Regular Paper

Communications of the CCISA
Vol. 28 No. 4 Nov. 2022

利用異常偵測技術於可疑帳號辨識之研究

陳彥翔[1]、林祝興[2*]、賴俊鳴[3]
[1,2,3]東海大學資訊工程學系
[1]zwasd5123@gmail.com、[2]chlin@thu.edu.tw、[3]cmlai@thu.edu.tw

摘要

近幾年來，「假新聞」、「假訊息」等威脅，在資訊戰中已達到國安等級，也成為了許多國家研究的重點。但此議題並非為新興現象，例如，早在 2014 年俄羅斯介入影響烏克蘭的克里米亞歸屬公投，以及最近的烏俄戰爭中，我們都可以看到不管是俄羅斯或是其餘國家，許多社群媒體帶風向的情況。因此，本論文專注於發布可疑訊息的帳號以及貼文，並利用 Twitter 官方的計畫網站 —「Transparency」網站中，Twitter 定義可疑帳號為跟政府或州有關的假訊息操弄帳號，公布經調查確認為可疑帳號以及貼文的資料。有別於以往的識別方式，我們利用機器學習中的「異常偵測」技術，訓練出一個能以高準度分辨出異常訊息以及異常帳號之分辨器。在資料收集方面，我們建立基於 ETL 框架的資料爬取系統，爬取了名人的官方帳號以及推文。並利用官方已經證實身分之有「藍勾勾」的帳號所發布之正常貼文，來驗證分辨器誤判之情形。從實驗結果，我們發現準確度達到 96%，獲得很好的效果。

關鍵詞：可疑帳號、假訊息、自然語言處理、機器學習、異常偵測、ETL、爬蟲

* 通訊作者 (Corresponding author.)

Regular Paper

Communications of the CCISA
Vol. 28 No. 4 Nov. 2022

On Identifying Suspicious Accounts Using Anomaly Detection Technology

[1]YAN-SIANG CHEN, [2]CHU-HSING LIN, [3]CHUN-MING LAI
[1,2,3]Department of Computer Science, Tunghai University
[1]zwasd5123@gmail.com, [2]chlin@thu.edu.tw, [3]cmlai@thu.edu.tw

Abstract

In recent years, threats such as "fake news" and "disinformation" have reached the level of national security in information warfare, and have become an important research issue. For example, as early as 2014, Russia intervened to influence Ukraine's Crimea referendum, and in the recent Ukrainian-Russian War, we can see that in many communities, whether Russia or the others, the media takes the wind. This article focuses on the accounts and posts that publish suspicious information, and uses Twitter's official project website — Transparency website. Twitter defines suspicious accounts as accounts that manipulate disinformation related to the government or state, and publishes them after investigation and confirmation. Different from the previous identification methods, in this paper we use the "anomaly detection" technology in machine learning to train a classifier that can distinguish abnormal messages and abnormal accounts with high accuracy. For the dataset, we established a data crawling system based on the ETL framework, and crawled official accounts and tweets of celebrities. And use the normal posts posted by the accounts with blue tick, whose identities have been officially confirmed, to verify the performance of the classifier. From the experimental results, we found that the accuracy of our identification method reached 96%.

Keywords: suspicious account, misinformation, natural language processing, machine learning, anomaly detection, ETL, crawler

壹、前言

1.1 研究背景

根據 Varieties of Democracy (V-Dem) 跨國調查計劃，將近 200 個國家的資料中，在「數位社會」單元，其中一項針對各國「遭受到外國假資訊攻擊程度」[1]。該報告顯示在 2018 年全球國家中遭受到外國假訊息攻擊最嚴重的國家就是「台灣」，而且遭受攻擊的目的大都基於特定的政治議題。

在 2018 年 Twitter 官方開始進行透明度報告 (Transparency Report)，每年都會將研究觀察到的可疑帳號以及其可疑帳號的推文資料公布。公布資料全都是可疑推文的帳號及其相關資料，此資料在 Twitter 研究可疑帳號的議題是一大助力。本篇論文利用 Twitter 官方公布之資料作為我們研究分辨器的資料。

在近期研究中，我們發現目前的研究轉向於帳號的行動所產生的特徵以及衍生出的圖譜作為方向[11] [12] [13]。但經過搜尋相關論文，我們並未發現利用「異常偵測」相關技術做為研究方向。因此，在本論文，我們擬定並嘗試利用該技術，探討是否能夠設計出可疑帳號以及訊息的分辨器。

1.2 研究目的

在相關的研究中，最知名的是俄羅斯網軍針對 2016 年美國總統大選所做的政治操弄，以及使用的俄羅斯「巨魔資料集」(Russian Troll dataset) [10]。我們研讀論文[2]，並在搜尋各文章後發現，在社群媒體中可疑帳號是少數，但是我們卻無法準確判斷是否為可疑帳號的情況下，很難直接利用爬蟲在社群媒體爬出足夠的資料集。

因此，我們找尋 Twitter 官方所發布的可疑帳號以及貼文。因為其中資料皆為可疑帳號以及可疑貼文，因此，在資料皆為同一類型時，我們想到異常偵測的方式。也許此方式可以訓練出一個分辨器，能明確的分辨出異常帳號，並在正常帳號出現時不被誤判。如此，就能完美的設計出可疑帳號以及可疑推文的分辨器。

貳、研究方法

2.1 異常偵測

異常偵測 (Anomaly Detection) [3]是機器學習演算法的一個常見應用，主要用於非監督式學習問題，其目的在於能在一群資料中分辨異常值/離群值 (Anomaly/Outlier) 的方法。此技術被應用在各個領域上，例如：金融業應用以辨識異常財務行為、資訊安全

