航空技術學院學報 第十九卷　第十九期　第 22 －26 頁(民國 109 年)
Journal of Air Force Institute of Technology, Vol. 19, pp. 22-26, 2020

使用深度學習技術進行惡意程式分類
Malware Classification Using Deep Learning

[1]劉大哲 [2]劉奕賢 [3]蔡一郎 [4]李竹芬 [5*]李忠憲
[1235]國立成功大學電機工程學系/電腦與通信工程研究所
[4]國立虎尾科技大學財務金融系
{[1]tcliu, [2]ihliu} @cans.ee.ncku.edu.tw, [3]yilang@nchc.narl.org.tw, [4]chufenli@gmail.com,
[5]jsli@mail.ncku.edu.tw

摘要

由於近年來因為電腦和網路的快速發展、普及與應用，使得人們的生活愈來愈便利，卻也使得資訊科技犯罪的快速崛起，層出不窮，因此資訊安全已經變成一個非常重要的議題。然而面對越來越多變的惡意程式，使用傳統資料庫比對的方法可能會因為惡意程式做了加殼等等的動作而使特徵碼分析失去準確性，因此大家開始藉由機器學習與深度學習來對惡意程式進行分類，尤其是應用在大量分類中。而本論文研究了編碼部份以及多種分類的方法，並將彼此結合以提出一個有別於其他研究的惡意程式分類架構。
關鍵詞：惡意程式分類、神經網路、深度學習

Abstract

In recent years, due to the rapid development, popularization and application of computers and networks, people's lives have become more and more convenient, but they have also led to the rapid rise of information technology crimes. Therefore, information security has become a very important issue. However, in the face of more and more malware programs, the traditional database comparison method may cause the signature analysis to lose accuracy due to the actions of the packed-malware program, so everyone starts to learn by machine learning and Deep learning to classify malware, especially in a large number of categories. This paper studies the coding part and many kinds of classification methods, and combines them to propose a malware classification framework different from other research.
Keywords: Malware classification、dynamic analysis、Deep Learning

1. 緒論

在這個資訊爆炸且科技迅速成長的時代，網路已經成為了每天生活的必需品，每個人都可以透過網路輕易取得想要的資訊，而人們在享受網路普及帶來的便利性同時，隱藏的資安危機也暗中擴大。國際電信聯盟(ITU)在 2014 年就有調查指出，全世界使用網路的人口為 29 億人，這麼龐大的數字裡，自然會有心懷不軌的人想要使用非法的手段從中獲取利益、又或者是為了滿足個人慾望而進行破壞，其中最大部份便是透過惡意程式來達到其目的的。

惡意程式只不過是一個統稱，它包含了病毒、蠕蟲以及木馬等等的惡意軟體，而傳統使用資料庫比對特徵碼已經無法應對現今的惡意程式。其一原因為惡意程式的數量實在是增加的太快，根據卡巴斯基實驗室在 2017 年的報告指出[1]，每天大約有 36 萬個新惡意程式出現。其二為許多惡意程式製作者在製作惡意程式時，會使用加殼等技術來防止防毒軟體利用特徵碼分析來進行偵測，甚至是從現有的惡意程式中透過自動化的工具大量改寫而來，經過改寫而大量產生的惡意程式同時也會造成特徵資料庫越來越大。結合上述兩種原因，傳統資料庫比對的方法已無法應對現在的趨勢，也因此有許多研究開始轉向人工智慧著手，利用機器學習與深度學習的方法對惡意程式進行分類，並且設法提高準確率。

在論文第二章節的部份是文獻探討，此章節會介紹一些惡意程式分析相關的研究，並且比較使用的方法。第三章節透過前面學者所整理的文獻中，本研究將試著提出一個新的惡意程式分類架構作為探討並在最後一個章節中根據新架構與其他學者的方法做一個討論與結論。

2. 文獻探討

此章節會先簡單探討靜態分析與動態分析之差異，然後介紹一些常用的機器學習與深度學習演算法

2.1 惡意程式分析

惡意程式分析方法分為靜態分析與動態分析。簡述如下：

靜態分析：顧名思義就是當你在分析時並不會啟動惡意程式的執行檔，而是將執行檔的程式碼進行分析 ，正因為是針對程式碼進行分析所以不會有受到感染的風險。但往往進行靜態分析需要對執

