Università degli Studi Roma Tre

Dipartimento di Matematica e Fisica

Corso di Laurea Magistrale in Matematica

Tesi di Laurea Magistrale in Matematica

Graph-Based Sybil Detection: a caccia di falsi profili nel grafo di un Social Network

Relatori:

Prof. Marco Liverani

Dott. Stefano Guarino

Candidato:

Matteo D'Angelo

Anno Accademico 2014/2015

Sintesi

Il rapido sviluppo tecnologico di cui siamo stati testimoni negli ultimi anni ha reso i dispositivi elettronici sempre più potenti, economici e pervasivi nella vita delle persone, trasformando completamente il modo in cui siamo abituati ad intendere ed a gestire i mezzi di comunicazione e la socialità.

In particolare, fin dalla loro prima apparizione nel Web, i cosiddetti *Online Social Network*, spesso chiamati più semplicemente *Social Network* (SN), si sono velocemente diffusi, vedendo crescere a dismisura il loro numero di utenti e la mole di dati sensibili e/o privati che tali utenti condividono. Il successo dei SN è stato determinato dalla combinazione di due fattori principali: da una parte, essi forniscono nuovi strumenti per rimanere in contatto con i propri amici o familiari, stabilire nuove relazioni in base a interessi comuni, organizzare eventi, condividere pensieri, foto, video e molto altro ancora; dall'altra, permettono di fare tutto ciò facilmente ovunque ci si trovi, necessitando unicamente di una connessione ad Internet, assecondando i frenetici ritmi di vita odierni.

I rischi e le minacce legate ad un utilizzo improprio dei SN sono notevoli, soprattutto a causa della scarsa consapevolezza da parte dell'utente medio dell'effettivo utilizzo ed accessibilità delle informazioni personali condivise su un SN e dei danni che una condivisione inconsapevole possa provocare. Se la *confidenzialità* dei dati che un utente reputa privati e che non intende condividere pubblicamente rappresenta probabilmente il più elementare requisito di sicurezza di un SN, l'inevitabile identificazione tra identità reale e virtuale rende i SN il contesto ideale per un gran numero di tipologie di attacco, le cui finalità si spingono ben al di là del furto di informazioni. Ad esempio, falsificando profili di personaggi o marchi famosi, un'attaccante può danneggiarne l'immagine o sfruttarne la notorietà per ottenere profitti [7]. Più in generale, disporre di un gran numero di profili falsi permette di acquisire illecitamente le informazioni private e la lista dei contatti di un utente, di manipolare/invalidare campagne pubblicitarie, ricerche, sondaggi e votazioni on-

line, di diffondere notizie false, *malware*[*], *spam*[†] [22] e di attuare comportamenti volti ad alterare la popolarità e l'influenza di account reali. Simili attacchi sono resi ancora più efficaci dall'utilizzo di *socialbot*, ovvero programmi capaci di simulare il comportamento umano nell'interazione con un SN [11] e da alcune funzionalità dei SN, come le ricerche personalizzate e i *gruppi*, che agevolano ulteriormente l'implementazione di attacchi mirati ad una specifica categoria di persone, o volti a colpire contemporaneamente un grande numero di utenti.

Un profilo di un SN che non corrisponde alla vera identità di una persona viene solitamente denominato *Sybil*[‡]. Conseguentemente, gli attacchi basati sull'utilizzo di falsi profili sono detti *Sybil attack*. Per via dell'impatto crescente dei SN nella vita quotidiana delle persone, avere il controllo di un gran numero di Sybil permette di influenzare l'economia, la politica e la società, introducendo nuovi rischi per la sicurezza di tutti, dai normali utenti alle grandi aziende.

Un'ampia gamma di attività critiche (come combattere il terrorismo [21], prevedere l'evoluzione del mercato azionario [8], stabilire l'affidabilità dei richiedenti prestito [14] e classificare contenuti, utenti, prodotti ed imprese [24]) dipende infatti al giorno d'oggi da algoritmi di *social engineering*, da analisi dei *social media* e dal cosiddetto *crowd computing*. Profili finti e malevoli possono compromettere il funzionamento di simili strumenti, semplicemente partecipando alla vita di un SN. L'importanza della cosiddetta *Sybil detection*, ovvero della capacità di individuare prontamente profili falsi e malevoli, è stata recentemente evidenziata non solo dalla comunità scientifica, ma anche da media ed influenti blog. Nel recente passato, infatti, blogger e giornalisti hanno riconosciuto l'impatto dei Sybil attack, sottolineando le innumerevoli ragioni che possono spingere alla creazione e all'utilizzo di profili falsi. La scintilla che ha acceso l'attenzione anche di un pubblico di non addetti ai lavori è stata la sospetta inflazione di *follower* nei profili di politici, celebrità e marchi famosi, probabilmente dovuta ad un tentativo di attrarre follower autentici e aumentare la propria fama [14]. Nel web esistono veri e propri negozi on-line che creano e vendono, per pochi euro, falsi profili o semplicemente "like" e "follower" verso alcuni servizi o personaggi [1, 2, 3].

Alcuni blogger si sono spinti persino a proporre strategie per il rilevamento dei Sybil, che però, in mancanza di un impianto teorico a supporto e di algoritmi che

[*]Programmi creati appositamente al fine di arrecare danni ai sistemi nei quali riescono ad infiltrarsi.