用以辨識異常流量.....等等。以圖 1 的資料二維分布範例，是多筆資料在一個二維平面時分布狀況，此時異常偵測的目的在於「定義一樣以及不一樣」。

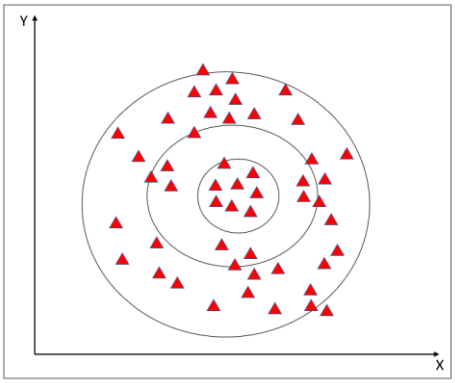

圖1： 資料二維分布範例

圖 1 中的圓圈也可以看出不同的界線以及分布，可以發現群聚的中心點以及離中心點較遠的點。我們可以將中心附近的界線畫出，界定出的界線將可以分類出正常以及異常值。如圖 2 所示。藍色區塊為聚集的正常值，而我們將可以調整參數等等界定異常的範圍遠近。例如圖中綠色為設定界定範圍後在界線內的點也為正常值，而粉紅色的在界線範圍外的為異常值，藉此來區分正常以及異常值。

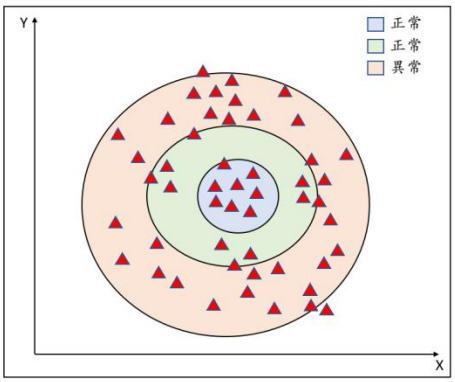

圖2： 二維資料劃分異常正常值範例

異常偵測領域的不同技術又分成兩類，我們將其整理成如圖 3 以方便解釋：

- Labeled：訓練集中每個資料都有標籤，可以給電腦一個正確的答案，在遇到未知時也可以輸出未知 (Unknown) 標籤，有未知標籤的功能稱之為開放式識別 (Open-set recognition)。
- Unlabeled：所有的資料都是沒有標籤的，電腦只能利用演算法去歸納出資料是否有一定規律，其中又分為兩種資料狀況：
  1. Clean：所有的資料都為同一種類，沒有任何的不同或是異常的樣本，所有資料都視為正樣本。

2. Polluted：在此領域異常樣本通常會極度不平衡，而且會有大機率出現異常資料沒有被標注出來的情況，資料及狀況會是一堆的正樣本裡面夾雜了少數的異常樣本，在此領域中異常樣本數通常為非常少。

因為我們所使用的 Twitter 官方公布之資料來做任務的選定以及訓練方式的調整，可以很明顯的發現我們所需要的是使用到異常偵測領域中 Unlabeled 的情況。且資料皆為同一種類，為 Clean 的資料情況。在非監督式異常偵測中共有三項代表技術：Auto-Encoder、One-Class SVM、以及 Isolation Forest，以下章節針對此三種技術做探討。

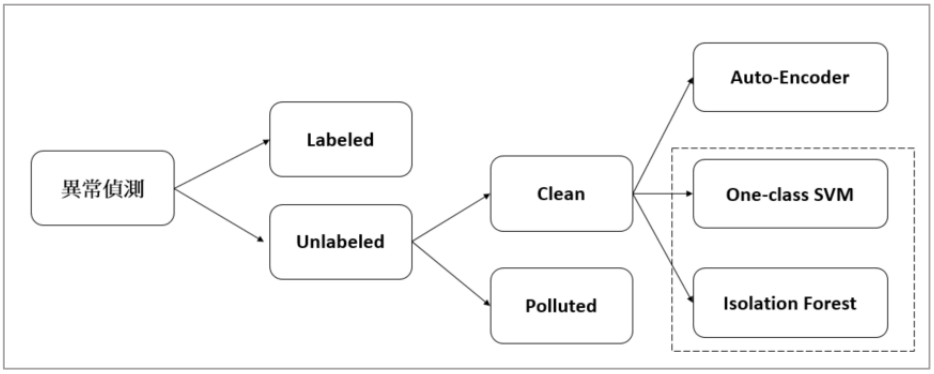

圖3： 異常偵測領域的不同技術分類圖

2.2 Auto-Encoder

其中 Auto-Encoder (自動編碼機) [4] 大多用於圖片領域，是深度學習中的一種非監督式學習演算法。其運作如圖 4 所示。在此深度學習結構上做了一個對稱，左邊為 Encoder，右邊為 Decoder，概念上有點類似於圖片的壓縮以及解壓縮的概念。左邊利用編碼器 (Encoder) 截取少量資訊，類似於特徵擷取，但右邊的解碼器 (Decoder) 又能利用這樣的少量資訊還原圖片，就可以藉此方式自動分辨是否為異常。但此技術大多用在圖片領域，本次實驗並不適合使用。

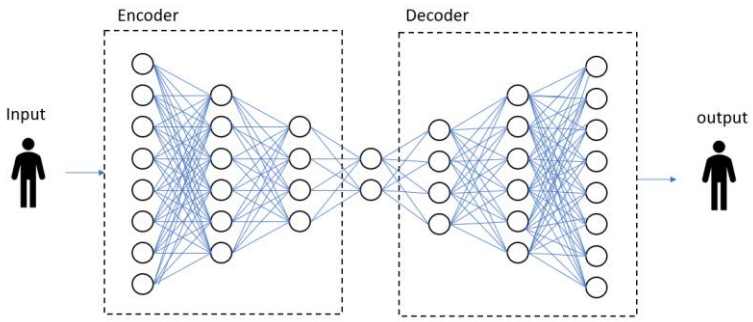

圖4： Auto-Encoder 示意圖

2.3 One-Class SVM

One-Class SVM [5] 顧名思義就是只有一個分類的分類器，利用統一的樣本的特徵去訓練出一個決策邊界。並利用此決策邊界將資料二分為與此樣本相同類或是不同類。在邊界內即為正常，在邊界外則為異常。其中演算法使用到的 Kernel 是 RBF (Gaussian Radial Basis Function)，是 SVM 裡面最常使用的函數，能將資料依據泰勒級數投射到更高維的空間並可能會有利於做切割。例如，圖 5 的例子就是投射到高維產生一個切面的示意圖。

