Regular Paper

Communications of the CCISA
Vol. 26 No. 1 Feb. 2020

基於協同神經網路與彙整機制建構零時差攻擊偵測系統

陳志龍[1]、魏國瑞[2]、陳映親[1]、李榮三[1,*]
[1] 逢甲大學資訊工程學系
[2] 三甲科技股份有限公司
tsche107389@gmail.com, weiray654@gmail.com, ycchen.blythe@gmail.com,
leejs@fcu.edu.tw

摘要

基於特徵檢測的入侵偵測系統在防護網路攻擊中扮演著十分重要的角色，然而針對已知特徵進行檢驗的做法，卻存在無法偵測零時差攻擊之缺陷，造成入侵偵測系統容易忽略新型惡意行為，進而無法有效地分辨正常與惡意流量，導致企業或個人資料外洩而造成極大的損失。在本篇論文中，我們結合 AutoEncoder 以及深層神經網路，提出可檢測未知攻擊的入侵偵測系統，不僅可檢驗已知的惡意行為，亦可改善無法抵禦零時差攻擊之缺點。在本系統架構中，零時差攻擊偵測模組負責辨別收集的流量是否有未知攻擊的出現；而已知攻擊分類方法則作為攻擊類型的分類器以進一步判斷流量具體屬於何種已知攻擊。接著，為使系統學習未知攻擊並將其轉換為已知攻擊，我們加入攻擊彙整機制，透過基於 DBSCAN 的具投票制度之分群法，將特徵相近的未知攻擊聚合成新型態的攻擊類別，使系統的攻擊偵測能力可隨著惡意攻擊的數量而不斷成長。實驗結果表明，本論文所提出的新型態入侵偵測系統能有效的偵測未知攻擊，並具備優良的分類結果。

關鍵詞：入侵偵測系統、零時差攻擊、AutoEncoder、彙整機制

Regular Paper

Communications of the CCISA
Vol. 26 No. 1 Feb. 2020

Zero-day Intrusion Detection System based on Dual Neural Network and Aggregation Mechanism

Chih-Lung Chen[1], Kuo-Jui Wei[2], Ying-Chin Chen[1], Jung-San Lee[1, *]
[1] Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 407, Taiwan
[2] AAA Security Technology Company, Limited
tsche107389@gmail.com, weiray654@gmail.com, ycchen.blythe@gmail.com,
leejs@fcu.edu.tw

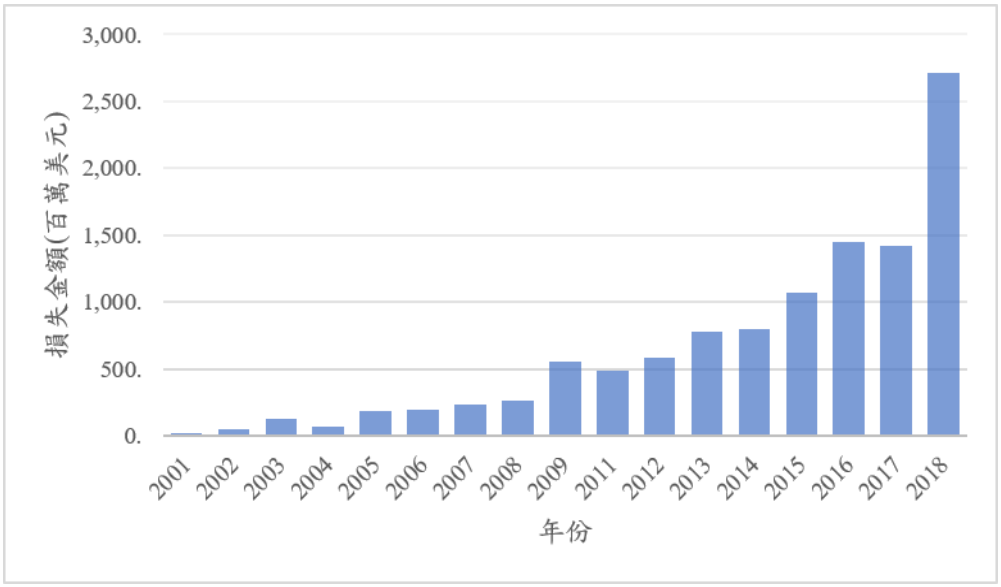
Abstract

Despite signature-based intrusion detection system(IDS) has played an important role in the field of cyber security, there remains a crucial challenge that the zero-day attack is hard to be solved. This drawback may bring a large amount of loss to an enterprise or an individual. In order to address above issue, we aim to propose a novel IDS framework which is able to conquer zero-day attacks. The framework consists of an AutoEncoder and a deep neural network(DNN), where AutoEncoder is applied to detect zero-day intrusion, and DNN is employed for classifying known attack, respectively. In particular, we have introduced aggregation mechanism based on DBSCAN algorithm and voting system for sorting the zero-day samples and retraining the IDS. The experimental results have demonstrated that the new method can solidly work in a zero-day attack detection and known attack classification.