[†]Messaggi più o meno diretti che includono link a siti esterni, spesso contenenti malware, o pubblicità non desiderata.

[‡]Sybil è il nome del protagonista dell'omonimo libro di Flora Rheta Schreiber, a cui viene diagnosticato un disturbo di dissociazione dell'identità.

ne dimostrino l'efficacia e la consistenza, rappresentano solo intuizioni utili allo sviluppo di tecniche automatizzate e realmente efficaci per affrontare il problema.

Nella letteratura scientifica, gli approcci proposti per contrastare la proliferazione di falsi profili nei SN possono essere suddivisi in due principali categorie: classificatori basati su *Machine Learning* e classificatori basati su *grafi*. Ad alto livello, entrambe le categorie incentrano il rilevamento dei falsi profili nell'assegnazione di un punteggio ai nodi, il quale descrive la probabilità che essi siano o meno Sybil. Tuttavia, i due classificatori differiscono significativamente nella modalità con cui tale valore viene calcolato.

In questa tesi, analizzeremo i pro e contro di entrambe le tecniche, per poi concentrarci sui meccanismi di difesa basati sull'esplorazione e il partizionamento dei grafi. Prendendo spunto dalla teoria dei grafi e dallo studio delle proprietà di camminate aleatorie su di essi, tale approccio non solo riveste un elevato interesse matematico, ma garantisce anche maggiore profondità e rigore scientifico, permettendo quindi di ottenere risultati più solidi.

Sybil nei Social Network

La prima definizione formale, fornita da Boyd ed Ellison nel 2007 [12], descriveva un *Online Social Network*, o più semplicemente *Social Network* (SN), come un servizio web-based che permette alle persone di:

- Creare un profilo, pubblico, semi-pubblico o privato
- Stabilire una connessione, ovvero un contatto o relazione diretta, con altri utenti
- Osservare la propria lista di contatti e quella degli utenti che ne fanno parte

Come evidenziato da Boshmaf *et al.* in [9] un SN può essere rappresentato tramite un cosiddetto *grafo sociale* (o *social graph*), in cui i vertici rappresentano persone o gruppi di persone, mentre gli spigoli le varie tipologie di relazioni sociali che intercorrono tra di esse. In altre parole, possiamo descrivere un SN con un grafo $G = (V, E)$, in cui ogni nodo $u \in V$ rappresenta un'unica identità (virtuale) e ogni spigolo $e = (u, v) \in E \subseteq V \times V$ una ben definita relazione tra due identità.

Per via dell'enorme mole di dati (privati e non) che circola ogni giorno su un SN e dell'elevatissimo numero di utenti che li utilizzano, essi hanno una notevole influenza sulle scelte di vita e le abitudini delle persone, nonché un crescente impatto sociale, politico ed economico. È quindi naturale che i SN abbiano attirato

da subito l'attenzione della comunità scientifica, interessata in generale a studiarne le dinamiche, ma soprattutto ad analizzarne la sicurezza.

Pur senza trascurare la pericolosità di altri tipi di attacchi, quando si parla di sicurezza dei SN la categoria di attacco più pericolosa, sia per versatilità che per potenziale impatto, è senza dubbio la falsificazione di profili. Come detto in precedenza, gli attacchi basati su falsi profili sono detti Sybil attack.

Sybil attack

Gli utenti di un SN sono spesso inconsapevoli dei danni che può provocare condividere informazioni personali, o considerare attendibili informazioni originate da una fonte la cui affidabilità è difficilmente verificabile. Per via della tendenza diffusa, ma pericolosa, a concepire la popolarità di un profilo come un metro della sua credibilità, utenti comuni sono sempre più spesso esposti a frodi, spam, malware e promozione di materiale illegale, diffuso da account apparentemente autorevoli. In generale, vista la natura aperta e anonima dei SN, la possibilità di creare un numero illimitato di nodi Sybil, che sono a priori indistinguibili dai nodi onesti, è una forte minaccia per gli utenti di un SN, nonché per tutte le applicazioni e gli strumenti pervasivi nella vita delle persone che per il proprio funzionamento si basano tipicamente sulle informazioni reperite tramite SN.

Il termine Sybil attack è stato attribuito per la prima volta a questo tipo di attacco, a discapito dei sistemi *peer-to-peer*, dal ricercatore *Microsoft* John Douceur [16]. I sistemi peer-to-peer rappresentano un'architettura logica di rete in cui i nodi hanno tutti la stessa importanza. Douceur ha osservato che tali sistemi, se privi di strumenti in grado di identificare univocamente ogni singola entità con un nodo della rete, possono essere sempre soggetti a Sybil attack, che rappresentano una minaccia diretta alla *ridondanza* delle informazioni su cui si basano tali sistemi. Questo evidenzia il fatto che il problema della Sybil detection non è solamente circoscritto all'ambito dei SN, ma ricopre un minaccia per un gran numero di sistemi che possono essere rappresentati mediante le cosiddette *Complex Network*.

Tra gli approcci proposti nella letteratura scientifica contro la proliferazione di falsi profili nei SN, rivestono particolare importanza le tecniche basate su Machine Learning [5, 14, 18, 19, 23], che tentano di dedurre la correlazione tra un vettore di caratteristiche misurabili $\mathbf{x}$ e la *natura* binaria y (onesta o Sybil) di un profilo.