行檔進行逆向工程反組譯，甚至有些加殼保護的惡意程式還須先經過特定工具脫殼才能進行逆向工程，因此取得過程較為複雜。

動態分析：由於需要執行惡意程式，所以會在虛擬環境下執行，並且透過動態分析工具紀錄程式在執行中的惡意行為，像是存取 File、DLLs、Registry 與 API 函數的呼叫等等..。而動態分析面對有加殼保護的惡意程式時顯得十分有用，其原因為惡意程式在執行時會自動解密然後執行程序。但動態分析也有他的不足，像是反虛擬機器監控。

從以前到現在，都有學者分別使用動態分析和靜態分析去對惡意程式做分析，如表 1 所示，不管是使用哪一種方法不外乎都是將特徵取出後交給機器學習或深度學習去訓練，其中 Kolosnjali 等人[4]提出了一個概念，既然取出的特徵有很多種，那將不同的特徵使用不同的模型分開訓練，會比一起丟入同一種模型訓練來得好。

表 1 相關研究分析方法

相關研究	靜態分析	動態分析
Ye et al. (2009)[2]	字串	無
Lin et al. (2015)[3]	無	Api calls
Kolosnjali et al. (2017)[4]	PE files and Opcodes	無
Kalash et al. (2018)[5]	Malware binary file to grayscale image	無

2.2 機器學習與深度學習

以往有許多研究使用機器學習與深度學習來為惡意程式做分類，以下我會簡單介紹一些常用到的方法。

2.2.1 支持向量機(Support Vector Machine)

SVM 最簡單的概念，就是希望能在不同類別資料集中，找到一個最佳的超平面(hyper plane)將不同類別的資料分開來。而 SVM 基本上是一個二位元分類器(binary classifier)，利用支援向量來算出圖 1 中的橘色實線，我們稱之為分類線，用來分類資料。但是現實生活中所遇到的分類問題大部分是屬於多類別的，因此 SVM 也能應用在多類別的分類上[2][6][7][8]，像是一對多(one-against-all)和一對一(one-against-one)的方法。

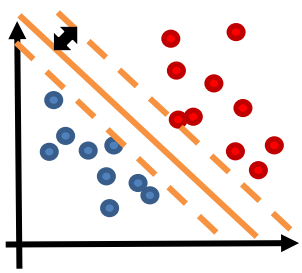


圖 1 SVM 之示意圖

2.2.2 多層感知器(Multi-layer Perceptron)

MLP[9]由感知器推廣而來的，主要就是因為含有多個神經元層，因此也叫深度神經網路(Deep Neural Networks)，是一種非常基礎且簡單的深度學習的神經網路，在神經網路裡每一層都與下一層完全連接。MLP 由三個層面所組成，分別為輸入層、隱藏層以及輸出層，中間可以有多個隱藏層，而最簡單的 MLP 就像圖 2，輸入層與輸出層中間只包含一個隱藏層，即三層結構。

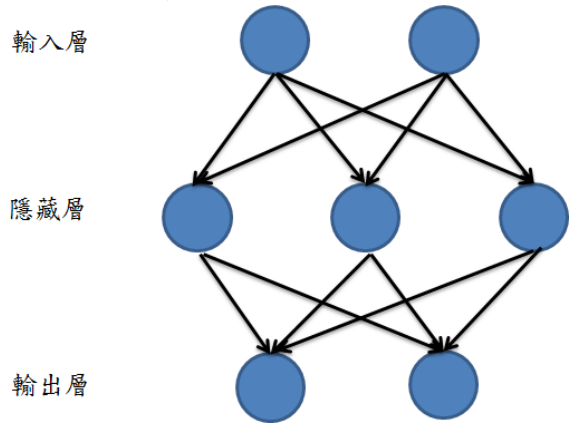


圖 2 MLP 之示意圖

2.2.3 卷積神經網路 CNN(Convolution Neural Network)

卷積神經網路的優勢是處理多陣列型態表達的資料，像是 RGB 三通道彩色的圖片。而 CNN 與一般神經網路的差別為 CNN 是對原的始圖像直接進行動作，然而一般神經網路是先對影像取出特徵(例如:灰階化)才做操作。CNN 主要由以下三個部分組成，分別是卷積層、池化層以及完全連結層。