Keywords: IDS, Zero-day attack, AutoEncoder, Aggregation

壹、前言

隨著科技高度發展，網際網路已成為人們生活中密不可分的一部份，無論是電腦、行動裝置，甚至家電皆能連上網路。網際網路的拓展逐漸改變現代人的生活習性，如商務行為、通訊方式或網路學習等。在這些新穎技術帶來便捷生活的同時，卻也成為網路攻擊逐漸猖獗的誘因，當使用者於電子裝置中儲存越多個人敏感資訊，駭客進行惡意行為賺取非法利益的動機也越大，個資外洩的風險將會逐漸上升。根據美國聯邦調查局的犯罪投訴中心(Internet Crime Complaint Center, IC3)統計[27]，近年來因網路攻擊而造成的實際損失逐年上升，如圖一所示，2016 年造成 14.5 億美元的損失，到 2018 年則已達到 27.1 億美元，在短短兩年內增加將近一倍，此意味著駭客的惡意行為對於企業或是一般民眾所造成的損害在未來將持續擴大。



圖一：IC3 統計歷年網路攻擊造成損失金額

駭客進行網路攻擊的手法變化多端，舉凡利用網路漏洞入侵伺服器、散播電腦病毒以感染主機，透過殭屍網路進行分散式阻斷服務攻擊等，皆是造成經濟損失的威脅來源。其中最令人感到棘手的便是進階持續性滲透(advanced persistent threat, APT)攻擊，駭客為了非法利益而針對特定對象進行長時間的埋伏及觀察，並為其量身打造攻擊手法，藉以入侵目標主機並竊取機敏資料。這些被設計出來的攻擊手法多為首次出現，因此於攻擊當下並無相對應的防護措施，故入侵成功率較高，這類的攻擊稱為零時差攻擊(zero-day attack)。高成功率使得零時差攻擊發生的頻率不斷上升，根據[18]顯示，2017 年間，零時差攻擊於所有違規事件中所佔比例為 25%，而到 2018 年上升至 37%；再者，2018 年對企業組織實際造成影響的攻擊中，有 76%為零時差攻擊[28]。上述資料顯示，零時差攻擊發生的頻率不只逐年增加，其更是造成企業重大損失的罪魁禍首，因此如何防禦零時差攻擊顯然已成為捍衛資訊安全的首要研究目標。

在眾多防護手段中，以惡意行為檢測效果與花費成本間的平衡作為考量，使入侵偵測系統成為企業最常選擇的防護措施之一，而如何建構強力的入侵偵測系統，以防禦四面八方的網路攻擊，也隨之成為熱門議題。迄今為止已有許多入侵偵測系統相關的研究，可根據其檢測方法分為異常偵測型(anomaly-based)[9][25]以及特徵檢測型(signature-based)[1][16][26]，兩者具有不同的偵測方式及特性。異常偵測型透過流量與正常行為之間的差異進行檢測，其概念如同白名單，當流量特徵與已知正常行為相差甚遠時，即認定該流量屬於惡意行為。透過上述的方式，即使該惡意行為是首次出現，此類入侵偵測系統亦能偵測之，然而卻無法得知其是透過何種手法進行攻擊，並且十分依賴做為基準的正常流量之特徵，若基準內的正常流量不夠完整，無法涵蓋所有的特徵範圍，誤判率(false alarm rate)將居高不下；而特徵檢測型入侵偵測系統則與黑名單的概念相似，是基於流量特徵與各種惡意行為的相似度判斷其是否屬於惡意攻擊，由於此種檢測方式已事先得知各攻擊種類的行為特徵，因此可透過流量與各已知攻擊類型的特徵相似度，判斷其屬於何種惡意行為。但由於檢測範圍僅涵蓋已知的行為特徵，故面對零時差攻擊時反而無法有效地進行偵測。若可改善此缺陷，使特徵檢測型入侵偵測系統具備發現未知攻擊之能力，將可更全面地偵測惡意流量，降低零時差攻擊成功的機會。

隨著近年來機器學習與深度學習的崛起，入侵偵測系統的研究開始引入兩者以提升其偵測能力，如 Al-Qatf 等人[2]透過結合 AutoEncoder[23]與 support vector machine(SVM)[5]增加分類的準確率；而 Shone 等人[26]利用 AutoEncoder 以及 random forest[3]從資料中取出重要特徵進行分類以提升準確率，並減少訓練模型的時間；Zhang 等人[30]則使用 long short term memory(LSTM)[15]，除提升準確率外，更降低誤判率。儘管上述所提及的研究皆導入機器學習或深度學習，專注於如何建構高準確率的入侵偵測系統，但當面對未知攻擊時，依舊無法有效地進行分類，因此仍存在無法偵測零時差攻擊之問題[4]。

另一方面，隨著網路技術的快速發展，亦衍伸出許多不同的通訊協定，其中網路流量的特徵也隨之變化，例如每個封包所傳遞的資料量大小等相關資訊皆隨時間而演進。因此為適應不斷變化的網路流量，系統持續更新亦是入侵偵測系統中十分重要的一環，若系統可伴隨著網路發展進行自我學習，能分辨更多新型態網路攻擊，並隨著時間推移而提升其偵測能力，將不易因無法適應網路的變化而遭到淘汰，能大幅提升其可用性。但在變化莫測的網路世界中，零時差攻擊的手法相當多樣化，且在眾多零時差攻擊樣本中，攻擊手段可能相差甚遠，若將全部樣本視為同類型並用於更新模型，將可能降低模型的分類能力[29]，因此事前區分這些未知攻擊類型顯然是提升模型分類能力之前置工作。

為解決傳統方法無法偵測零時差攻擊之問題，本計畫基於 AutoEncoder 提出一個可檢測未知攻擊的入侵偵測系統，使模型能夠判別該流量是否屬於零時差攻擊。此系統亦結合深層類神經網路作為分類器，辨識該攻擊是透過何種手法進行惡意行為。此外，為使入侵偵測系統在未來能夠持續使用，我們基於 density-based spatial clustering of

applications with noise(DBSCAN)[6]，在架構中加入攻擊彙整機制，在此機制啟動時，將針對被系統偵測的零時差攻擊進行分群，最後產生結果可作為模型自我學習的訓練樣本。研究架構具備以下三項功能：