Tali tecniche, non richiedono nessuna assunzione *a priori* sulla struttura della rete, ma non sono affiancate da una solida base teorica che permetta di stabilire il loro livello di accuratezza. Di conseguenza esse sono soggette a falsi allarmi ed

errori di classificazione in quanto, la grande varietà e l'imprevedibilità dei comportamenti sia dei nodi onesti che di quelli Sybil, causa delle difficoltà nella realizzazione del *training set*. Tuttavia, queste tecniche sono le più diffuse in letteratura e le più utilizzate. L'approccio tipicamente ingegneristico che hanno alla base, infatti, le rende particolarmente pratiche e scalabili.

Sybil detection con Machine Learning

Con *Machine Learning* (ML), o *Apprendimento Automatico*, si indica generalmente l'insieme dei metodi che permettono ai dispositivi artificiali di rilevare automaticamente dei modelli nei dati, con l'obiettivo di utilizzarli per prevedere dati futuri o per effettuare altri tipi di decisioni in condizioni di incertezza. Un modo per raggiungere questo obiettivo è postulare l'esistenza di un qualche tipo di meccanismo parametrico per la generazione dei dati, di cui non conosciamo i valori esatti dei parametri, attraverso tecniche di tipo statistico.

Il processo di estrazione di leggi generali a partire da un insieme di dati osservati viene denominata *induzione* ed è il meccanismo fondamentale del metodo scientifico: a partire dall'osservazione di fenomeni, come ad esempio la misurazione di un insieme di variabili, ricavare leggi generali che ne descrivono il comportamento. Il processo complessivo nel quale, a partire da un insieme di osservazioni, si vuole effettuare previsioni future rispetto a nuove situazioni prende il nome di *inferenza*.

L'approccio più utilizzato nell'ambito delle ML è l'*apprendimento controllato* (o *supervised learning*) ed ha come obiettivo quello di prevedere, dato un *input* di cui si conoscono un insieme di parametri, il valore corretto dell'*output*. In altre parole, a partire dalla conoscenza di un insieme $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ di input-output, denominato *training set* (o *"insieme delle osservazioni"*), vogliamo dedurre la correlazione statistica che intercorre tra l'input $\mathbf{x}_i$ e l'output y_i delle osservazioni contenute in $\mathcal{T}$, calcolando una distribuzione di probabilità *a posteriori* condizionata da esso, ovvero $\Pr[y_i \mid \mathbf{x}_i, \mathcal{T}]$. Tale distribuzione, eventualmente, può essere utilizzata per avere un'approssimazione della relazione che lega l'output all'input, ossia, assumendo che $y_i = f(\mathbf{x}_i)$, viene effettuata una stima della funzione f. Infine, mediante queste stime è possibile dedurre, per un qualunque valore di input, il corrispondente valore di output. Il training set $\mathcal{T}$, è l'insieme costituito dalle N coppie $(\mathbf{x}_i, y_i)$, in cui $\mathbf{x}_i$ rappresenta l'input ed è un vettore di dimensione D, de-

notato con $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(D)})$ e chiamato insieme delle *feature*, mentre y_i è l'output corrispondente.

Nel caso specifico della Sybil detection, siamo interessati alla *classificazione binaria*, in quanto vogliamo classificare un nodo come onesto o Sybil. In particolare, diremo che un nodo $\mathbf{x}$ è Sybil se il corrispondente output y soddisfa $y = 1$, al contrario è onesto se $y = 0$. Ad ogni nodo vengono attribuite delle feature, ovvero dei parametri misurabili, che descrivono il nodo stesso. Ci sono molte feature che possono essere considerate per descrivere un nodo, le più utilizzate nell'ambito della Sybil detection, in particolare nel rilevamento di Sybil in *Twitter*, sono le seguenti:

- Numero di *hashtag* per *tweet*: descrive il numero di parole precedute da # per ogni messaggio condiviso, detto anche *tweet*. Più hashtag sono presenti e più il messaggio raggiunge un vasto pubblico. Di conseguenza, i nodi con l'intento di produrre spam utilizzano molti hashtag

- Numero di *link* per *tweet*: descrive il numero di link a siti web esterni per ogni tweet. I nodi Sybil condividono spesso molti link che portano a siti web contenenti virus

- Numero di caratteri per tweet: tipicamente i nodi Sybil condividono tweet con pochi caratteri

- Se la foto del profilo contiene una faccia: Tipicamente i profili onesti utilizzano la propria foto per essere riconosciti, mentre i nodi Sybil potrebbero avere una qualsiasi altra foto o addirittura non averla

- Rapporto tra *follower* e *following*: descrive il rapporto tra i nodi seguiti (*following)* e quelli che mi seguono (*follower*)

Come osservato nel caso generale, la classificazione avviene calcolando la probabilità che un nodo qualunque sia Sybil, attraverso la stima della probabilità a posteriori condizionata dal training set. Una volta calcolata la probabilità di essere Sybil per ognuno dei nodi di cui non conosco l'etichetta, viene realizzata una graduatoria. Successivamente su tale graduatoria viene definito un valore di soglia e con esso classifico i nodi: se la probabilità di essere Sybil è maggiore della soglia, allora il nodo è Sybil; al contrario, se la probabilità di essere Sybil è minore della soglia, allora il nodo è onesto.

