資料科學 期末專題

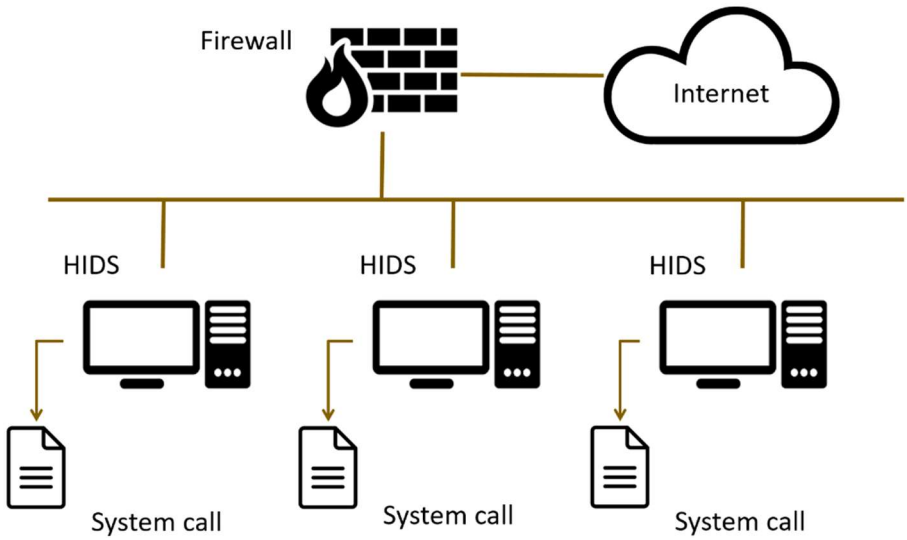
Transfer Learning for Host-based Intrusion Detection System

309513041 蔡政勳 309513121 李偉齊

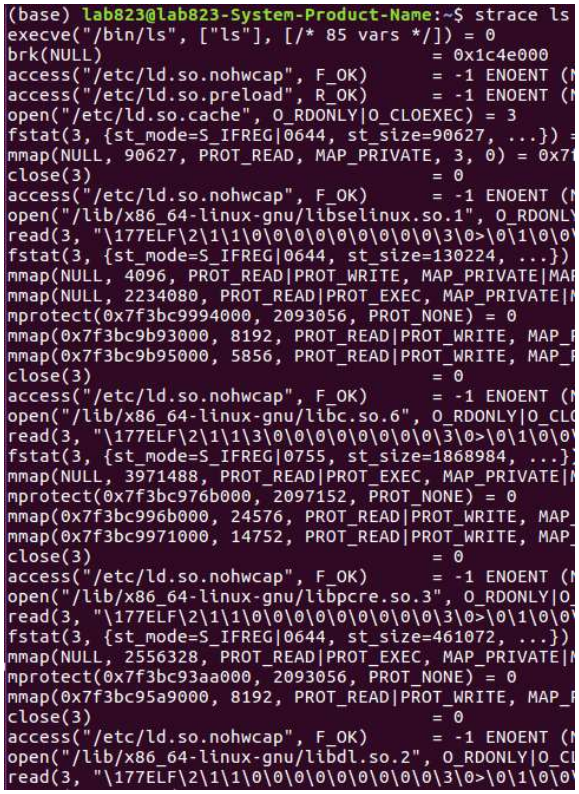
1.Introduction

在現在這個網路發達的時代，資訊安全的問題層出不窮的出現在我們生活周 遭，許多病毒以及駭客攻擊往往在我們沒有發覺時侵入我們的電腦，為了改 善此現象，我們希望能開發出一套入侵偵測系統，有別於一般的防毒軟體， 這套系統能更強力的偵測出許多未知的攻擊。

我們打算使用 user 的 short range system calls，編碼後當成 training data 來訓 練出一個 normal behaviors 的 model 可以偵測出與其不同行為的活動，用來 當作入侵偵測的一種方式。再利用 transfer learning 來使其可應用到其他 user 的 stations，目標是降低判斷錯誤率以及盡可能的在 real-time 實行。



2.Data Pre-Processing



我們使用 hw1 的對 ptt 爬蟲程式，在他進行時的 short range system calls 來當作正常行為的 training data, 使用的編碼方式是直接拿 system code 在 系統中的編號，此為第一種編碼方式。

但這種方法可能沒辦法完全表示資料的分布特性與他們之間的關係，我們想試試第二種方式:

使用 one-hot encoding 來做為資料的前處理方式

EX: 1 = [1 0 0 0 0 0 0 .....]