(1) 零時差攻擊檢測：根據輸入資料與已知類型的特徵值進行相似性比較，並判斷其是否為零時差攻擊。
(2) 已知攻擊分類：針對輸入資料進行特徵分析，判斷其屬於正常或惡意行為，若為惡意行為，則判定其使用何種攻擊手法，可提供資安人員進行相對應的處理。
(3) 攻擊彙整機制：系統將蒐集偵測出的零時差攻擊，並且對其進行分群，以分離使用不同手法的攻擊，最後所產生之結果可作為模型的訓練樣本。

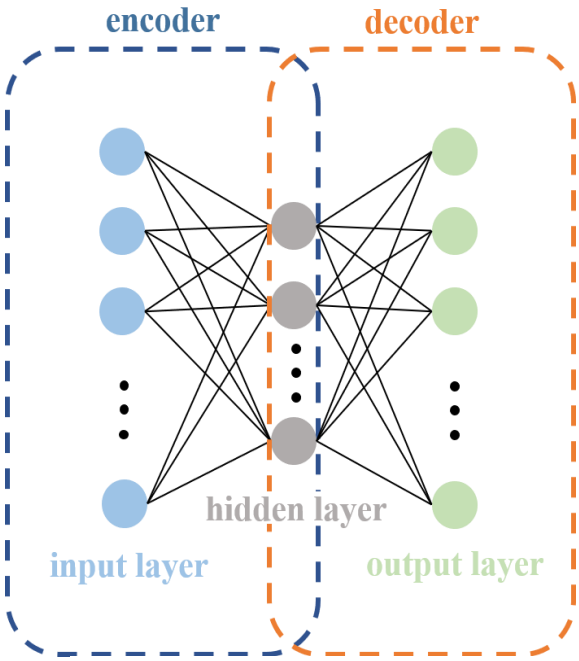
本論文將於第二章針對本研究中所使用的相關技術進行講解，第三章為研究方法及流程架構，接著實驗結果與分析將於第四章中進行說明，最後於第五章提出關於本研究的結論。

貳、文獻探討

深度學習的技術已廣泛地應用於入侵偵測系統，在本論文中，我們引入深度學習中的 AutoEncoder，扮演模型架構中零時差攻擊檢測的角色，使模型具備分離舊有及新興攻擊類型的能力。在後續進行零時差攻擊樣本的資料分群時，我們使用 DBSCAN 分群演算法，根據資料之間的相似度進行分群。在本章節，我們將針對 AutoEncoder 之架構及特性進行介紹，並說明 DBSCAN 演算法的運行流程。

2.1 AutoEncoder

AutoEncoder 是一種特殊的神經網路架構，許多研究將其用於特徵提取以及資料降維，主要利用大量無標籤的資料進行非監督式學習，使模型的輸出結果與輸入資料相似。AutoEncoder 的架構如圖 2 所示，主要可分為輸入層、隱藏層以及輸出層，其中特別的是，隱藏層的神經元數目小於輸入層的神經元數目，並且輸出層的神經元數目與輸入層相同。此架構中，從輸入層到隱藏層之部分被稱為編碼器(Encoder)；而從隱藏層到輸出層則被稱為解碼器(Decoder)。



圖二：AutoEncoder 之架構圖

在 AutoEncoder 中，編碼器會將輸入資料壓縮成較低維度的潛在特徵，再透過解碼器將其還原回原始資料，使輸出資料近似於原本的輸入資料，以驗證潛在特徵是否能完整表示原始資料，如公式(1)及公式(2)所示，編碼器與解碼器函式分別可用 f 及 g 表示，並以 x 與$\hat{x}$代表輸入以及輸出資料，當中的 W_f、W_g、b_f、b_g 分別為編碼器與解碼器中神經元的參數，σ 為激勵函數。輸入資料進入編碼器時，經過神經元內的參數運算後產生潛在特徵 z，再經由解碼器中的神經元計算出$\hat{x}$。

$$z = f(x) = \sigma(W_f \cdot x + b_f). \quad (1)$$

$$\hat{x} = g(z) = \sigma(W_g \cdot z + b_g). \quad (2)$$

為達到使輸出資料$\hat{x}$近似於輸入資料 x，在模型訓練階段，使用損失函數 $loss_{AE}$ 來調整 AutoEncoder 中的參數 W_f、W_g、b_f、b_g，如公式(3)所示，其中 $loss_{AE}$ 為計算 x 與$\hat{x}$之間差異的函數，其中最常擔任此角色的函數為均方差(mean square error, MSE)，其輸出稱為「重構誤差」，當 $loss_{AE}(x, \hat{x})$愈小，表示 x 與經過 f 與 g 後所產生的$\hat{x}$愈接近。

$$loss_{AE}(x, \hat{x}) = MSE(x, \hat{x}) = \| x - \hat{x} \|^2. \quad (3)$$

2.2 DBSCAN

在攻擊彙整機制中我們加入 DBSCAN 分群演算法，以下將進行此演算法的介紹。在演算法中以 D 表示所有資料點的集合，使用 $dist(p, q)$表示 p 與 q 兩點之間的距離，$N_{Eps}(p)$則是與點 p 之間的距離小於 Eps 的點集合，可表示為 $N_{Eps}(p)=\{q \in D | dist(p, q) \leq Eps\}$，並設定一門檻值 $MinPts$，且定義以下概念：