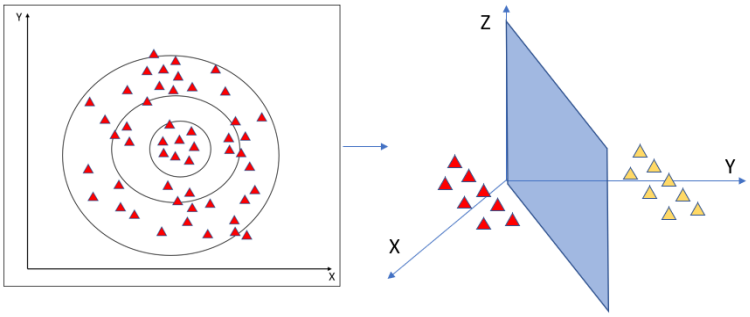

圖5： RBF 投射到高維做出決策平面

2.4 Isolation Forest

孤立森林 [6] 是一個非監督式的異常偵測演算法，具有線性時間複雜度以及高準確度。該演算法利用二元樹的建構方式對每組資料做隨機切分。因為異常點的數量較少且大多具有疏離性，所以異常點能在切分時快速被切割出來，如圖 6 所示，在每次的切割中可以分割成兩群，當數值較分離時就會很快就分割出來。Isolation Forest 比傳統方法，例如 K-means、Local Outlier Factor 等，對高維度資料具有較好的強韌性。孤立森林處理的步驟分兩個階段：

第一階段：

1. 從訓練資料中隨機挑選樣本點作為子集。
2. 隨機選定一個維度的特徵，並在當前節點資料中產生一個切割點，此切割點位於資料最大以及最小值之間。
3. 切割點的平面將把資料切分成兩個子空間。
4. 重複 2 跟 3 步驟，直到剩下一個資料。
5. 重複循環 1 到 4 的步驟直到所需孤立樹數量。

第二階段：

利用產生的這些樹，遍歷所有的樹，並計算在森林中的平均高度，最後分數越小表示資料越異常。

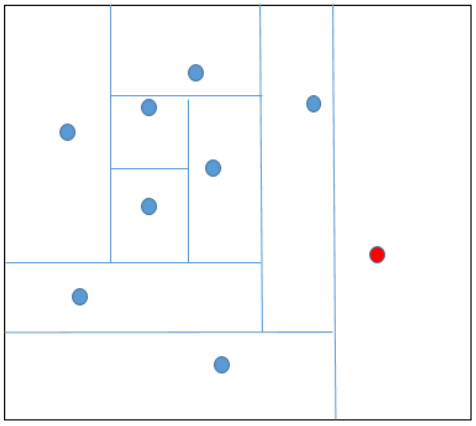

圖6： Isolation Forest 效果

2.5 ETL 架構系統

ETL 是當今因應大數據以及大量的 AI 系統下所衍伸出的架構，其架構分別有三種不同的部分，分別是：

1. 擷取（Extract）：從資料源將資料萃取出來，原封不動的寫入暫存區，存入資料庫做資料保存
2. 轉換（Transform）：資料清理，簡單來說只要是資料前處理都可以在這一塊，因為調用的是暫存區，因此使用上較為快速，處理完成後將其存入資料庫，注意要將和資料源分開不可搞混。
3. 載入（Load）：經過轉換後的資料通常會有很多的特徵可以使用，在做載入前應先設定好維度模型，維度模型簡單來說就是先將需要的特徵以及其空間先準備好，再到暫存區或資料庫做導入。

一套 ETL 系統就好比一間餐廳，需要幾個關鍵，分別是：

1. 一套好的菜單，在系統上等於好的 Input 設定。
2. 一份好的食材供應商，在系統上就是好的數據源。
3. 好的料理方式，在系統上就是好的資料前處理。

能先備好處理完的食材，並在不同的菜單來時做導入就是此架構的重點。

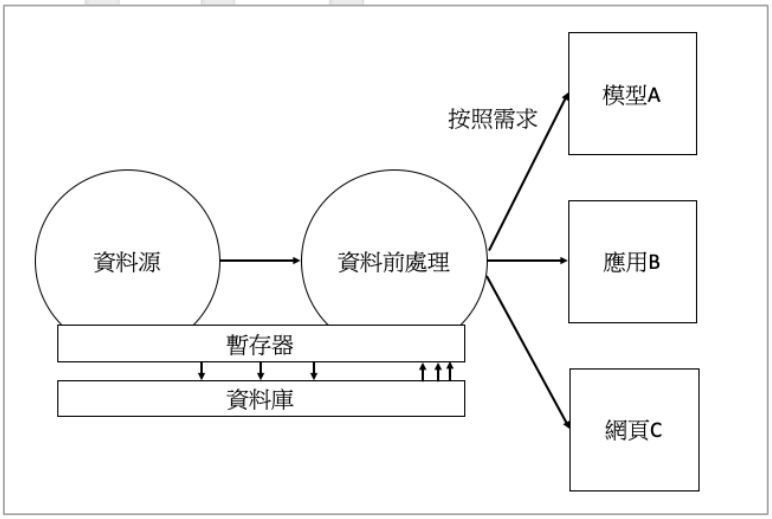

圖7： ETL 架構說明

2.6 資料介紹

大約每六個月，Twitter 就會發布一份透明度報告*，詳細說明它已經暫停了多少國家支持的帳號，以及政府要求提供帳號持有人資訊的數量。Twitter 公開了經過相關研究後確定為國家操縱訊息活動的帳號以及其推文。Twitter 相信能藉由此披露的訊息，將能使公眾以及研究界得到更好的研究資料以及對 Twitter 平台的透明度。從 2018 年 10 月開始推出了第一份外國訊息操作檔案，使研究人員能夠調查、學習和培養未來的媒體素養。本論文利用 2018 年 Twitter 所公布的資料檔案作為研究資料。