其中，使用 One Hot Encoding 的好處有:

1.離散特徵的取值之間沒有大小的意義，比如 color：[red,blue],那麼就使用 one-hot 編碼

2.使用 one-hot 編碼， 將離散特徵通過 one-hot 編碼對映到歐式空間，是因為，在迴歸，分類，聚類等機器學習演算法中，特徵之間距離的計算或相似度的計算是非常重要的，而我們常用的距離或相似度的計算都是在歐式空間的相似度計算，計算餘弦相似性，基於的就是歐式空間。

Source Domain

1. Only contains normal sequences.
2. Collected the system calls which generated from python crawler.

Target Domain

ADFA dataset

% time	seconds	usecs/call	calls	errors	syscall
42.31	0.010337	3	3664	504	read
21.09	0.005152	1	3839	1097	stat
10.06	0.002457	3	961		poll
9.49	0.002319	3	917		write
4.52	0.001104	2	493	24	ioctl
2.91	0.000710	1	500	32	open
2.91	0.000710	1	815		fstat
1.88	0.000460	1	482		close
1.39	0.000340	1	435		mmap
1.29	0.000314	1	360	3	lseek
0.70	0.000171	2	94		getdents
0.47	0.000115	0	452		select
0.41	0.000101	1	135		munmap
0.23	0.000057	1	81		mprotect
0.23	0.000056	1	53		brk
0.07	0.000016	1	24		futex
0.03	0.000007	1	8		socket
0.02	0.000004	2	2		bind
0.00	0.000000	0	1		lstat
0.00	0.000000	0	68		rt_sigaction
0.00	0.000000	0	1		rt_sigprocmask

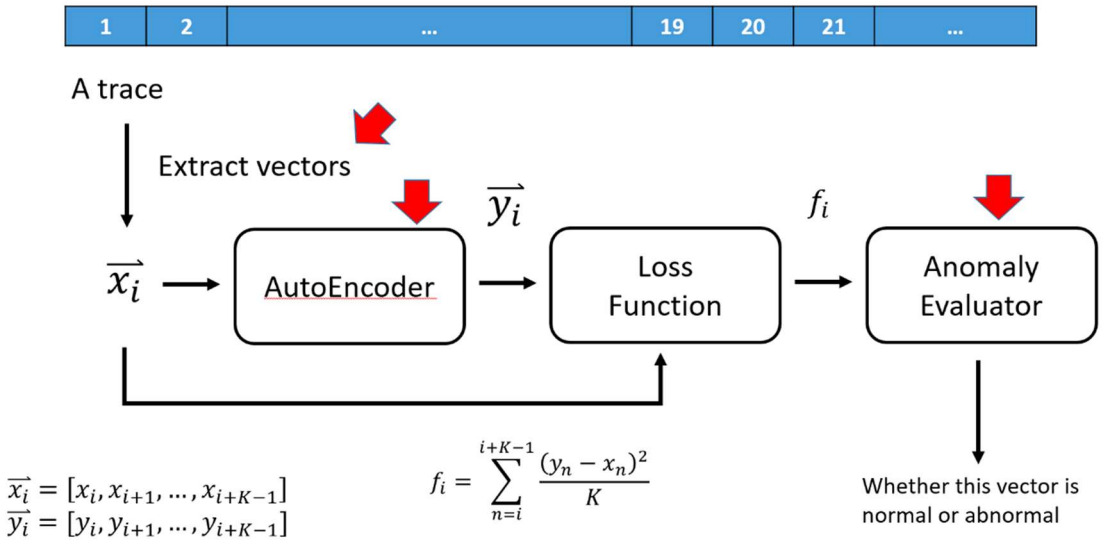
Data Type	Trace Count
Normal Training Data	833
Normal Validation Data	4373
Attack Data	10 Attack per Vector, 60 totally

6

而目標之一是在別的 host，甚至是系統都要可以運作，因此我們需要做 transfer learning(後面會說明)。

選用剛剛提到的爬蟲資料的 system call 作為 source domain 的資料，另外選用 ADFA 的 data 作為 target domain，詳細資料如上圖。