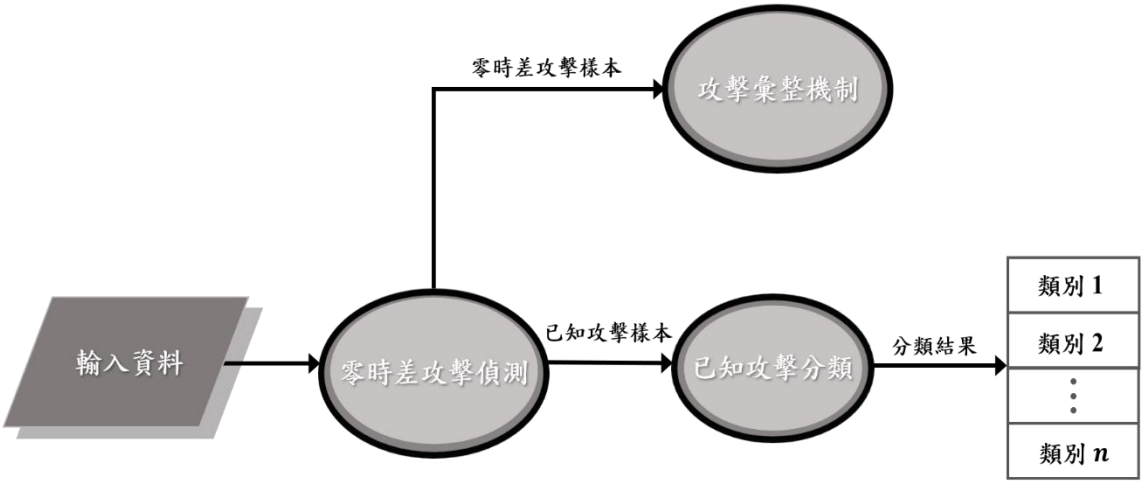
核心點：若 N_{Eps}(p)的數量大於 *MinPts*，p 被認為是核心點。
邊緣點：若 N_{Eps}(*p*)的數量小於 *MinPts*，*p* 被認為是邊緣點。
直接密度可達：若 *q* 為一核心點，且 $p \in N_{Eps}(q)$，則我們稱 *p* 從 *q* 直接密度可達(*p* is directly density-reachable from *q*)。
密度可達：若我們稱 *p* 從 *q* 密度可達(*p* is density-reachable from *q*)，表示 *p* 與 *q* 之間存在著一條由多個點所組成的鏈 $p_1, p_2, \ldots, p_n$，其中 p_1 為 *p*，而 p_n 為 *q*，並且當 $i \in \{1, 2, \ldots, n\text{-}1\}$時，$p_i$ 從 p_{i+1} 直接密度可達。

根據以上定義，DBSCAN 演算法透過下列執行流程將資料點分為數群：

- Step 1: 任意挑選資料點集合 D 內的點 q。
- Step 2: 若 *q* 屬於邊緣點，返回 Step 1；反之，若 *p* 為核心點則執行 Step 3。
- Step 3: 建立一最大的集合 $S = \{p \in D | p \text{ is density-reachable from } q\}$，並形成群集。
- Step 4: 重複 Step 1 至 Step 3，直到所有點都處理完畢。

參、研究方法

我們透過結合零時差攻擊偵測、已知攻擊分類以及攻擊彙整機制以建置完整的入侵偵測系統，其架構如圖 3 所示。首先，欲偵測之流量輸入至系統並進行零時差攻擊偵測，判斷其是否為前所未見的流量類型。此時，若流量類型屬於已知類型則進行分類，判斷其為何種類型之惡意行為，以進行相對應的處理；反之，則系統判定為零時差攻擊並將其暫存。接著，當系統蒐集一定數量的零時差攻擊樣本時，將啟動攻擊彙整機制，分離差異性較大的攻擊樣本，並彙整較相似的攻擊類型，做為未來系統更新的訓練樣本。以下將針對零時差攻擊偵測、已知類型分類以及攻擊彙整機制進行介紹。



圖三：系統架構圖

3.1 零時差攻擊偵測

本論文使用 AutoEncoder 進行零時差攻擊偵測，區分零時差攻擊與已知類型。當資料 x 輸入至系統中，首先透過 AutoEncoder 以及其損失函數 $loss_{AE}$ 計算重構誤差作為該筆資料的離群分數，在本論文中我們使用 MSE 作為其損失函數，並且使用 rectified linear units(ReLU)[20]做為公式(1)(2)中 f 與 g 的激勵函數。透過公式(1)-(3)計算出離群分數後，可藉由公式(4)判斷 x 的類型，當該分數高於預先設定的門檻值 TH，x 將被判定為新類型；相反地，當分數小於等於 TH 時則認為 x 屬於舊有類型。

$$\begin{cases} zx\ (zero\text{-}day\ attack),\ loss_{AE}(x,\hat{x}) > TH, \\ kx\ (\text{known class}),\ loss_{AE}(x,\hat{x}) \le TH. \end{cases} \tag{4}$$

3.2 已知類型分類

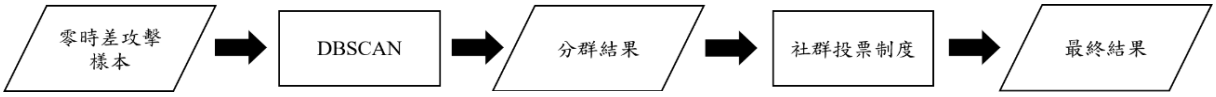
當輸入資料 x 被判定為已知類型 kx，將會進入此元件，由分類器判定其屬於何種類型。在此架構中，由深層類神經網路(deep neural network, DNN)擔任分類器之角色，此 DNN 主要由一輸入層、一個隱藏層以及一輸出層所組成。在隱藏層中，我們使用 ReLU 做為激勵函數，防止 DNN 在訓練時出現梯度消失[14]的問題，避免模型無法進行有效的分類。

$$loss_{DNN}(y,\hat{y}) = -y \cdot \log \hat{y}. \tag{5}$$

$loss_{DNN}$ 為 cross-entropy 損失函數，如公式(5)，其中 y 及 $\hat{y}$ 分別為輸入資料的真實標籤(ground truth label)與 DNN 之輸出，藉由此公式計算 y 與 $\hat{y}$ 的差距以調整 DNN 中神經元的參數，使 DNN 的輸出結果愈接近真實標籤，增加分類的準確率。