Twitter 所公布的資料檔案中涵蓋了三個大項，分別是帳號、推文以及媒體資料。本論文專注於推文資料上，其中包含的欄位如下(由 Twitter 官方提供) [7]：

1. tweetid - tweet identification number
2. userid - user identification number (anonymized for users which had fewer than 5,000 followers at the time of suspension)
3. user_display_name - the name of the user (same as userid for anonymized users)
4. user_screen_name - the Twitter handle of the user (same as userid for anonymized users)
5. user_reported_location - the user's self-reported location (*)
6. user_profile_description - the user's profile description (*)
7. user_profile_url - the user's profile URL (*)
8. follower_count - the number of accounts following the user (*)
9. following_count - the number of accounts followed by the user (*)
10. account_creation_date - date of user account creation

* https://transparency.twitter.com/

11. account_language - the language of the account, as chosen by the user
12. tweet_language - the language of the tweet
13. tweet_text - the text of the tweet (mentions of anonymized accounts have been replaced with anonymized userid)
14. tweet_time - the time when the tweet was published (UTC)
15. tweet_client_name - the name of the client app used to publish the tweet
16. in_reply_to_tweetid - the tweetid of the original tweet that this tweet is in reply to (for replies only)
17. in_reply_to_userid - the userid of the original tweet that this tweet is in reply to (for replies only)
18. quoted_tweet_tweetid - the tweetid of the original tweet that this tweet is quoting (for quotes only)
19. is_retweet - True/False, is this tweet a retweet
20. retweet_userid - for retweets, the userid who authored the original tweet
21. retweet_tweetid - for retweets, the tweetid of the original tweet
22. latitude - geo-located latitude, if available
23. longitude - geo-located longitude, if available
24. quote_count - the number of tweets quoting this tweet
25. reply_count - the number of tweets replying to this tweet
26. like_count - the number of likes that this tweet received (^)
27. retweet_count - the number of retweets that this tweet received (^)
28. hashtags - a list of hashtags used in this tweet
29. urls - a list of urls used in this tweet
30. user_mentions - a list of userids who are mentioned in this tweet (includes anonymized userids)
31. poll_choices - if a tweet included a poll, this field displays the poll choices separated by

(*)代表帳號被 Twitter 官方停權時收集資料

(^)代表不涵蓋當發布此資料時已經被暫停刪除或已經採取其他行動的用戶參與行動

2.7 資料前處理

Twitter 的資料集共有 8,768,633 筆資料，在此研究中，為了效率我們先隨機選取其中 10,000 筆資料。並且在資料前處理，我們參考 [11][16] 的研究後並加以改善，總共處理以下幾項特徵。註：有特徵生成也有轉換。

1. anonymized 匿名性：利用 tweetid, userid, user_display_name 跟 user_screen_name 製作匿名性，代表好友數量小於 5000 人。

2. profile_description_lenghth：自我介紹的長度。
3. follower_following_ratio
4. user_profile_url：frequency encoding
5. time_stamp：轉成 time_stamp 利用時間戳記轉換。
6. account_language：frequency encoding
7. tweet_language：frequency encoding
8. tweet_time：轉成 time_stamp 利用時間戳記轉換。
9. tweet_client_name:frequency encoding
10. in_reply_to_tweetid：二值化
11. quoted_tweet_tweetid：二值化
12. is_retweet：二值化
13. latitude：二值化
14. longitude：二值化
15. quote_count:數值化
16. reply_count:數值化
17. retweet_count:數值化
18. hashtags：數值化+ frequency encoding
19. urls：數值化+ frequency encoding

對每個欄位經特徵工程處理後，總共有 33 個欄位，此時針對欄位間的相關性做皮爾森相關分析(Pearson Correlation)，如圖 8，看看是否有非必要或是重複性的資料。我們發現有許多資料是有相關性的，因此對完全正相關的欄位做檢查並丟棄完全相同的資料。在丟棄完全相同的資料後，此時剩下 25 個欄位。再次，做圖 9 的皮爾森相關分析。我們發現還是有高度相關的資料，但留下的資料已經是經過檢查並非相同。我們判斷會發生此情形，因這些欄位資料幾乎為 0，少數為 1 的情況，故出現高度正相關。這些欄位也會一併加入訓練。

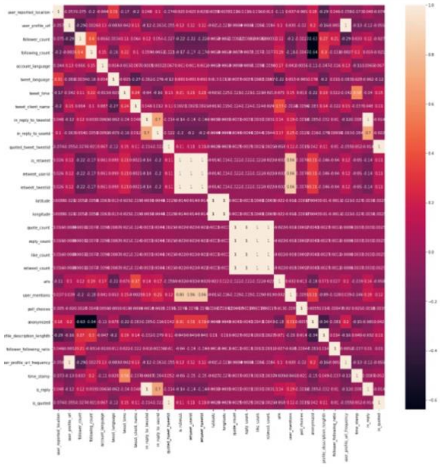

圖8： 所有特徵值未調整前的皮爾斯矩陣

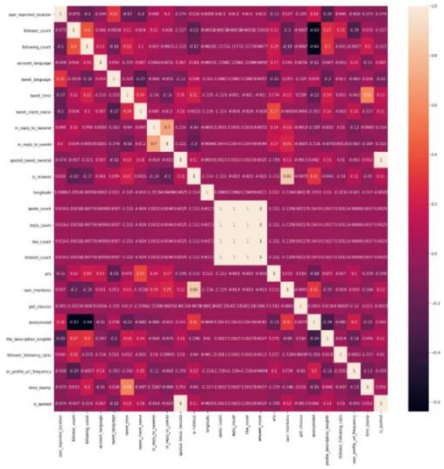

圖9： 特徵值調整後的皮爾斯矩

2.8 自然語言處理轉換

在資料中有兩個欄位需要做自然語言處理轉換，分別是 user_profile_description 及 tweet_text。我們在處理這些文字時，經過 7 個步驟做自然語言前處理：