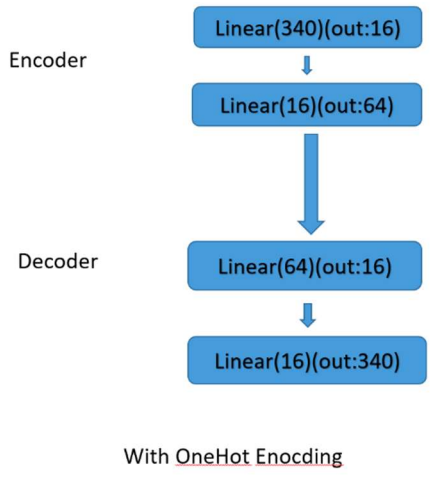
3.Model Structure

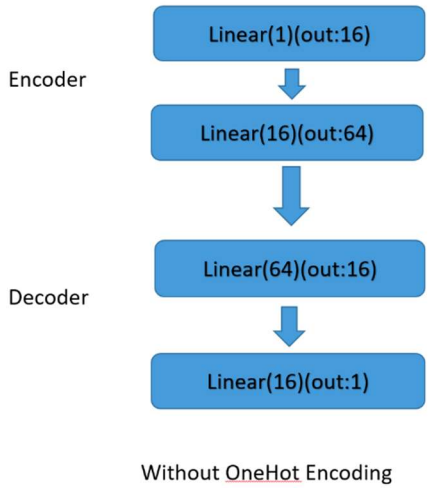


上圖整體架構是學長跟老師研究提出的碩士論文，將一段一段的 system code 送進 Auto Encoder 進行訓練，以 decoder 後與 input 資料的 mean square error 做為 loss function，訓練完成後及產生了一個異常偵測模型，接下來在對行為進行預測時，只要該行為與訓練資料中正常行為差異甚大時， 產生的 loss 也會過大，我們就能依此判斷出哪些行為是異常行為。

本篇論文使用 Auto Encoder 的方式來做異常偵測，我們提出以下不同的架構來偵測，比較跟原本的差異並期待降低偵測錯誤的情況，使這個 model 效果更好:

(1.)Auto Encoder(AE)

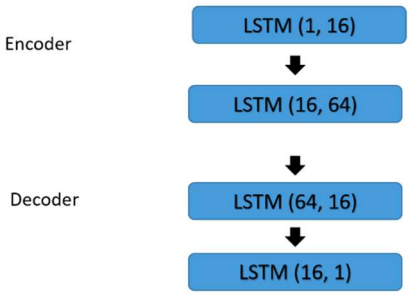




Auto encoder 會學習到 input data 的表示方式，並在輸出的地方盡可能還原成 input vector。

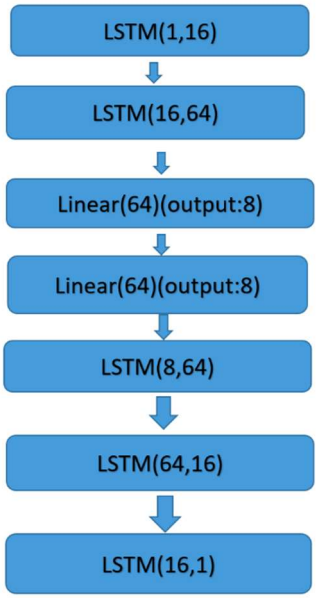
系統中的 system calls 共有 340 種，所以在有使用 one hot encoding 的部分，系統輸入是 340 維。