3.3 攻擊彙整機制

若輸入資料的離群分數超過 TH 時，將被儲存到暫存器並等待攻擊彙整機制的啟動。當暫存器達到中具有一定數量的未知攻擊，則執行攻擊彙整機制以分離差異性較大的攻擊類型，作為後續模型更新的訓練資料。以下將針對此機制進行介紹。



圖四：攻擊彙整機制之流程圖

攻擊彙整機制流程如圖 4 所示，經由 DBSCAN 將零時差攻擊資料分群後，接著執行投票制度以矯正分群的結果。在投票階段，我們透過試著搬移資料點的群集，並讓該資料點的鄰居進行投票以判斷該次遷移是否恰當，藉此調整資料點至更適合的群集。在此階段，每個點被賦予一個值，名為契合度(matching value)。我們藉由資料點與其鄰居

計算出契合度，當資料點與越多鄰居處於相同群集時，此值越高，反之亦然。因此只要有任一點更動群集，每個資料點的契合度皆有可能隨之改變，故此值可作為在變換某資料點的群集時，每個點對於該次遷移的認同度。契合度的計算方式如下：以 p 表示一資料點，距離 p 最近的 k 個資料點之集合為 $N_p=\{n_p^1, n_p^2, \ldots, n_p^k\}$，$N_p$ 內的點稱為 p 的鄰居；N_p 內與 p 屬於相同群集的鄰居集合為 $SN_p=\{sn_p^1, sn_p^2, \ldots, sn_p^r\}$，$r \in [0, k]$且 SN_p 為 N_p 的子集，並以 $dist(p, q)$表示 p, q 兩點間的歐基里德距離。利用公式(6)-(7)總和資料點與其鄰居的距離倒數以計算出區域密度 $density_p$，其中僅計算與 p 距離小於 Eps 的鄰居之距離，當與自身鄰居的距離越小，該點的區域密度越大。接著，透過公式(8)計算出 p 與 N_p 的融入程度 iv_p(integrating value)，若 SN_p 內的元素越多，則 iv_p 越大。此外，我們使用兩點之間的距離倒數作為權重，當鄰居與資料點的距離愈遠，其影響程度愈小，最後藉由公式(9)計算出該資料點的契合度 mv_p。

$$w(p,q)=\begin{cases}1, & dist(p,q)<\text{Eps}\\ 0, & dist(p,q)\geq\text{Eps}\end{cases}. \tag{6}$$

$$density_p=\sum_{i=1}^{k} w(p,n_p^i)\cdot\frac{1}{dist(p,n_p^i)}. \tag{7}$$

$$iv_p=\sum_{i=1}^{r} w(p,sn_p^i)\cdot\frac{1}{dist(p,sn_p^i)}. \tag{8}$$

$$mv_p=\frac{iv_p}{density_p}. \tag{9}$$

當計算出每個點的 mv 時，便可透過調整資料點所屬的群集，並觀察其餘資料點的 mv 之變化，當整體 mv 提高則表示其周遭的資料點較偏好此次遷移後的結果；反之則表示多數資料點認為先前的狀態較佳，藉以判斷該資料點是否適合調整群集。首先計算 $N_{Eps}(p)$內所有點的 mv 之總和mv_{total}^{p}，隨後將目標資料點移至最多鄰居的群集，再計算一次總和$mv_{total}^{p'}$，接著比較mv_{total}^{p}與$mv_{total}^{p'}$，若後者較大代表該資料點處於新群集時整體契合度較高，因此較適合進行該次移動；反之則代表資料點不適合新群集，因此使其返回至原本的群集。重複如此動作，直到所有資料點不再變動，完成所有點的微調，提升分群的效果，其詳細流程如下所示：

- Step 1: 每個群集皆被賦予一個計數器，以 $counter_i$ 表示群集 C_i 之計數器，其初始值為 C_i 內資料點的數量。
- Step 2: 任意挑選一資料點 p，其群集為 C_i，並找出另一群集 C_j，為所有群中擁有最多 p 的鄰居之群集。
- Step 3: 計算所有資料點的契合度並加總，以mv_{total}^{p}表示，接著將點 p 移至 C_j，並重新計算所有契合度之總和$mv_{total}^{p'}$。
- Step 4: 比較mv_{total}^{p}及$mv_{total}^{p'}$，若mv_{total}^{p}較大則 p 返回至 C_i，並且將 $counter_i$ 減一；反之，若mv_{total}^{p}較小則p被分為 C_j，並將 $counter_j$ 重置為 C_j 之資料數。
- Step 5: 若所有群集的計數器皆為 0 時，則結束演算法，否則重複 Step 2 至 Step 4，

直到所有點皆不再更動。

肆、實驗結果與分析

在實驗階段我們使用 Python 程式語言，並且使用 Keras[17]與 Scikit-learn[24]進行系統的實作，實驗環境如下：64-bit Windows 10、Intel i5-9400 2.90GHZ CPU、16 GB RAM 以及 NVIDIA RTX 2060 GPU。在此章節中，將介紹本論文中所使用的資料集以及預處理的方法，並評估 AutoEncoder 用於零時差攻擊偵測之效益。此外，亦進行分類準確率之比較，最後評估本論文所提出的 DBSCAN 加入社群投票制度 DBSCAN-V 之分群效能。