以往有許多研究都是使用 CNN 來對惡意程式作分類[4][5][10]，像是將惡意程式可視化為圖像的應用，就是利用了 CNN 對圖片的優勢。首先會先讀取惡意程式二進制檔案中的前 8 位數值，接者將此向量的值由二進制轉換為十進制值，再將這些十進位的值透過灰階顏色與數值轉換表，進而轉換成可視化的灰階圖象，流程如圖三所示，最後透過 CNN 來分析灰階圖象，進而分類出惡意程式。

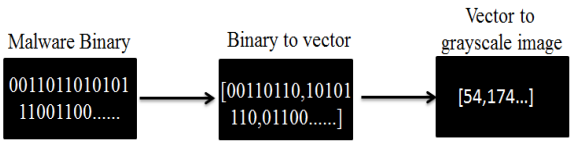


圖 3 惡意程式轉灰階圖象之流程圖

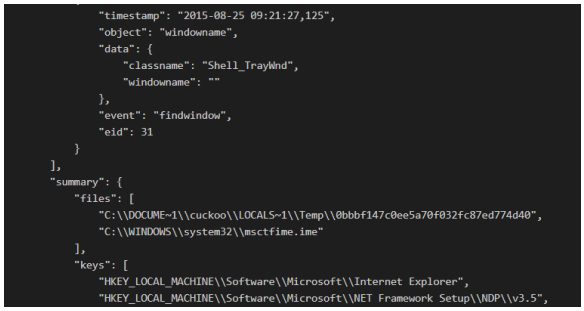
3. 研究方法

根據第二章節所提到的，Kolosnjali 等人[4]將不同特徵使用不同模型分開訓練得到較好的準確率。藉由這個概念我們也想到不同的特徵或許會有

各自適合的編碼方式,因此若是找到各自適合的編碼方式並搭配分開訓練,一定能將準確率再提高。此章節會先陳述我們的資料來源、如何編碼,並且介紹為何使用融合多層感知器當作惡意程式分類架構。

3.1 資料來源

資料是由南科國家高速網路與計算中心所提供的[11]，而這些資料為大量惡意程式在 cuckoo sandbox 沙箱運行後得到的行為日誌。利用這些資料裡面紀錄的 Files、DLLs 以及 API 函數呼叫等等，當作惡意程式分類的特徵。



```
            "timestamp": "2015-08-25 09:21:27,125",
            "object": "windowname",
            "data": {
                "classname": "Shell_TrayWnd",
                "windowname": ""
            },
            "event": "findwindow",
            "eid": 31
        }
    ],
    "summary": {
        "files": [
            "C:\\DOCUME~1\\cuckoo\\LOCALS~1\\Temp\\0bbbf147c0ee5a70f032fc87ed774d40",
            "C:\\WINDOWS\\system32\\msctfime.ime"
        ],
        "keys": [
            "HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Internet Explorer",
            "HKEY_LOCAL_MACHINE\\Software\\Microsoft\\NET Framework Setup\\NDP\\v3.5",
```


圖 4 惡意程式經 cuckoo sandbox 沙箱運行後得到的行為報告

3.2 編碼

當我們拿到資料後除了選取我們所要的特徵外，最重要的就是編碼，如此一來才能將這些特徵當作機器學習或深度學習的輸入，以下介紹三種編碼方式：

3.2.1 獨熱編碼(One-Hot Encoding)

在數據處理和特徵工程中，經常會遇到像是 Male 、 Female 這類無序型特徵，因此需要找一個方法讓這兩個屬性距離原點是相同距離，而 One-hot encoding 就是解決這的問題的方法。首先會將 Male, Female 從一個欄位拆成兩個欄位，因此 Male 對應到的編碼資料就是(1, 0)，Female 對應到的編碼資料就是(0, 1) 這兩個使用者對於原點的距離都是 1，就達成我們想要的結果了。

假設一個特徵若有 a 個值的話，使用獨熱編碼就會變成了 a 個 2 元的特徵。這些特徵彼此互斥，每次只會有一個被激活，因此特徵的值越多，數據也會變得更稀疏，但在一定程度上也擴充了特徵的數量。

3.2.2 詞集模型(Set-of-words model)與詞袋模型(Bag-of-words model)