Nella tesi sono stati analizzati tre schemi di difesa contro la proliferazione dei nodi Sybil in un SN che utilizzano le tecniche di ML: *SybilBelief* [19], *SybilFrame* [18] e *Class A Classifier* [14]. Quest'ultimo è un classificatore binario costruito a corollario di un'accurata analisi delle più promettenti feature utilizzate in letteratura, in termini di compromesso garantito tra accuratezza e costi computazionali. Al contrario, i primi due schemi utilizzano non solo tecniche basate su ML, ma sfruttano anche alcune informazioni sulla struttura topologica del social graph. Questo evidenzia il fatto che, sebbene gli schemi che seguono la linea delle ML per il rilevamento di nodi Sybil ottengano discreti risultati, l'utilizzo delle proprietà dei grafi permette di ottenere una maggiore garanzia in termini di accuratezza e ciò giustifica ulteriormente il nostro interesse verso quel tipo di schemi.

Graph-Based Sybil Detection: Principi di Progettazione

Come osservato in precedenza, la possibilità di creare un numero illimitato di nodi Sybil, potenzialmente indistinguibili dai nodi onesti, rappresenta una seria minaccia alla sicurezza dei SN. Identificare i nodi Sybil è tutt'altro che banale, principalmente a causa della varietà di pattern di interazione sociale riscontrabili sia tra i nodi onesti sia tra i Sybil. Questi ultimi, in particolare, possono essere creati per ragioni molto diverse tra loro ed impiegati per tipologie di attacco talmente varie da impedire di identificare chiari schemi di comportamento ed azione. Come sottolineato in [13], tali difficoltà hanno costretto i *provider* dei SN ad abbandonare gli schemi di difesa basati su *Machine Learning*, spesso chiamati *Automated feature-based Sybil Detection*, per due principali ragioni: (i) tali schemi tipicamente presentano un alto numero di falsi negativi e falsi positivi, che ne limitano notevolmente l'efficacia e l'applicabilità, visto che gli utenti onesti solitamente non reagiscono bene ad un'errata sospensione del proprio profilo; (ii) un attaccante consapevole dei meccanismi di difesa utilizzati può adattare il modello d'attacco realizzando profili in grado di sfuggire al rilevamento [10]. Una possibile soluzione consiste nell'utilizzo di schemi semi-automatici, che, al termine dell'esecuzione dell'algoritmo di ranking, richiedono un'ulteriore scrematura dei risultati, eseguita tramite ispezione umana [28], o sottoponendo gli utenti ad un test per determinare se siano umani o meno. Gli esempi più noti di simili test sono i *CAPTCHA*[§] e il riconoscimento di foto [17].

[§]Acronimo di "Completely Automated Public Turing test to tell Computers and Humans Apart" [4].

Nessuno di questi metodi si è però dimostrato in grado di limitare effettivamente il problema, il che ha portato allo sviluppo degli schemi che prendono il nome di *Graph-based Sybil Detection* (GBSD). L'obiettivo di questi schemi è quello di identificare accuratamente le identità Sybil basandosi prevalentemente sulle proprietà topologiche del grafo di un SN, le quali possono essere identificate principalmente in [6]:

- *Distribuzione dei gradi*: rappresenta la distribuzione di probabilità del grado dei vertici del grafo
- *Diametro*: rappresenta la più grande distanza tra una coppia di vertici del grafo, dove per distanza si intende il più piccolo numero di spigoli che connette due vertici
- *Conduttanza*: rappresenta quella metrica che permette di quantificare la "compattezza" del grafo, ovvero la probabilità con cui una camminata possa raggiungere le parti più isolate del grafo
- *Coefficiente di Clustering*: rappresenta quella metrica che permette di quantificare la "densità" del grafo, ovvero essa calcola la frequenza dei *triangoli* nel grafo, che nel caso dei SN si traduce nella probabilità che "l'amico di un mio amico sia mio amico"

Inoltre, negli schemi GBSD è pratica comune assumere tre proprietà che caratterizzano la struttura del grafo [25, 26]:

i. La regione dei nodi non-Sybil è ben connessa e *fast-mixing*. Ciò significa che la distribuzione di probabilità della posizione occupata da una camminata aleatoria nella regione onesta converge velocemente alla sua distribuzione stazionaria

ii. Anche se un utente disonesto può creare un numero arbitrario di nodi Sybil, non può stabilire un altrettanto arbitrario numero di relazioni con i nodi non-Sybil

iii. È sempre possibile identificare uno o più nodi di fiducia che rompono la simmetria del grafo. In assenza di essi un attaccante potrebbe riprodurre la struttura della regione onesta e di conseguenza nessuno schema di difesa sarebbe in grado di distinguere le due regioni

In poche parole, il rilevamento dei nodi Sybil avviene tramite tecniche di analisi ed esplorazione dei grafi che partono da nodi onesti e dipendono dall'assunto che i nodi Sybil tendono ad essere poco connessi alla rete se confrontati con i nodi onesti e che questi ultimi formano una regione molto densa [9].

Modello Operativo di un Algoritmo GBSD

Gli schemi basati sull'analisi del grafo di un SN hanno differenti caratteristiche. La loro qualità dipende molto dalla topologia della rete e da come vengono distribuiti i nodi *trusted*, cioè i nodi di cui ci si può fidare. Sebbene ci siano queste differenze, osserva Viswanath in [26], tutti gli schemi etichettano i nodi come Sybil o non-Sybil in base alla loro posizione rispetto ai nodi trusted, cercando di partizionare il social graph in due regioni distinte: *regione onesta* e *regione Sybil*. Gli spigoli che connettono le due regioni vengono chiamati *attack edge* o *spigoli d'attacco*. La Figura 1 rappresenta il modello della rete.