4.1 資料集與前處理

本研究實驗所使用的資料集為 NSL-KDD[19]，是入侵偵測系統領域中經常被使用的資料集，其改善 KDD Cup '99[13]中的缺陷，如各類型資料數量與現實不符或資料重複等問題，因此 NSL-KDD 作為許多研究的基準資料集。在此資料集中可分為五大類別，分別為 Normal、DoS、U2R、R2L 以及 Probe，並且可細分為正常流量以及 22 種不同的攻擊，其類型與數量如表 1 所示。為進行未知攻擊偵測之實驗，我們將訓練資料中訓練集內樣本數量少於 20 的類型作為未知類型，即表 1 中灰底部分，其餘則作為已知攻擊。在 NSL-KDD 中，含有訓練資料以及測試資料，為建構 AutoEncoder 及 DNN，我們將前者拆分成比例為 3:1 的訓練集與驗證集，前者用於調整神經網路中的參數；後者則用於防止模型出現過度擬合的問題。模型訓練結束後，則使用測試資料作為模型的評估樣本，計算準確率以評估訓練的成效。

在 NSL-KDD 資料集中，每一筆資料皆具有 41 項特徵，且可分為連續值以及分類值特徵，屬於連續值的特徵如封包大小、連線時間等；分類值則有通訊協定類型、連線狀態等。在前處理階段，我們針對連續值特徵進行標準化(z-score)，使範圍限制為[0, 1]之間，以避免過度擬合的發生。

表一：NSL-KDD 中各類型資料於訓練集與測試集的數量

類別	攻擊類型	數量	
		訓練集	測試集
Normal	normal	67343	9711
DoS	back	956	359
	land	18	7
	neptune	41214	4657
	pod	201	41
	smurf	2646	665
	teardrop	892	12
U2R	loadmodule	9	2
	buffer_overflow	30	20
	rootkit	10	13
	perl	3	2
R2L	ftp_write	8	3
	guess_password	53	1231
	imap	11	1
	multihop	7	18
	phf	4	2
	spy	2	0
	warezclient	890	0
	warezmaster	20	944
Probe	ipsweep	3599	141
	nmap	1493	73
	portsweep	2931	157
	satan	3633	735

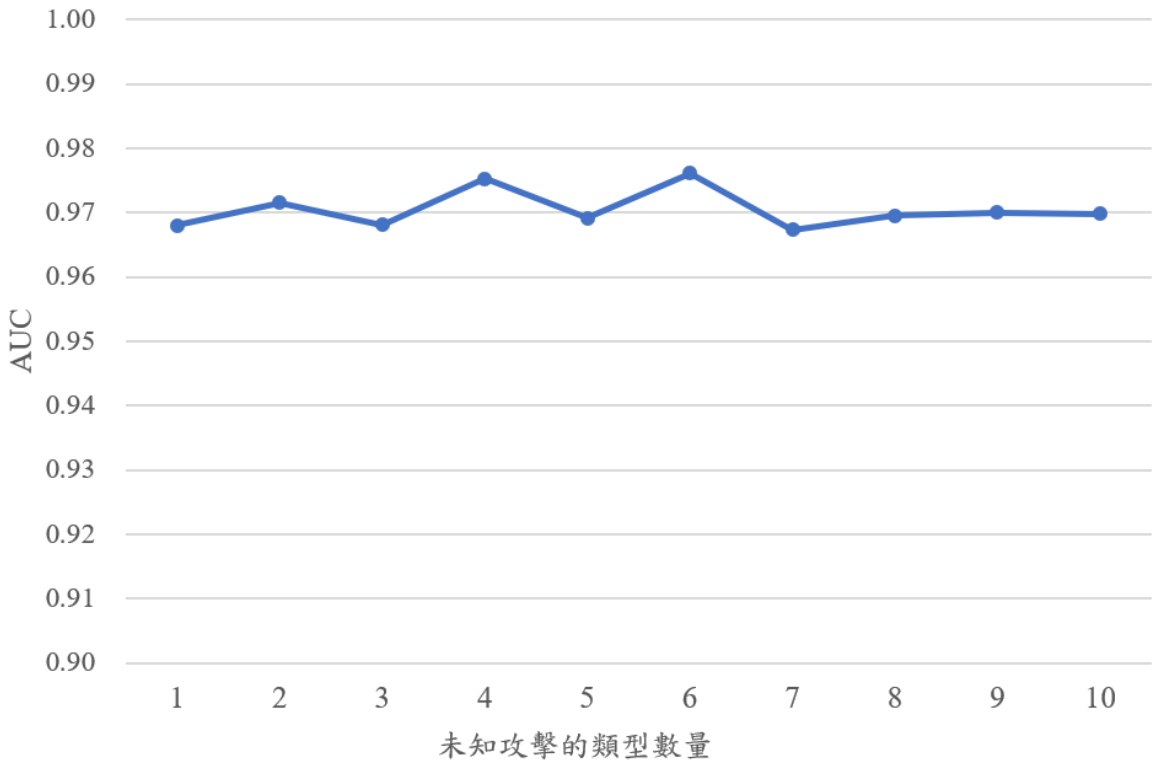
4.2 AutoEncoder 進行零時差攻擊偵測之效益

在本節針對未知攻擊類型的數量對於模型的影響進行相關實驗，我們透過逐漸增加已知攻擊類型的數量，模擬在不同情況下，對於未知攻擊的偵測能力的穩定性。表 2 為 AutoEncoder 之架構，輸入層與輸出層的神經元數目參照資料集的特徵數，因此設定為 41；而隱藏層則根據交互驗證所得。在此我們使用 Area under the Curve of ROC(AUC)[10] 評估模型偵測能力，AUC∈ [0,1]，當 AUC 的值越接近 1 表示模型辨別已知與未知類型的能力越強，可以從圖 5 發現，在不同的類型數量下，其 AUC 值皆維持於 0.95 以上，