(2.)LSTM-AE

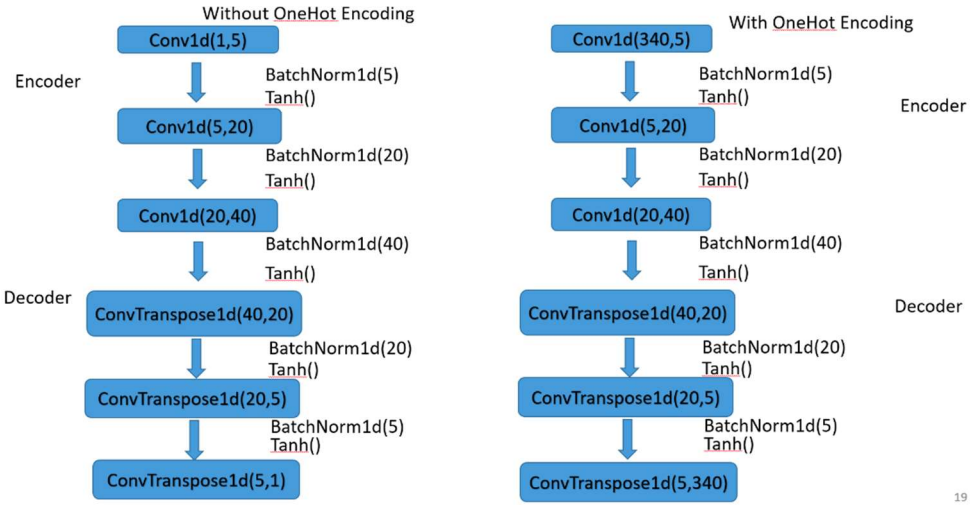


將 AE 中的 layer 替換成 lstm，期望可以因為有短記憶功能的 lstm 而表現好。而此架構下沒有使用 one hot encoding 是因為 input 的維度太高程式跑不動，之後發現了可以使用 PCA 來降維，是其中一個以後實驗改善的方向。

(3.)LSTM-VAE



(4.)CNN+AE

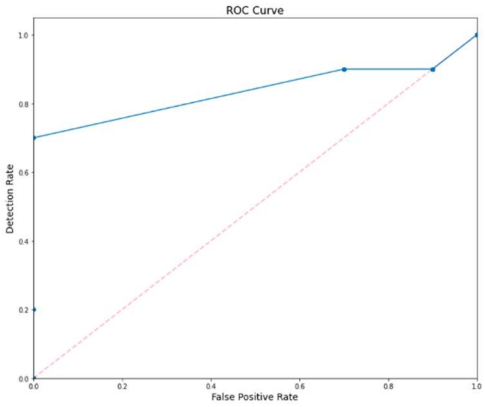


實驗結果:

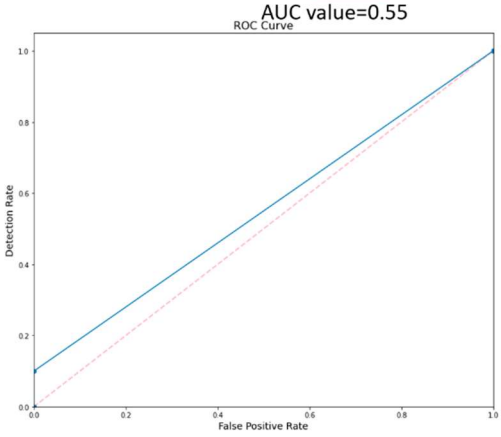
(1.)Test Source Domain

AE:

Without OneHot
AUC value=0.835

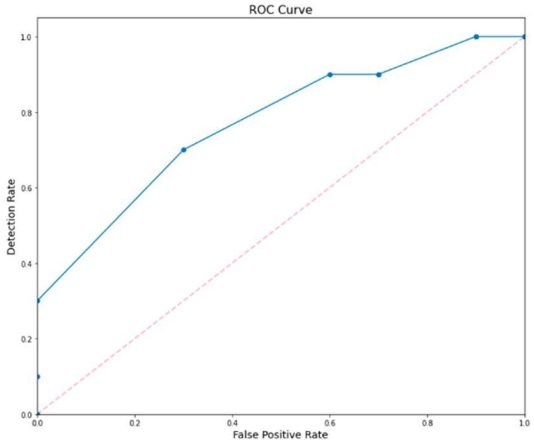


With OneHot
AUC value=0.55



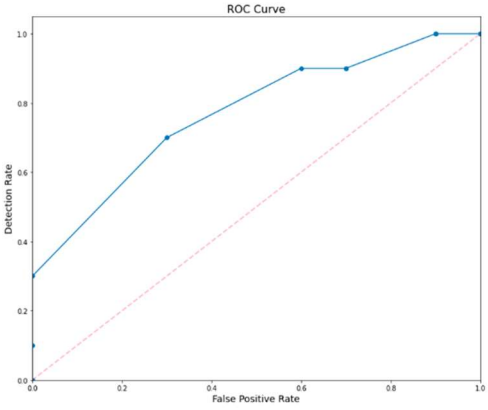
LSTM-AE

Without OneHot
AUC value=0.77



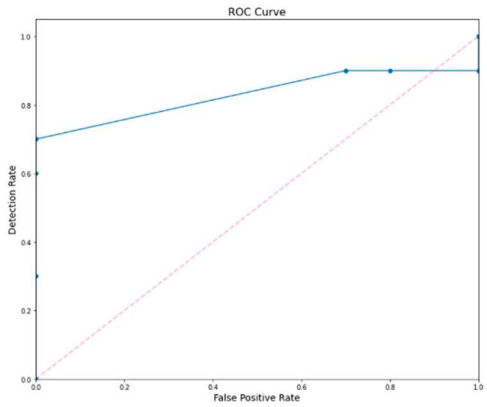
LSTM-VAE

Without OneHot
AUC value=0.77

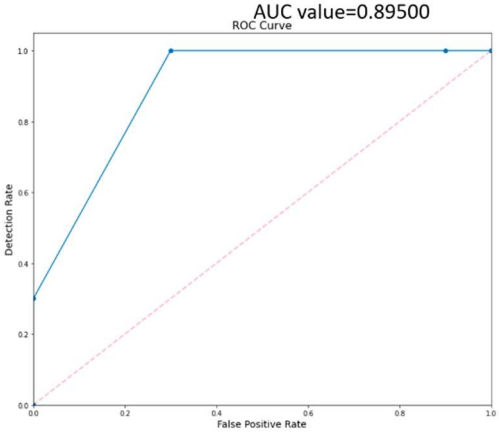


CNN-AE

Without OneHot
AUC value=0.83



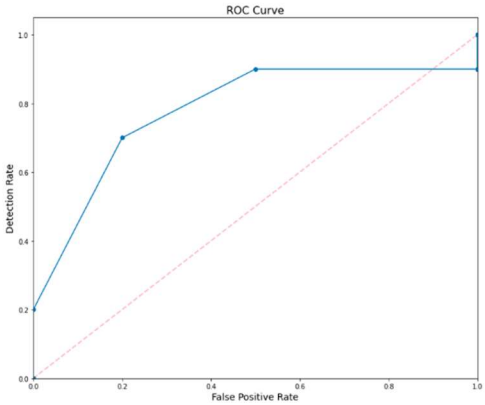
With OneHot
AUC value=0.89500



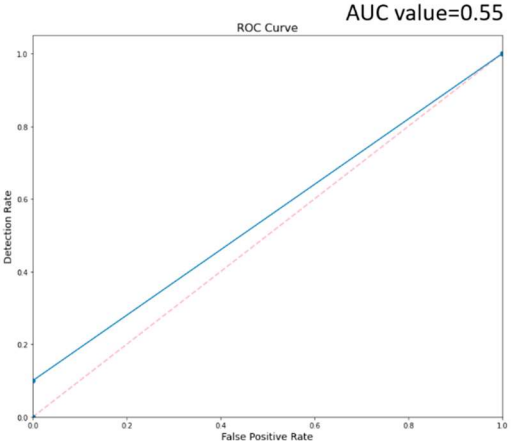
(2.)Test Target Domain

AE

Without OneHot
AUC value=0.78

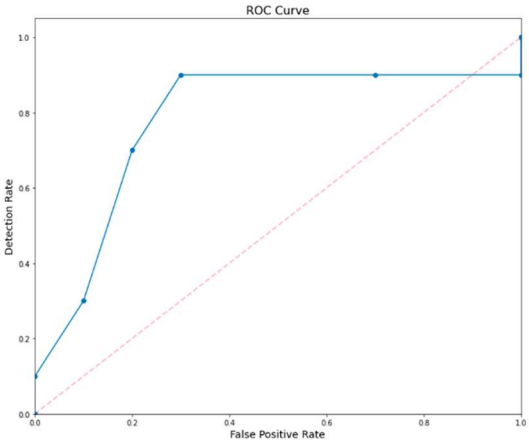


With OneHot
AUC value=0.55



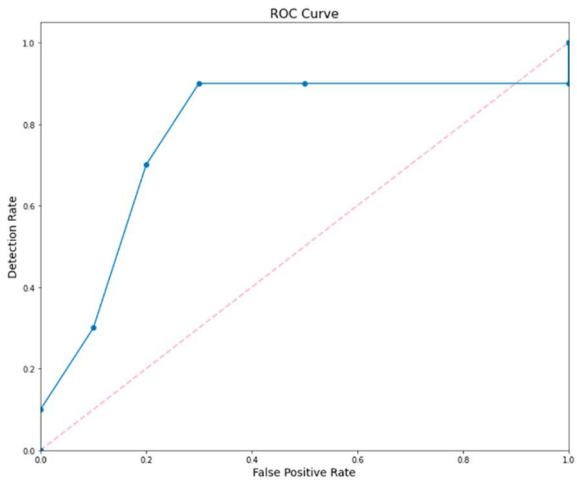
LSTM-AE

Without OneHot