1. 刪除符號：刪除例如「!」、「?」、「，」等等符號。
2. Google 翻譯：因資料集的文字訊息不只一種語言，大多為俄語，因此先利用 Google 翻譯 API 做轉換成英文。
3. Tokenize：將所有字詞作斷詞。
4. POS：字詞有不同的詞性，也會有過去式、未來式、複數等不同的詞性，先做標註的動作。
5. Lemmatization：將所有詞性做還原，例如：dogs 還原成 dog 或是 ran 還原成 run 等。
6. 刪除 Stop_word：例如：”This is a snake.”，This, is, a 都是 Stop word，它們都沒有意思，只是組成句子的一部分，因此會做刪除。
7. 刪除非字詞：在社群媒體上經常會出現非字詞的可能性，例如，當時的口號縮寫、或打錯字等情況。利用英文辭典列表刪除非字詞的情況。

經過上述的自然語言處理，我們首先嘗試自己訓練一個 Word2Vec 模型 [9]，以英文詞 news 來看看訓練後的結果，如圖 10 所示。可以發現其效果非常差，許多無關的詞都被高度分辨為相似。

```
('air', 0.9902373552322388)
('holy', 0.9795589447021484)
('High', 0.9792883396148682)
('Tula', 0.976150631904602)
('town', 0.9748088121414185)
('Gingerbread', 0.9728690385818481)
('present', 0.969202995300293)
('capital', 0.9672648310661316)
('region', 0.9669650793075562)
('Siberia', 0.9628652930259705)
```

圖10： 自己訓練的 Word2Vec 驗證

我們導入已經 Pre-train 的 Word2Vec，是利用「glove.twitter.27B.100d.txt」由 Glove 收集 Twitter 資訊 [8]，共收集 2B tweets, 27B tokens, 1.2M vocab 的資料並預訓練後的 Word2Vec。再次，利用英文詞 news 來驗證其準確度，如圖 11 所示，詞向量較為接近且字詞也較為合理。

```
('press', 0.7792887091636658),
('newspaper', 0.7733882665634155),
('media', 0.771870493888855),
('interview', 0.7700864672660828),
('reported', 0.752292275428772),
('newspapers', 0.7404190897941589),
('reports', 0.7394641637802124),
('daily', 0.7387121915817261),
('quoted', 0.7365307211875916),
('television', 0.7280975580215454)]
```

圖11： Pre-train Word2Vec 驗證

在 pre-train 的向量驗證效果後，我們想驗證此資料集的字詞向量是否很集中，因此，利用主成分分析(PCA)降維後可視化其分布，如圖 12 所示，可以發現字詞分布是非常集中的。

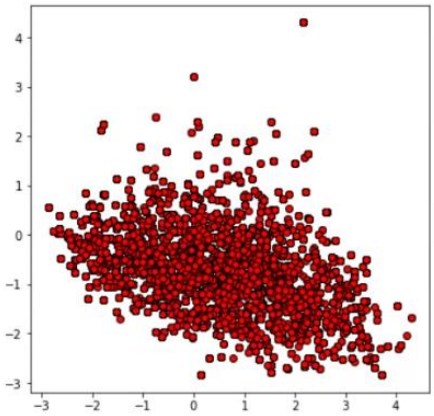

圖12： 資料集字詞向量可視化

參、實驗

3.1 Twitter 資料抓取系統

在本實驗中也建置了一套針對 Twitter 的推文抓取系統，系統建立上基於 ETL 的架構，共分成了三個部分的處理，分別是擷取（Extract) 、轉換（Transform) 、以及載入（Load) 的架構，此架構讓整套資料體系更適合進行資料的保存、處理、分析以及使用，其架構如圖 13 所示，

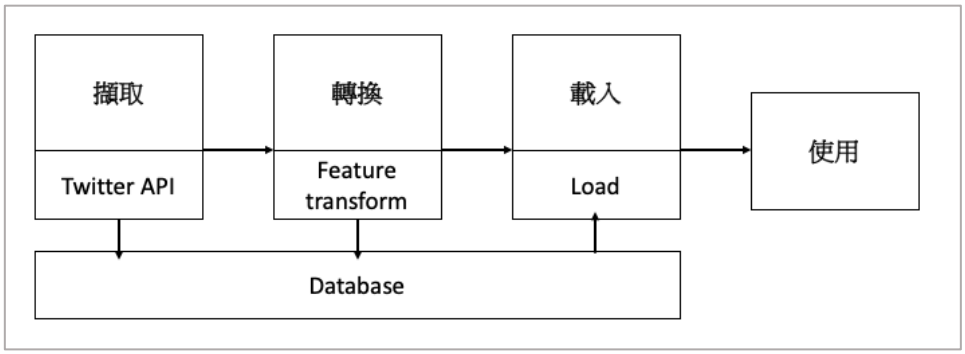

圖13： ETL 資料爬取系統

在本次實驗中，我們先使用 Twitter API 作為我們的爬取工具，抓取 Twitter 上的使用者資訊以及推文資訊，並將其存入 MySQL Database 中，作為我們的原生主要數據，如圖 14 所示，在此數據還處於暫存的狀態下，利用後續介紹到的資料處理產生新的 Feature 並存入資料庫中