表示 Autoencoder 在不同情況下皆擁有一定的偵測水準。

表二、 AutoEncoder 各層的神經元數量

Layer	Neurons
Input layer	41
Hidden layer	20
Output layer	41



圖五：AutoEncoder 對於偵測不同數量的未知攻擊類型之效益比較

4.3 已知類型之分類效益

我們比較 DNN 與 logistic regression[21]、decision tree[22]、random forest[3]以及 SVM[5]對於網路攻擊分類的準確率，DNN 的架構如表 3 所示，輸入層的神經元數量為資料的特徵數，在 NSL-KDD 中，每一筆資料皆有 41 種特徵；輸出層的神經元則與類型數量相同，在本實驗中已知類型為 13 種，因此我們將輸出層的神經元數目設定為 13；而隱藏層的神經元數量則根據 Heaton 等人於[12]中所提出的經驗法則，設定為輸出層與輸入層之神經元總和的三分之二。實驗結果如表 4 所示，其中顯示本篇論文所使用的

DNN 由於可透過隱藏層來學習提取特徵之能力，因此在分類方面相較於無法提取深層特徵的 logistic regression、decision tree、random forest 以及 SVM，可得到更加的結果。

表三：不同模型的分類準確率比較

Layer	Neurons
Input layer	41
Hidden layer	27
Output layer	13

表四：不同模型的分類準確率比較

Metrics / Model	Accuracy(%)
DNN(ours)	**89.94**
Logistic Regression	88.57
Decision Tree	88.53
Random Forest	88.41
SVM	87.76

4.4 加入社群投票制度之分群效益

在眾多分群法中，affinity propagation(AP)[8]是經典的演算法之一，因此我們將其加入實驗進行測試。透過替換分群演算法以及使用投票制度，針對不同的標籤數量進行投票制度的效能評估。此實驗使用 F-measure[7]做為標準，F-measure 為 Precision 以及 Recall 之調和平均數，其值介於 1 到 0 之間，越接近 1 表示分群結果越好。一個群的 Precision 與 Recall 分別可由公式(10)-(11)計算，其中 $C=\{C_1, C_2, \ldots, C_n\}$ 為 n 個群集的集合，$|C_i|$ 表示 C_i 內的資料數量，而 V^i 代表 C_i 中數量最多的類型，V^i_{sum} 為 C_i 中 V^i 的數量，V^i_{total} 則表示 V^i 所有的數量。計算出 C_i 的 Precision 以及 Recall 後便可透過公式(12)計算該群的 $F\text{-}measure_{Ci}$，並藉由公式(13)計算出該次分群結果的整體平均 $F\text{-}measure_{avg}$。

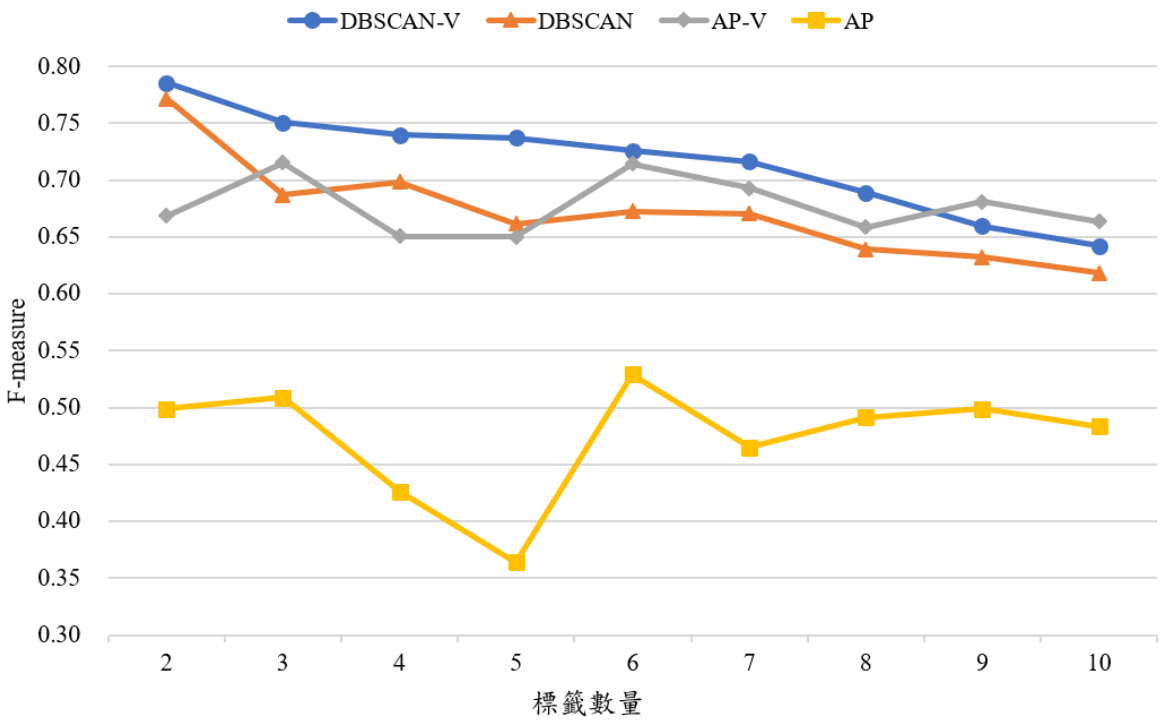
$$precision_{C_i} = \frac{V^i_{sum}}{|C_i|}. \tag{10}$$

$$recall_{C_i} = \frac{V^i_{sum}}{V^i_{total}}. \tag{11}$$

$$F\text{-}measure_{C_i} = 2 \cdot \frac{precision_{C_i} \cdot recall_{C_i}}{precision_{C_i} + recall_{C_i}}. \tag{12}$$

$$F\text{-}measure_{avg} = \frac{1}{n}\sum_{i=1}^{n} F\text{-}measure_{C_i}. \tag{13}$$

