資料切分資料分割 (訓練與驗證集)

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)
```

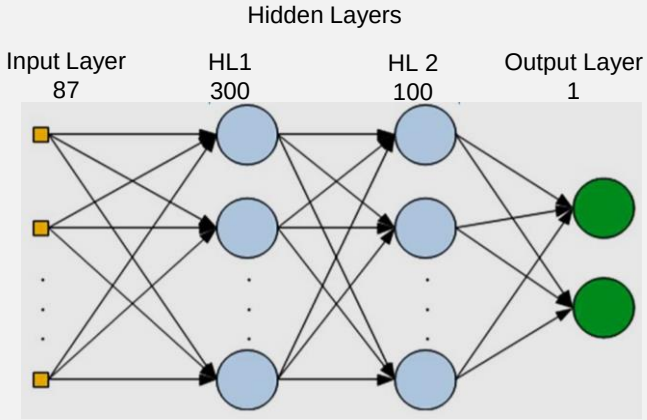
模型建立與訓練

MLP模型架構為：Input Layer ；兩個隱藏層；Output Layer

- 各層 Layer 節點數: Input : 87；隱藏層1 : 300；隱藏層2 : 100 ；Output : 1
- 各 Layer間應用 ReLU 函數
- Output Layer 應用 Sigmoid函數（二元分類）
- 加入Dropout層 減少訓練的時間&避免過擬合
- 使用Adam Optimizer自動調整學習率
- Loss function : Binary Cross Entropy

模型架構圖:

```
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)
)
```



參數優化

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

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

參數優化

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

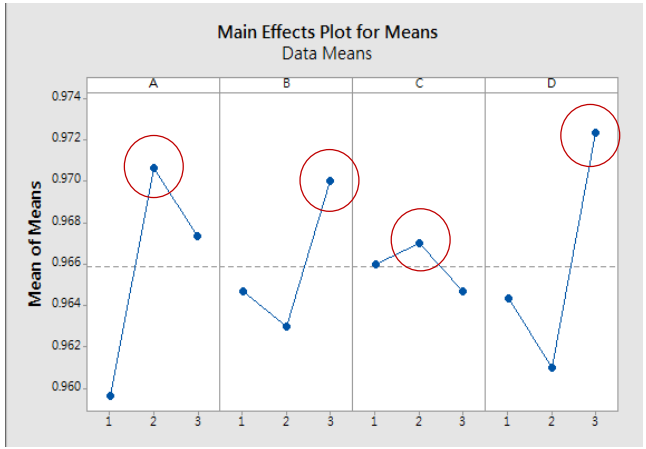
實驗設計

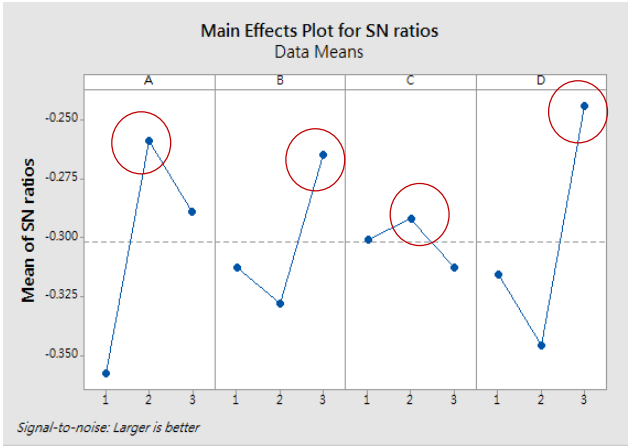
L9 實驗設計參數組合:

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

準確度最高的為實驗4，準確度達到0.975

實驗設計 Minitab結果





Response Table for Signal to Noise Ratios
Larger is better

Level	A	B	C	D
1	-0.3578	-0.3128	-0.3006	-0.3156
2	-0.2587	-0.3277	-0.2920	-0.3458
3	-0.2886	-0.2646	-0.3125	-0.2437
Delta	0.0991	0.0632	0.0206	0.1020
Rank	2	3	4	1

Response Table for Means

Level	A	B	C	D
1	0.9597	0.9647	0.9660	0.9643
2	0.9707	0.9630	0.9670	0.9610
3	0.9673	0.9700	0.9647	0.9723
Delta	0.0110	0.0070	0.0023	0.0113
Rank	2	3	4	1

最佳參數水準組合：A2 B3 C2 D3

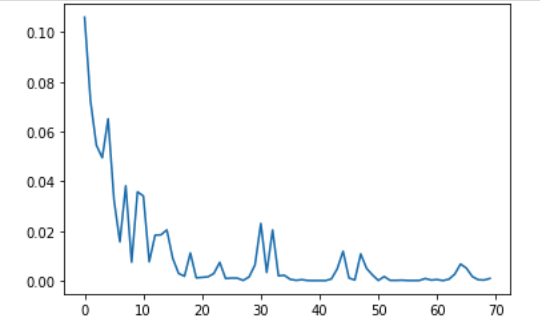
實驗設計 Minitab結果

最佳參數水準組合：A2 B3 C2 D3

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

最佳組合

Dropout	0.3
Learning Rate	0.001
Epochs	70
Batch Size	64



最佳參數水準 Test Accuracy : 0.976

模型評估與比較 compare with ML models

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

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

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。有著最佳預測效果

結果與未來展望

結果與未來展望

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