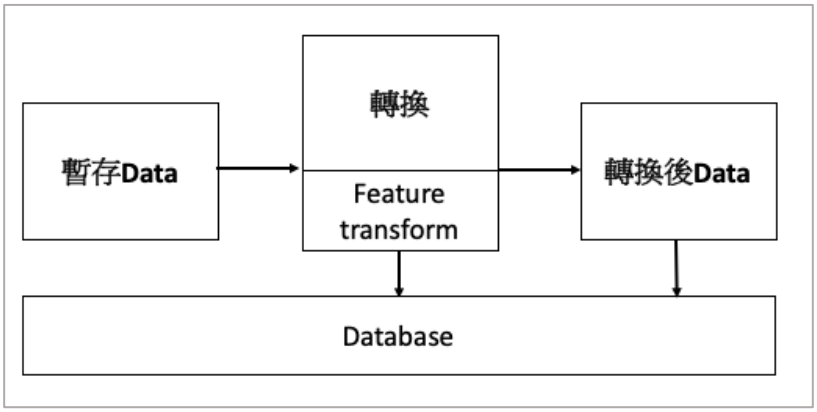

圖14： ETL 系統 Feature 轉換

最後依照所需資料，從資料庫過濾並載入到後續的系統中使用。

3.2 One-class SVM

資料集經過切割成 7,000 筆訓練集，與 3,000 筆測試集，先利用訓練集做出一個 One-class SVM 的異常偵測分辨器。其資料分布以及結果，如圖 15 所示，左側為三維的分辨狀況綠色點點為正常值，紅色 X 為異常值，右邊則為二維。在此分辨器中有 10% 的寬限，意思為可以誤判 10%的異常帳號訊息，以此訓練集來說有 6,300 個異常帳號訊息，以及 700 個被分類為正常帳號的訊息。

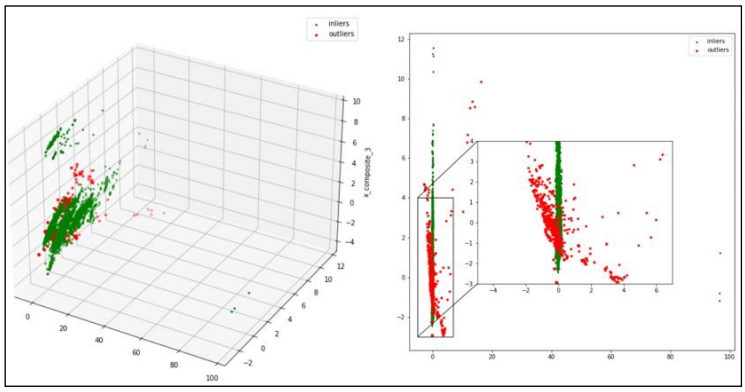

圖15： One-class SVM 訓練集可視化

接下來以測試集來驗證此模型。可以發現結果分布以及分類情況，如圖 16 所示。分辨器來預測測試集的結果中，共有 2,714 條可疑帳號訊息被分辨出來，並有 286 條被分類為正常帳號訊息，準確率為 90%。

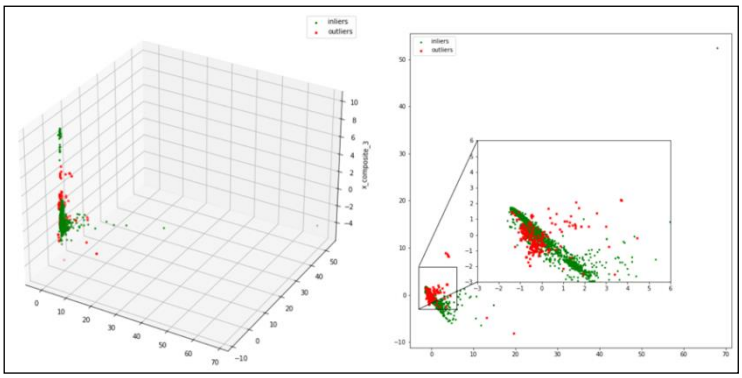

圖16： One-class SVM 測試集可視化

3.3 Isolation Forest

在第二個異常偵測的方法中，我們使用相同的訓練集以及測試集，其訓練後資料分布以及分類結果，如圖 17。我們一樣使分辨器有 10%的寬限，結果也為 6,300 個異常帳號訊息，以及 700 個被分類為正常帳號的訊息。

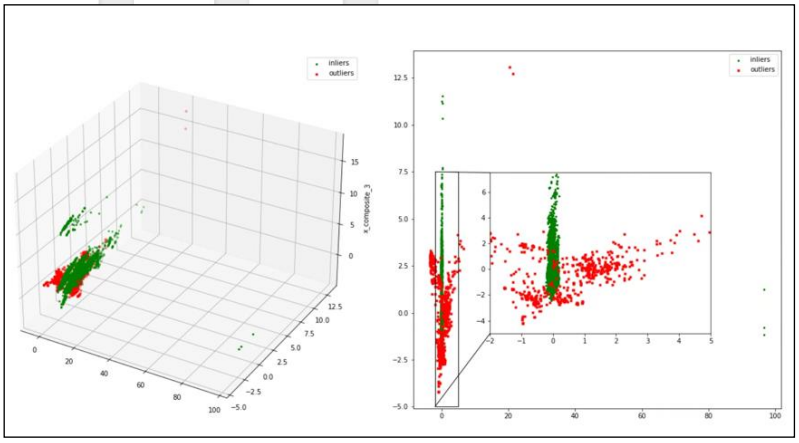

圖17： Isolation Forest 訓練集可視化

接下來以測試集來驗證此模型。可以發現結果分布以及分類情況，如圖 18 所示。分辨器來做測試集的結果中，共有 2,734 條可疑帳號訊息被分辨出來，並有 266 條被分類為正常帳號訊息，準確率 91%。