詞袋模型與詞集模型最初被用在語言處理以及信息檢索領域，而近年來在圖像領域和語音識別領域也漸漸使用此方法。最近詞袋模型也被用在文件的分類方面，通常會將詞出現的頻率作為特徵訓練分類器。

詞袋模型與詞集模型能夠把一個句子轉化為向量表示，是比較簡單的一種方法，它不考慮句子中單詞的順序，只考慮詞表（vocabulary）中單詞在這個句子中的出現次數。不同的是詞集模型是單個文本中的單詞出現在字典中，不管出現多少次就將其設為 1。而詞袋模型是單個文本中的單詞出現在字典中，出現多少次就將其數值加 1。詞袋模型和詞集模型都是基於詞之間保持獨立性，沒有關聯爲前提。這使得其統計方便，但同時也丟失了文本間詞之間關係的信息。

3.2.3 TF-IDF(Term Frequency - Inverse Document Frequency)

TF-IDF 可以說是詞袋模型的延伸，它是一種加權技術常常用在文字探勘以及資訊檢索中，為一種統計方法。剛剛說過詞袋模型為統計單詞在單個文本中出現的頻率，而與之不同的是 TF-IDF 除了統計單詞在單個文本中出現的頻率外，還用來評估單詞對於單個文本中的重要程度。

而我們在 API 函數呼叫這類的特徵，正是運用了 TF-IDF 來對我們的特徵進行編碼。其原因在於每一隻惡意程式他所呼叫的 API 函數次數皆不盡相同，因此若是使用詞袋模型去統計頻率，會造成較不客觀的結果。因此我們使用 TF-IDF。

3.3 系統架構

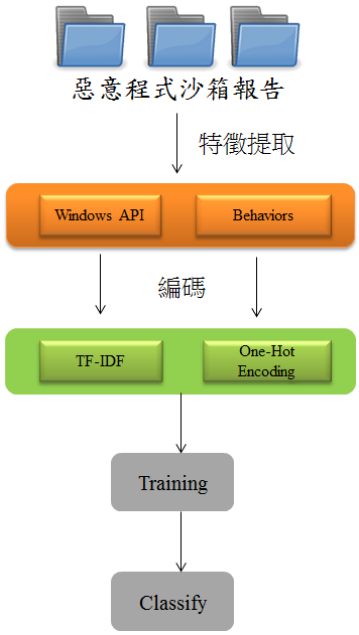


圖 5 惡意程式分類系統架構圖

我們使用的模型並非單純的多層感知器，我們從沙箱報告中選取的特徵不只 API 函數，還有 Files 和 Keys 等之類的行為特徵，這些特徵各自的型態不盡相同，我們認為將這些特徵一起編碼並使用多層感知器去訓練不是很適當。因此我們各自對特徵進行不同的編碼，我們對 API 函數使用 TF-IDF 編碼，而 Files、keys 這類的行為特徵則是使用獨熱編碼，並且將這兩種編碼後的結果分別丟入多層感知器去訓練，訓練完成後再將其合併起來，形成融合多層感知器。融合多層感知器的優勢在於可以併行處理資料，同時處理 TF-IDF 編碼與獨熱編碼的資料，並且具有相當強的自適應處理能力。我們模型除了能夠適應自變化的惡意程式，更能因應惡意程式各種行為增加分類的準確率。

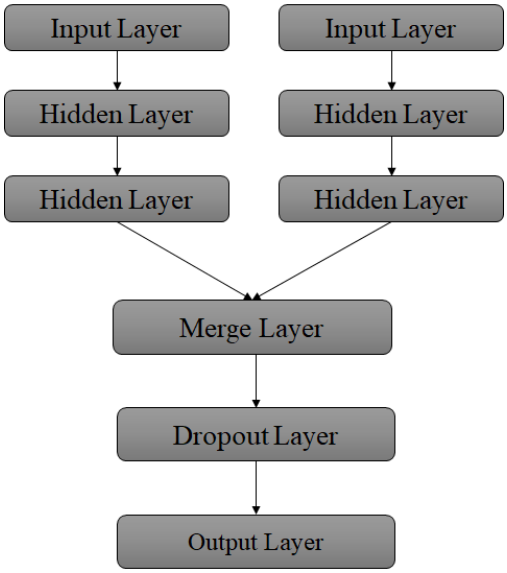


圖 6 融合多層感知器之示意圖

由於融合多層感知器為監督式學習，所以除了特徵外資料還必須具備標籤欄位，而標籤的部分我們根據沙箱報告裡 VirusTotal [12]的各家防毒軟體回饋的結果，如圖 7 所示，我們可以看出各家防毒軟體對該惡意程式的判斷，我們將這些結果做多數決的統計，用來生成資料集內的標籤欄位。藉由標籤欄位我們可以用來佐證我們模型的正確性。