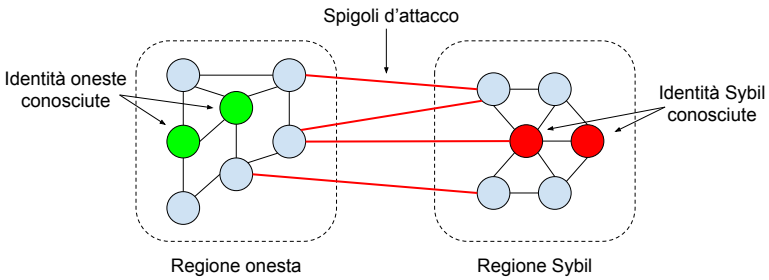

Figura 1: Esempio di modello del sistema

Per quanto detto, ad alto livello, possiamo pensare tali schemi come algoritmi di *graph partioning* [26], i quali sono spesso basati su *camminate aleatorie* sul grafo. L'idea alla base di queste tecniche è quella di assegnare ad ogni nodo un valore di ranking attraverso l'implementazione di tali camminate, sfruttando un insieme di nodi di cui si conosce l'identità. Una volta che ad ogni nodo è stato assegnato un valore di ranking, la natura di un nodo viene stabilita a seconda del fatto che il suo ranking superi o meno un opportuno valore di soglia. La Figura 2 mostra il modello di rilevamento dei nodi. Tipicamente tale soglia viene scelta in modo tale da minimizzare la conduttanza del taglio indotto sul grafo dal partizionamento in regione onesta e regione Sybil [9]. Questo segue dal fatto che l'ipotesi (ii) afferma che il numero di attack edge sia piccolo rispetto al numero di spigoli presenti all'interno delle due regioni e quindi, idealmente, la conduttanza di tale taglio è minima.

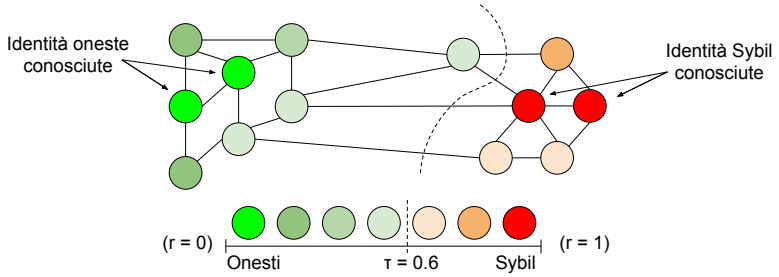

Figura 2: Esempio del modello di rilevamento. La colorazione descrive il ranking: il verde rappresenta i nodi onesti e il rosso i nodi Sybil. Il valore di soglia suddivide il grafo in due regioni disgiunte.

Come già sottolineato, nel rilevamento dei nodi Sybil ricopre un ruolo importante la possibilità di sfruttare le proprietà topologiche del social graph per distinguere la regione onesta da quella Sybil. Sebbene nessuna delle proprietà considerate sia da sola sufficiente a contrastare in pieno gli attacchi ad un SN, è naturale chiedersi quali delle proprietà citate in precedenza sia più utile per rilevare un attacco ed identificare i nodi Sybil. A tal fine, in [6] gli autori suggeriscono un approccio interessante: misurare la *resilienza* di varie proprietà di un grafo sociale di fronte ad un Sybil attack, ovvero in che modo tali proprietà "reagiscono" all'adesione di nuovi nodi Sybil alla rete. Da tali studi si evince che la proprietà più resiliente è la conduttanza, in quanto essa, sotto ragionevoli ipotesi, viene modificata significativamente da un Sybil attack.

Progettazione di un Algoritmo GBSD

L'analisi della resilienza delle proprietà di un SN di fronte all'introduzione di nodi Sybil fornisce indicazioni chiare riguardo la legittimità di un'intuizione diffusa all'interno della comunità scientifica: sfruttare la conduttanza come fondamenta degli algoritmi GBSD. Questa linea infatti è stata seguita da molti studiosi, tra cui Yu *et al.* nello schema *SybilGuard* [30] e successivamente in *SybilLimit*, algoritmi che sfruttano un approccio *decentralizzato* per la Sybil detection, con l'obiettivo di permettere ai nodi trusted di valutare con alta probabilità l'onestà degli altri nodi che costituiscono il sistema. Gli autori affermano che falsi positivi e falsi negativi chiaramente saranno presenti, ma saranno limitati da una rigorosa struttura teorica. Il loro approccio ha indirizzato molti dei lavori successivi verso questa direzione, tra cui i più rilevanti *SybilRank* [13], *SybilInfer* [15] e *SybilDefender* [27], i quali però sfruttano una modalità *centralizzata*.

Oltre all'intuizione che la conduttanza sia il parametro chiave per la rilevazione dei Sybil, tali schemi condividono due caratteristiche che sono alla base della progettazione di un qualsiasi algoritmo GBSD: (i) l'utilizzo di *random walk* per esplorare e campionare il grafo e (ii) l'obiettivo (talvolta implicito) di partizionare il grafo identificando le comunità cui appartengono i nodi onesti conosciuti ed etichettando come Sybil i nodi che non ne fanno parte.