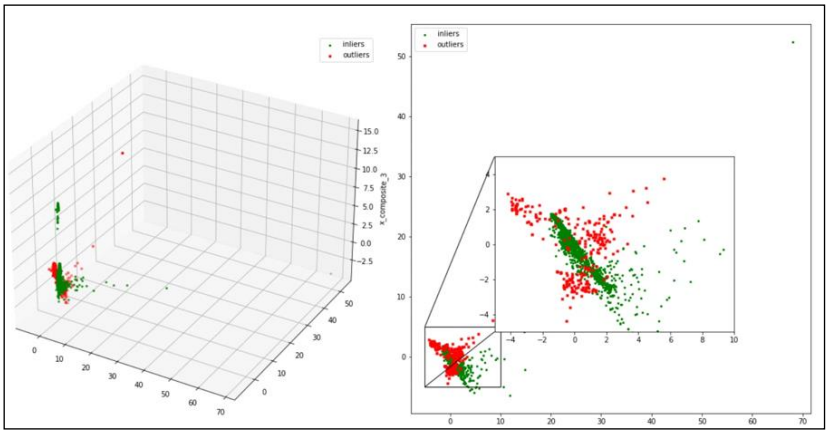

圖18： Isolation Forest 測試集可視化

3.4 正常帳號誤判情況驗證

在訓練完一個異常帳號訊息的分辨器後，我們必須同時考慮到正常帳號訊息是否會被歸類為異常帳號。我們利用 Twitter API 抓取有 Twitter 官方認證「藍勾勾」的名人帳號訊息。分別抓取了 LeoDiCaprio、JimCarrey、Trump、Biden、蔡英文以及柯文哲的推特推文訊息。總共抓取了 1,000 筆，並經過前面所述的資料前處理產生正常帳號訊息的資料集，並利用這些資料分別放入兩個分辨器作驗證。

首先，我們使用剛剛訓練好的 One-class SVM 做測試，其結果會，如圖 19 所示。可以發現此正常帳號訊息資料中的 1,000 筆都被分辨為異常帳號了，因此此模型是不合適的。

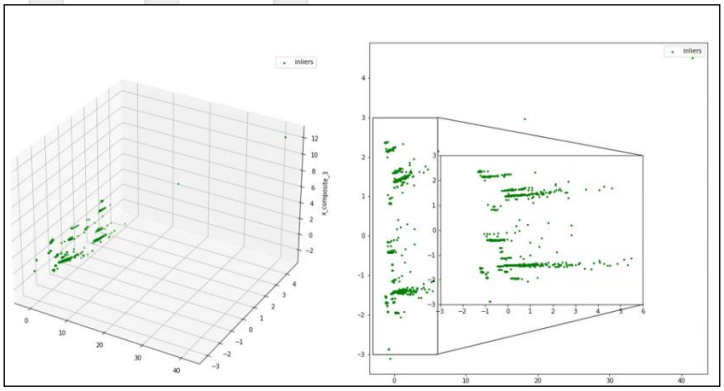

圖19： One-class SVM 正常帳號訊息分辨狀況

接著，我們使用剛剛訓練完成的 Isolation Forest 做測試，如圖 20 中所示。可以發現正常帳號訊息中的 1,000 筆都被很好的分辨為正常帳號了，此模型在此方面也表現得很不錯。

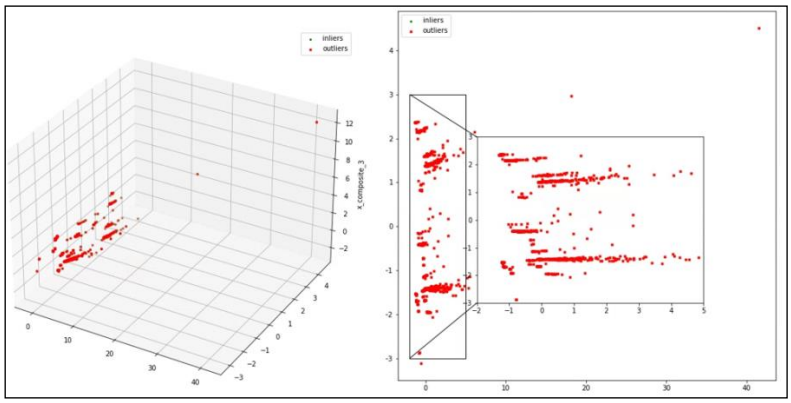

圖20： Isolation Forest 正常帳號訊息分辨狀況

最後，經數據發現 Isolation Forest 模型在分辨異常帳號訊息以及分辨正常帳號訊息的成效相比 One-class SVM 的數據好很多。我們想要再次驗證其模型能力，因此我們混合了 1,000 筆的異常帳號訊息以及 1,000 筆的正常帳號，總共 2,000 筆訊息打亂。並用同樣的 Isolation Forest 模型做分類，其結果如圖 21 中所示。其中，分辨出來異常帳號訊息數量為 923 筆，正常帳號訊息數量為 1,077 筆。正常帳號訊息一樣全數被正確分辨出，而異常帳號錯了 77 筆，準確率 96%。