AUC value=0.78



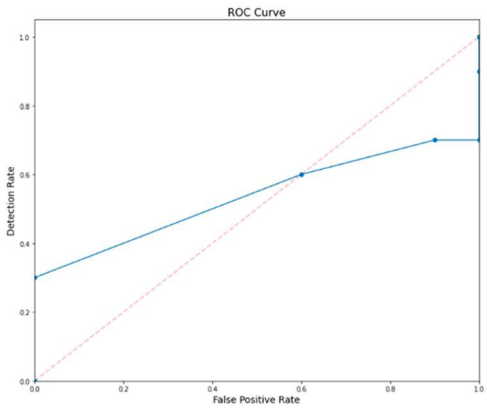
LSTM-VAE

lstm_vae model
AUC value=0.78

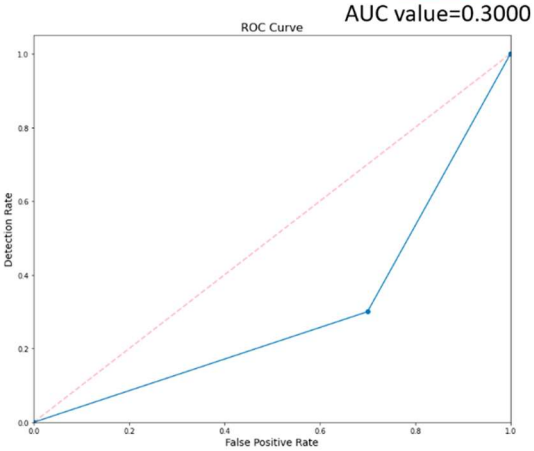


CNN-AE

Without OneHot
AUC-value=0.534999



With OneHot
AUC value=0.3000



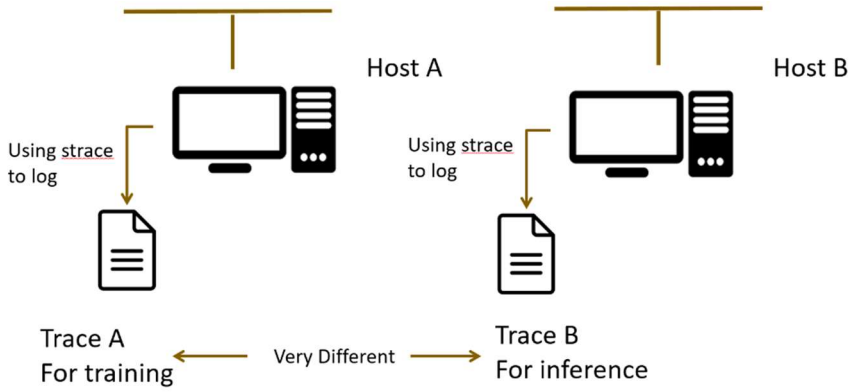
(3.)實驗結果比對

Test Source Domain	With OneHot ACC	Without OneHot ACC
Auto Encoder	0.550	0.835
LSTM-VAE	-	0.770
CNN-AE	0.895	0.830
LSTM-AE	-	0.770

Test Target Domain	With OneHot ACC	Without OneHot ACC
Auto Encoder	0.550	0.780
LSTM-VAE	-	0.780
CNN-AE	0.300	0.535
LSTM-AE	-	0.780

從上圖可以初步看出，在沒有使用 transfer learning 下，target domain 下的表現都很差。所以我們需要繼續將這部分的成果延續到 transfer learning。而我們決定採用表現最好最穩定的 LSTM-AE，跟論文採取的方式一樣。