L'utilizzo delle passeggiate aleatorie, in modalità differenti per ogni schema, permette il campionamento di qualche porzione di grafo in modo uniforme per identificare i nodi di cui fidarsi. Esaminiamo questa tecnica a partire da un caso semplice: consideriamo due regioni disgiunte H e S, rispettivamente la regione onesta e quella Sybil; consideriamo inoltre un nodo di partenza u, onesto e un altro nodo v. L'obiettivo è determinare se v sia onesto o meno. Supponendo che entrambe i nodi scelgano uno spigolo a caso della regione a cui appartengono, diremo che u accetta v se effettuano la stessa scelta. Nel caso in cui i vertici appartengono a regioni distinte, ovvero che v sia Sybil, allora il test funziona poiché la probabilità che u accetti v è 0. Altrimenti, la probabilità di collisione è molto bassa, in quanto pari a $\frac{1}{m_H}$ dove m_H è il numero di spigoli di H. Per migliorare il procedimento, possiamo sfruttare il *paradosso del compleanno*. Supponiamo che u possa selezionare un insieme S_u di $\sqrt{m_H}$ spigoli casuali nella regione onesta e in maniera analoga v nella regione a cui appartiene. Il nodo u accetta v se c'è una collisione, ovvero $S_u \cap S_v \neq \emptyset$, e questo avviene con una buona probabilità di successo:

$$1 - \Pr[S_u \cap S_v = \emptyset] = 1 - \left(1 - \frac{1}{\sqrt{m_H}}\right)^{\sqrt{m_H}} \sim 1 - \frac{1}{e}$$

In questo modo, la probabilità che un nodo onesto accetti un nodo Sybil rimane 0, mentre accetta un altro nodo onesto con una probabilità prossima ad 1.

Il passo successivo è quello di generalizzare tale procedimento per consentirne l'uso in un sistema distribuito. Un semplice approccio è quello di considerare una passeggiata aleatoria, sufficientemente corta per garantire una buona efficienza, e prendere, ad esempio, l'ultimo spigolo della camminata. Questa implementazione è corretta purché la passeggiata selezioni gli spigoli in maniera casuale. È qui che entra in gioco il mixing time e quindi la conduttanza, in quanto rappresenta la lunghezza minima della passeggiata affinché possa selezionare gli spigoli in modo equo. Infatti, se la regione onesta è *fast-mixing* (*i.e.*, il tempo di mixing è polinomiale nel logaritmo del numero dei nodi), la probabilità che una passeggiata casuale di lunghezza $O(\log n_H)$, dove n_H è il numero di nodi in H, finisca in u è appros-

simativamente $\frac{deg(u)}{2m_H}$. In questo modo, prendere uno spigolo casuale $e = (u,v)$ incidente con il vertice finale della camminata, significa prendere approssimativamente uno spigolo con una probabilità uguale a

$$\frac{deg(u)}{2m_H}\frac{1}{deg(u)} + \frac{deg(v)}{2m_H}\frac{1}{deg(v)} = \frac{1}{m_H}$$

ovvero in maniera casuale.

In realtà, la regione H e la regione S non sono disgiunte, ma sono collegate tra loro attraverso l'insieme A degli attack edge. Questo implica che una passeggiata casuale che inizi dal vertice $v \in V_S$, possa attraversare un attack edge, entrare nella regione H e selezionare uno degli spigoli selezionati da $u \in V_H$. Tuttavia, se il taglio tra H e S è sparso, ovvero la cardinalità di A non è troppo grande, allora questa situazione è sufficientemente poco probabile, cosicché il meccanismo continua ad avere una buona probabilità di successo.

Il partizionamento ottenuto mediante camminate aleatorie è strettamente legato ad alcuni parametri di input dello schema, in quanto essi definiscono i contorni delle due partizioni. Di conseguenza, è interessante capire quale sia l'impatto che il cambiamento dei parametri di input produce sulle partizioni Sybil e non-Sybil. A tal proposito, Viswanath in [26] effettua un'analisi dei risultati ottenuti dalla simulazione di alcuni schemi con differenti parametri. Esso osserva che il nucleo di uno schema di Sybil detection è costituito da un algoritmo che, dato il SN e i nodi trusted, realizza un ranking dei nodi. Successivamente, mediante il valore dei vari parametri, lo schema individua il punto di cutoff, con il quale si determina il partizionamento nelle regioni Sybil e non-Sybil. Da tale analisi, viene evidenziato il fatto che il ranking è sbilanciato verso i nodi che si trovano all'interno della community a cui appartiene il nodo trusted, cioè i nodi ben collegati con il nodo trusted hanno un'alta probabilità di essere classificati meglio rispetto agli altri.

Questo comportamento potrebbe suggerire la possibilità di sfruttare gli algoritmi di community detection per effettuare il rilevamento dei nodi Sybil. Tuttavia, questa dipendenza dalle community, rende gli schemi fondamentalmente vulnerabili ad attacchi Sybil quando si opera su reti in cui i nodi non-Sybil formano molte community debolmente connesse tra loro (come nel caso dei reali SN). Questo poichè a causa dei pochi spigoli tra le community, lo schema non riesce a distinguere quale sia il taglio tra due community o tra regione onesta e regione Sybil. Di conseguenza la struttura "multi-community" dei SN è uno dei limiti principali degli schemi GBSD. Un ulteriore limite è fornito dalla modalità con cui un attaccante

sferra un Sybil attack. Infatti, ancora dall'analisi di Viswanath [26], si evince che quando un attaccante invece di creare attack edge in maniera casuale, li crea il più possibile vicino a nodi trusted, il tasso dei falsi negativi aumenta notevolmente.