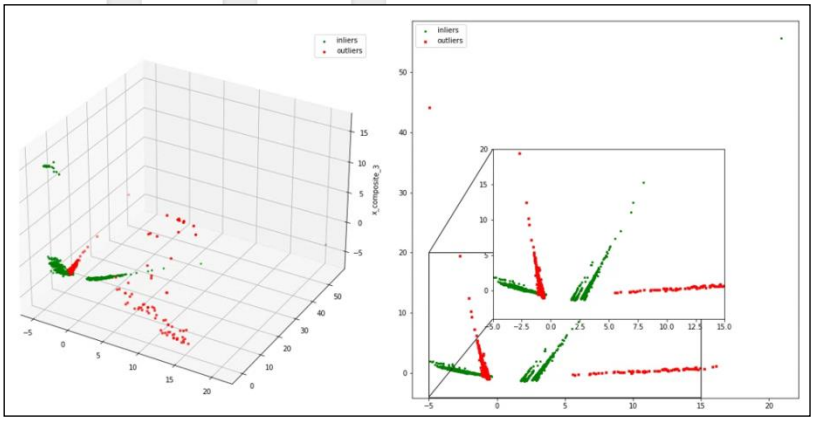

圖21： Isolation Forest 混合帳號訊息分辨狀況

肆、 結論

在一開始的數據處理上，我們建造了一套基於 ETL 的自動化數據處理系統，在數據擷取上利用 Twitter API 完成。也建造了一套基於 Twitter 官方數據的異常偵測系統，在此將異常偵測模型實驗結果總結[15]，其辨識準確率如表一、以及表二所示。以表一的 One-class SVM 而言，實際異常被分類為異常的有 900 條；而被分類為正常的有 100 條。實際正常卻被分類為異常的有 1000 條；而被分類為異常的有 0 條。計算 One-class SVM 的準確率為 45%。

表一： One-class SVM 分辨準確率

One-class SVM	實際異常	實際正常
預測為異常	900	1000
預測為正常	100	0
準確率	45%	

實際異常卻被分類為異常有 923 條；被分為正常的有 77 條。而實際正常被分類為異常的有 0 條；被分類為異常的有 1000 條。計算 Isolation Forest 分辨準確率為 96%。

表二： Isolation Forest 分辨準確率

Isolation Forest	實際異常	實際正常
預測為異常	923	0
預測為正常	77	1000
準確率	96%	

我們將利用效能衡量指標來評估我們的研究效能，總共有四項效能衡量指標。這四項指標可以利用混淆矩陣計算出來，表三為混淆矩陣，共有四種不同狀況，而效能衡量

指標就是由這四種情況來計算出，結果如表四所示。

表三： 混淆矩陣

	真實狀況為真	真實狀況為假
預測為真	TP	FP
預測為假	FN	TN

效能衡量指標計算方式：

Accuracy：(TP+TN)/(TP+FP+FN+TN)

Precision：TP/(TP+FP)

Recall：TP/(TN+FN)

F1-score：2/((1/Precision)+(1/Recall))

表四： 效能衡量指標

	One-class SVM	Isolation Forest
Accuracy	0.45	0.96
Precision	0.47	1
Recall	0.9	0.92
F1-score	0.61	0.96

最後，以所參考的論文[2]中的方法，以及我們所提以 Isolation Forest 為基礎的方法，計算 Precision 做比較，如表五所示。

表五： 相關論文比較

Method	Precision
Logistic Regression	78%
Decision Trees	91%
Adaptive Boosted Decision Trees	94%
Isolation Forest Based (our method)	100%

從表五可以發現，以 Isolation Forest 為基礎的方法其 Precision 比較高。而且參考論文在 Unseen 的資料中是沒有整體準確率可以驗證的，因為該論文中是以人工驗證，基準值也不同，在該論文中的資料集更是極度不平衡的。此外，該論文的資料集正反比例在 98.6%，所以 Baseline 在 98.6%。然而，該篇論文中也有指出此問題。值得一提的是，我們的方法恰好可以解決該論文的這個問題。

[誌謝]

本研究特別感謝 「國家科學及技術委員會」贊助經費，使研究得以順利完成，計畫編號：MOST 111-2221-E-029-016。

參考文獻

[1] V-Dem for digital society project 2018: http://digitalsocietyproject.org/foreign-intervention-on-social-media/

[2] J. Im, E. Chandrasekharan, J. Sargent, P. Lighthammer, T. Demby, A. Bhargava, L. Hemphill, D. Jurgens and E. Gilbert, "Still Out There: Modeling and Identifying Russian Troll Accounts on Twitter," 12th ACM Conference on Web Science, 2020.

[3] Anomaly Detection 2020:https://medium.com/學以廣才/異常檢測 -anomaly-detection-fa300fe6df71

[4] T. N. Kipf and M . Welling, "Variational Graph Auto-Encoders," Bayesian Deep Learning Workshop, NIPS 2016.

[5] B. Du, C. Liu, W. Zhou, Z. Hou, and H. Xiong, "Catch Me If You Can: Detecting Pickpocket Suspects from Large-scale Transit Records," 22nd ACM SIGKDD International Conference, 2016.

[6] L. Fei Tony, T. Kai Ming, and Z. Zhi-Hua, "Isolation-based Anomaly Detection," ACM Transactions on Knowledge Discovery from Data (TKDD), 2012.

[7] Twitter Transparency:https://transparency.twitter.com/en/reports/information-operations.html

[8] Global Vectors for Word Representation: https://nlp.stanford.edu/projects/glove/

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv: 1301.3781, 2013.

[10] M. Mazza, S. Cresci, M. Avvenutil, W. Quattrociocchi and M . Tensconi. "Rtbust: Exploiting temporal patterns for botnet detection on twitter." Proceedings of the 10th ACM Conference on Web Science. 2019.

[11] S. Guarino, N. Trino, A, Celestini, A.Chessa and G. Riotta. "Characterizing networks of propaganda on Twitter: a case study." arXiv preprint arXiv:2005.10004 （2020) .

[12] Y. Liu, and W. Yi-Fang. "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks." Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[13] K. Shu, D. Mahudeswaran, S. Wang and H. Liu. "Hierarchical propagation networks

for fake news detection: Investigation and exploitation." Proceedings of the International AAAI Conference on Web and Social Media. Vol. 14. 2020.

[14] 林祝興, 國家科學及技術委員會專題研究成果報告，MOST 110-2221-E-029-012.

[15] 陳彥翔, 林祝興, “利用異常偵測技術於可疑帳號辨識之研究,” 第三十二屆全國資訊安全會議, 2022。