4.Transfer Learning

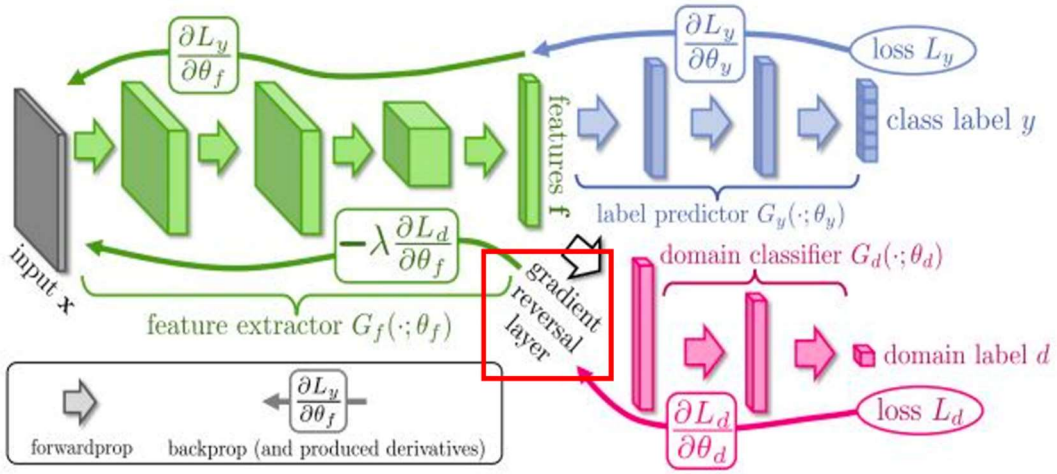


The normal trace in Trace B will be considered as attack by HIDS A.

如同前面所提到的，我們希望在各個不同的 host 之間都可以使用我們的偵測系統，所以使用 transfer learning。

DANN

使用 Domain-Adversarial Training of Neural Networks(DANN)的模型來進行。



(圖片來源: https://zhuanlan.zhihu.com/p/51499968)

DANN 主要分成三個部分 ：

(1.)Feature Extractor : 前面幾層的特徵萃取我們可以視為一個空間轉換，將 Source Data 與 Target Data map 到一個新的特徵空間中。

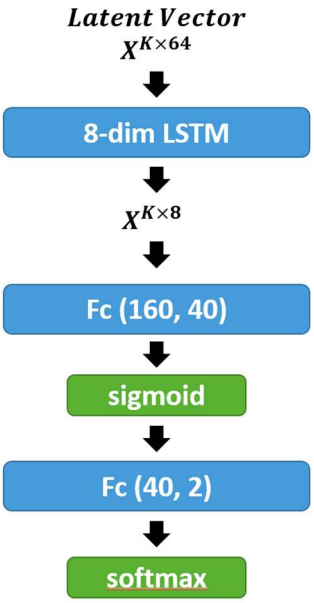
(2.)Domain Classifier :Domain Classifier 用來確認特徵空間中的點是來自 Source Domain 還是 Target Domain。

(3.)Label Predictor :Label Predictor 則是對特徵空間中的點進行分類。(也就是根

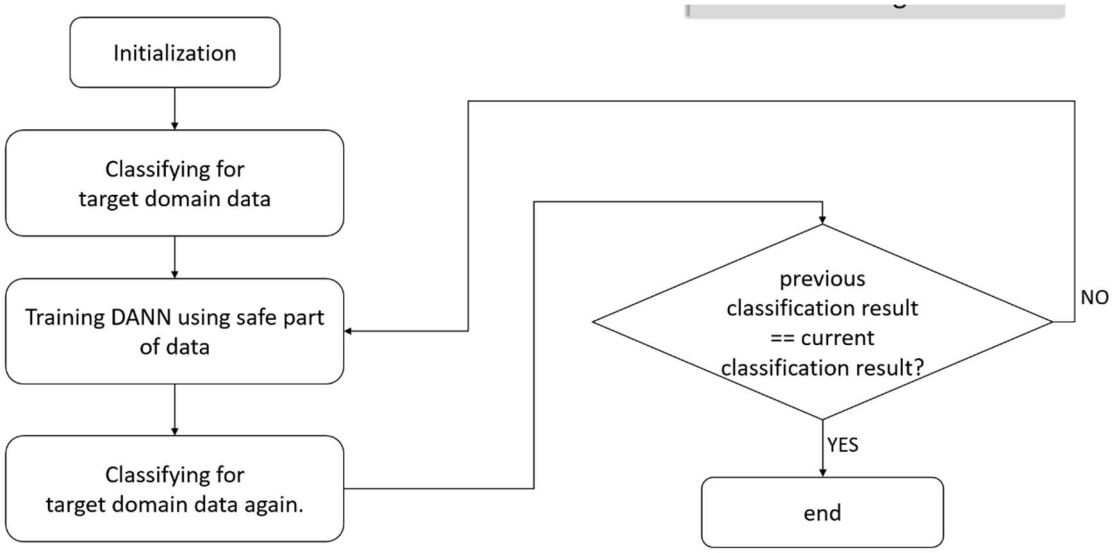
據特徵萃取後的特徵來決定要怎麼分類資料)