Graph-Based Sybil Detection: Stato dell'Arte

Il problema della Sybil detection iniziò a riscuotere notevole interesse da quando il ricercatore *Microsoft* John Douceur osservò che le reti *peer-to-peer* e più in generale i sistemi distribuiti, sono estremamente vulnerabili ai Sybil attack [16]. L'approccio proposto da Douceur per contrastare tali attacchi è quello di utilizzare una *Autorità centrale* che sia in grado di controllare la rete. Il compito dell'autorità centrale è quello di legare univocamente l'identità "reale" con quella "digitale" di ogni nodo della rete. Un simile approccio *centralizzato* è tipicamente semplice da implementare e da gestire ed è stato proposto anche nell'ambito dei SN, in particolare da Danezis con *SybilInfer* [15], da Cao con *SybilRank* [13] e da Wei con *SybilDefender* [27].

Sfortunatamente, meccanismi di difesa centralizzati non sono efficaci in tutte le reti, sopratutto in quelle la cui natura è aperta e anonima come i SN. In aggiunta, utilizzare una tale unità centrale introduce un cosiddetto *single point of failure*, ovvero concentra la responsabilità di una funzionalità importante del sistema in una singola componente, potenzialmente vulnerabile ad attacchi. Una possibile alternativa è quella di distribuire, in qualche modo, il compito di controllare la rete a più nodi contemporaneamente. Un tale approccio è detto *decentralizzato* e su di esso si basano i primi due schemi graph-based presentati in letteratura: *SybilGuard* [30] e *SybilLimit* [29].

SybilGuard, presentato da Yu *et al.* rappresenta uno dei primi schemi decentralizzati presentati in letteratura per limitare la proliferazione di nodi Sybil in una rete sociale. L'idea alla base di tale schema risiede nell'assunzione che la creazione di una moltitudine di identità Sybil causa sulla struttura del social graph un taglio sparso tra la regione onesta e quella Sybil, ovvero il numero degli spigoli che connettono le due regioni è molto piccolo rispetto a quello degli spigoli interni alle regioni. La ricerca di questo tipo di taglio, in un grafo di cui non si conosce la topologia globale, è un problema complesso [20]. Conseguentemente, in Sybilguard vengono analizzate le intersezioni tra particolari camminate aleatorie sul grafo, progettate in modo tale che un taglio sparso tra le due regioni possa essere usato contro un avversario per limitare il numero di identità Sybil che possano

essere create. Nonostante le intuizioni alla base di SybilGuard siano ragionevoli, l'accuratezza e le garanzie dello schema diminuiscono bruscamente una volta che il numero di attack edge superi una certa soglia. Tale limite viene superato in SybilLimit, nel quale, sfruttando le stesse tecniche e le stesse idee di SybilGuard, viene tollerato un maggior numero di attack edge prima che il rilevamento dei nodi perda accuratezza. Tale miglioramento discende dal modo in cui viene considerata la condizione di intersezione. Infatti, in quest'ultimo, tale condizione viene verificata se le passeggiate aleatorie coincidono nell'ultimo spigolo percorso, alle volte chiamato *coda*.

Le particolari camminate aleatorie sul social graph utilizzate in SybilGuard e in SybilLimit sono chiamate *random route* (o *strade casuali*). A differenza delle classiche random walk, in cui ad ogni passo viene scelto uniformemente a caso uno spigolo da percorrere, nelle random route ogni nodo rappresenta uno svincolo in cui la direzione di uscita è univocamente determinata dalla direzione di entrata. A tal fine, ogni nodo utilizza una permutazione pre-calcolata come una mappa uno-a-uno tra lo spigolo entrante e lo spigolo uscente. In altre parole, ogni nodo genera una *tabella di routing* (o *tabella degli instradamenti*) casuale, che rimane invariata a meno di un cambio del grado del nodo, per scegliere lo spigolo che la camminata deve percorrere.

Entro i termini definiti dai propri autori, entrambi gli schemi ottengono dei discreti risultati. Tuttavia, il loro maggior difetto deriva dal fatto che essi sono fortemente dipendenti dall'assunzione che la maggioranza delle camminate aleatorie iniziate nella regione onesta rimangano nella stessa.

Utilizzando un approccio *centralizzato*, con *SybilInfer* si cerca di superare questo problema considerando in maniera meno rilevante tale intuizione. A differenza dei precedenti schemi, la tecnica adottata da SybilInfer è quella dell'inferenza Bayesiana, la quale assegna ai nodi la probabilità di essere onesti. Sebbene il modello utilizzato in tale schema si avvicini molto a quello impiegato negli schemi basati sulle ML, esso sfrutta specifiche proprietà del social graph sia per realizzare il training set con cui allenare i parametri, che per campionare alcuni tagli su di esso e scegliere quello che meglio rappresenti la sua struttura. Nonostante le simulazioni effettuate con tale schema evidenzino dei buoni risultati in termini di accuratezza, gli autori non forniscono una struttura teorica che permetta di individuare un numero massimo di falsi negativi (*i.e.*, nodi Sybil identificati come onesti). In aggiunta, presenta dei costi computazionali elevati.