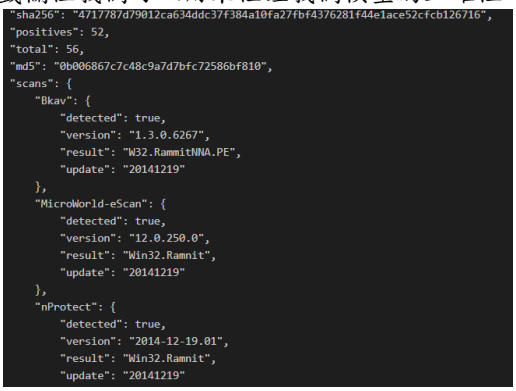


圖 7 各家防毒軟體的回饋結果

4. 結論

在目前的環境中資訊安全儼然已變成一個極為重要且嚴峻的議題，也正是因為資訊安全越來越重要，因此越來越多的惡意程式出現並增加極為快速，使得傳統特徵碼分析面臨窘境。因此本研究提出了一種方法希望使準確率提升。以往的研究不管動態分析或是靜態分析，皆是將特徵編碼後直接交由機器學習或深度學習去訓練，而本研究提出了希望藉由當不同特徵有各自適合的編碼方式時，能夠找到其適合的編碼方式並各自分開訓練，達到比我們使用同一種編碼方式，或是不同編碼方式但是一起訓練更好的準確率。

誌謝

感謝科技部計畫 MOST 107-2218-E-006-036-及 MOST 107-2221-E-006-140-提供經費支持本研究的進行。

參考文獻

[1]. KasperskyLab,https://www.kaspersky.com/about/press-releases/2017_kaspersky-lab-detects-360000-new-malicious-files-daily
[2]. Ye, Y.Y., Chen, L.F., Wang, D.D., Li, T., Jiang, Q.S., and Zhao, M.,SBMDS: an interpretable string based malware detection system using SVM ensemble with bagging,Journal in Computer Virology, Vol. 5, No.4, pp. 283-293, 2009.
[3]. Lin, C.-T.; Wang, N.-J.; Xiao, H.; and Eckert, C. 2015. Feature selection and extraction for malware classification. Journal of Information Science and Engineering 31(3):965–992.
[4]. B. Kolosnjaji, G. Eraisha, G. Webster, A. Zarras, C. Eckert, "Empowering convolutional networks for malware classification and analysis", Proc. Int. Joint Conf. Neural Netw. (IJCNN), pp. 3838-3845, 2017.
[5]. M. Kalash, M. Rochan, N. Mohammed, N. D. Bruce, Y. Wang, F. Iqbal, "Malware classification with deep convolutional neural networks", New Technologies Mobility and Security (NTMS) 2018 9th IFIP International Conference on, pp. 1-5, 2018.
[6]. T. Holz, C. Willems, K. Rieck, P. Duessel, and P. Laskov. Learning and Classification of Malware Behavior. In Fifth Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 08), June 2008.
[7]. C. Hsu and C. Lin, “A comparison of methods for multiclass support vector machines,” IEEE Transactions on Neural Networks, Vol. 13, 2002, pp. 274-282.
[8]. M. Kruczkowski, E.N. Szynkiewicz, "Support vector machine for malware analysis and classification", Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02, pp. 415-420, 2014.
[9]. F. Murtagh Multilayer perceptrons for classification and regression Neurocomputing, 2 (1991), pp. 183-197
[10]. L. Nataraj, S. Karthikeyan, G. Jacob, and B. Manjunath. Malware images: Visualization and autmoatic classification. In Proceedings of VizSec, 2011..
[11]. Malware database, https://owl.nchc.org.tw/
[12]. VirusTotal, [online] Available: http://www.virustotal.com.