所以整個 DANN 在做的事情就是，資料進入 Feature Extractor 後，先要將來自 Source Domain 及 Target Domain 的資料分布讓其盡量相似。
但 Domain Classifier 本來是希望能將 Domain 分得越開越好，因此必須在 Domain Classifier 與 Feature Extractor 之間加一個梯度反轉層 (Gradient reversal layer) 便可以達到相反的效果。

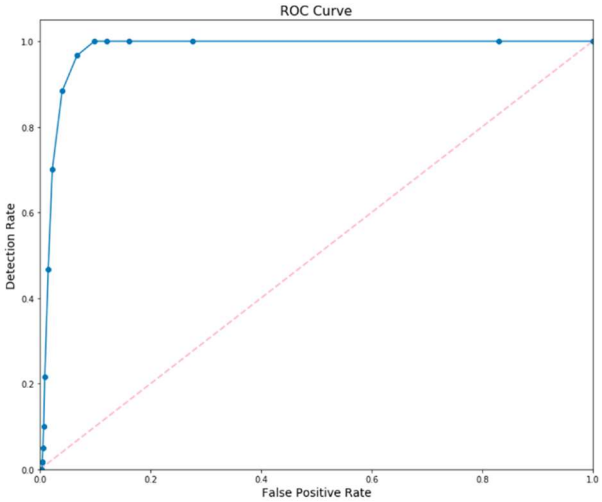
(Domain Classifier 架構:)



整個使用 DANN 流程:



實驗結果



DANN AUC value=0.9784004269594389

可以看到，在使用 DANN 後整個模型的判斷正確率是相當高的。

5.Conclusion and Future Work

在加入了 DANN 的模型後，整個入侵偵測系統在不同主機下的運行似乎變得可行，不過還有很多可以努力的方向，也想到了幾個可以接下去做的地方:

(1.) GAN increase attack data

我們在攻擊部分的測試資料比較少，因此可以使用 Cycle GAN 來生成攻擊的資料來使用。這部分在網路上已經有看到論文發表，細節的部份我們之後的實驗去詳細了解。

(2.) One Hot Encoding and PCA

前面有提到 one hot encoding 下作為有些模型的 input 維度會太高，因此可以使用 PCA 的方法來降維，大致上意思就是取最大特徵值所對應的特徵向量來做為代表。

而 one hot 加上 PCA 的好成效也廣為流傳。

(3.) Meta-learning-MAML

Meta-Learning 是一種讓機器學習學習方法的方式，在此模型下可以給予機器不同 task 使其改善學習的方法。並且幾乎可以適用在任何類神經網路的模型，期待在 target domain 的表現會更快好。詳細過程以及實作也是我們日後努力的目標。

6.Reference

- A High-Performance Deep Learning Architecture for HIDS - Tsern-Huei Lee, James C. Juang
- Ganin, Y., and Lempitsky, V. Unsupervised domain adaptation by backpropagation.
  ICML'15: Proceedings of the 32nd International Conference on International

Conference on Machine Learning (Jul. 2015), 1180–1189
- W. M. Kouw, M. Loog.” An introduction to domain adaptation and transfer learning.” arXiv preprint arXiv:1812.11806, 2018
- F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. arXiv preprint arXiv:1503.03832, 2015
- A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737, 2017
- E Hoffer and N Ailon. Deep metric learning using triplet network. Similarity-Based Pattern Recognition, 2015