我們依據經驗法則[11]將每個資料點的鄰居數 k 設定為總資料量之平方根，DBSCAN 中 *Eps* 的值設定為所有點與其鄰居的平均距離。由於加入社群投票制度的分群演算法(DBSCAN-V 及 AP-V)會考慮多數資料點的契合度，以決定目標資料點所屬群集，因此群集內的資料點會更加地密集，並得到較高的分群效益。圖 6 為實驗結果，可從中證實無論是 DBSCAN 或是 AP，當加入社群投票制度後，效果皆有明顯的提升，並且在多數的情況下，DBSCAN-V 所獲得的 F-measure 皆為最高。



圖六：DBSCAN 與 AP 以及加入社群投票制度之效益比較

伍、結論

在此篇論文中，我們提出可偵測未知攻擊及彙整新類型入侵偵測系統之架構，當中使用 DNN 作為已知攻擊類型的分類器，並利用 AutoEncoder 輸出重構誤差，以偵測零時差攻擊。此模型可準確地分類攻擊類型，且擁有偵測未知攻擊之能力，在零時差攻擊猖獗的網路趨勢，此研究能夠應用於抵禦網路攻擊以減少其造成的損害。此外，藉由實驗證明，我們提出基於社群投票制度的攻擊彙整機制能有效地將不同類型的零時差攻擊分離，並聚合相似的攻擊。倘若模型須進行更新，便可透過此機制產生的結果進行重新訓練，以適應快速變化的網路環境。

參考文獻

[1] A. Abusitta, M. Bellaiche, M. Dagenais and T. Halabi, "A deep learning approach for proactive multi-cloud cooperative intrusion detection system," *Future Generation Computer Systems*, vol. 98, pp. 308-318, Sep. 2019.

[2] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, A. A. Alhasanat, "Solving the problem of the k parameter in the KNN classifier using an ensemble learning approach," *International Journal of Computer Science and Information Security*, vol. 12, no. 8, pp. 33–39, Aug. 2014.

[3] A. Sharma, I. Manzoor and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert Systems with Applications*, vol. 88, pp. 249-257, Dec. 2017.

[4] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972-976, Feb. 2007.

[5] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[6] C. Fahy, S.X. Yang and M. Gongora, "Ant colony stream clustering: a fast density clustering algorithm for dynamic data streams," *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2215-2228, Jun. 2019.

[7] C.Y. J. Peng, K. L. Lee and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The Journal of Educational Research*, vol. 96, no.1, pp. 3-14, Sep. 2002.

[8] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation," *in Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp. 318-362, 1986.

[9] E. Kabira, J.K. Hu, H.Wang and G.P. Zhuo, "A novel statistical technique for intrusion detection systems," *Future Generation Computer Systems*, vol. 79, pp. 303-318, Feb. 2018.

[10] J. Heaton, "Deep learning and neural networks," *in Artificial Intelligence for Humans*, vol. 3, 2015.

[11] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[12] J.A. Hanley and B.J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, pp. 29-36, 1982.

[13] J.D. Wang and H.C. Liu, "An approach to evaluate the fitness of one class structure via dynamic centroids," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13764-13772, Oct. 2011.

[14] J.W. Zhang, Y. Ling, X.B. Fu, X.K. Yang, G. Xiong and R. Zhang, "Model of the intrusion detection system based on the integration of spatial-temporal features," *Computers and Security*, vol. 89, Feb. 2020.

[15] Keras, Keras: The Python deep learning library, https://keras.io/.

[16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[17] M. Al-Qatf, L.S. Yu, M. Al-Habib and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843-52856, Sep. 2018.

[18] M. Ester, H.-P. Kriegel, J. Sander and X.W. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226-231, 1996.

[19] MixMode, What are zero-day attacks? and how AI is being used to combat them, https://mixmode.ai/blog/what-are-zero-day-attacks-and-how-ai-is-being-used-to-combat-them/.

[20] N. Chaabouni, M. Mosbah and A. Zemmari, C. Sauvignac, P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671-2701, thirdquarter 2019.

[21] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *in Proceedings of Military Communications and Information Systems Conference*, pp. 1-6, nov. 2015.

[22] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, Feb. 2018.

[23] S. Hettich and S. Bay, KDD cup 1999 data—the UCI KDD archive, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[25] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107-116, Apr. 1998.

[26] Scikit-learn, Machine learning in Python, https://scikit-learn.org/stable/.

[27] Statista, IC3: total damage caused by reported cyber crime 2001-2018, https://www.statista.com/statistics/267132/total-damage-caused-by-by-cyber-crime-in-the-us/.

[28] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37-50, May 2018.

[29] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *in Proceedings of the 27th International Conference on Machine Learning*, pp. 21-24, Jun. 2010.

[30] Votiro, The four zero day attack stats and trends you need to know, https://votiro.com/2018-the-four-zero-day-attack-stats-and-trends-you-need-to-know/.

[作者簡介]

陳志龍：現正就讀逢甲大學資訊工程學系碩士班，研究領域包含網路安全與機器學習。

魏國瑞：於 2016 年取得逢甲大學資訊工程學系博士學位，現於三甲科技股份有限公司擔任營運總監，研究領域包含網路安全、電子商務以及影像處理。

陳映親：於 2018 年取得逢甲大學資訊工程學系碩士學位，研究領域包含網路安全、電子商務以及影像處理。

李榮三：於 2017 年在逢甲大學資訊工程學系擔任教授，研究領域包含網路安全、區塊鏈應用以及影像處理。