È con SybilRank che si ottiene uno schema più efficiente anche in questi ter-

mini. Gli autori affermano però che per ottenere tali miglioramenti occorra riformulare il problema della Sybil detection. In particolare, a causa di un alto tasso di falsi positivi, essi ritengono necessario utilizzare un'ispezione manuale dei nodi che lo schema classifica come sospetti. Per riuscire in questo intento si realizza un'efficiente graduatoria dei nodi in cui la maggioranza dei Sybil occupa le posizioni più in basso. In questo modo, l'ispezione manuale riguarda solamente la parte finale della graduatoria, dove è più probabile incontrare Sybil. Questa graduatoria è costruita in base ad un valore di "affidabilità" ottenuto mediante delle corte camminate aleatorie. Formalmente, l'attribuzione di un punteggio ad ogni nodo avviene, in maniera efficiente, usando delle *power iteration* (o *iterazioni di potenze*) terminate anticipatamente. Mediante questa tecnica, la fuoriuscita della fiducia dalla regione onesta non compromette l'accuratezza nella classificazione, poichè tale quantità è limitata e di conseguenza i Sybil ne ricevono poca. Inoltre, a differenza dei precedenti schemi, SybilRank fa uso di molti nodi trusted, in maniera tale da superare il problema della struttura multi-community di cui sono affetti i SN e che spesso causa errori di classificazione. Parallelamente a quest'ultimo, viene presentato SybilDefender, il quale combina un algoritmo di rilevamento di nodi Sybil e un algoritmo di Sybil-community detection per ottenere una maggiore accuratezza nel partizionamento del grafo. L'intuizione di utilizzare un algoritmo di community detection, in aggiunta a quello di classificazione, deriva dalla difficoltà di esaminare tutti i nodi del grafo per poi identificare la regione Sybil. Quindi rilevare separatamente le Sybil-community a cui appartengono i nodi Sybil identificati risulta più efficiente. Inizialmente le due componenti erano state pensate e realizzate per operare separatamente, ma Wei *et al.* hanno osservato che una loro combinazione risulta maggiormente efficiente. L'intuizione principale alla base di SybilDefender risiede nell'osservazione che il numero di attack edge è limitato, causando così nella struttura del grafo un taglio sparso. Questo consente di affermare che le camminate aleatorie che partono in una regione tendono a rimanere nella stessa, purché siano sufficientemente lunghe per esibire un tale comportamento. Le simulazioni effettuate dagli autori dimostrano che SybilDefender ottiene dei risultati migliori rispetto a SybilLimit. Infatti viene ridotto sia il numero di Sybil erroneamente classificati (falsi negativi) che il tempo di esecuzione.

Conclusioni

Nonostante le varie proposte fatte in letteratura nell'ambito della Sybil detection siano individualmente interessanti, è difficile avere un quadro completo e oggettivo su pregi e limiti di ogni schema. Questo poiché ogni autore presenta il proprio schema ponendosi solamente l'obiettivo di migliorare specifici fattori di quelli precedenti, senza analizzare in profondità ogni aspetto del problema.

A tal proposito, l'obiettivo di questa tesi è stato quello di fornire una panoramica generale sulla Sybil detection, analizzando i due principali approcci e sottolineandone pro e contro. L'utilità di confrontare i vari schemi è quella di fornire due spunti interessanti per i lavori futuri: da una parte vengono evidenziati i limiti comuni, per poterli analizzare e migliorare; dall'altra, gli aspetti più interessanti da cui partire per provare a realizzare un nuovo schema.

Seguendo questa linea, l'intuizione potrebbe essere quella di progettare uno schema che riesca a combinare entrambi gli approcci. In altre parole si potrebbe realizzare uno schema che analizzi e identifichi le principali attività e le relazioni di entrambe le classi dei profili (onesti o Sybil) e con esse ottenere le proprietà topologiche, sia globali che locali, di una rete da utilizzare per effettuare la classificazione.

Nell'immediato futuro, l'obiettivo è quello di implementare interamente ognuno degli schemi citati per cercare di capire ulteriormente le relazioni tra di essi e confrontarli su uno stesso dataset più aggiornato ed ampio di un reale SN.

Bibliografia

[1] Black Market Service 1. http://intertwitter.com/. 2

[2] Black Market Service 2. https://www.fastfollowerz.com/. 2

[3] Black Market Service 3. http://get-likes.com/. 2

[4] Luis Von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. In *Proceedings of the 22Nd International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'03, pages 294–311, Berlin, Heidelberg, 2003. Springer-Verlag. 7

[5] Mansour Alsaleh, Abdulrahman Alarifi, Abdul Malik Al-Salman, Mohammed Alfayez, and Abdulmajeed Almuhaysin. Tsd: Detecting sybil accounts in twitter. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 463–469. IEEE, 2014. 4

[6] Lorenzo Alvisi, Allen Clement, Alessandro Epasto, Silvio Lattanzi, Google Inc, and Alessandro Panconesi. Sok: The evolution of sybil defense via social networks, 2013. 8, 10

[7] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 551–560, New York, NY, USA, 2009. ACM. 1

[8] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8, 2011. 2

[9] Yazan Boshmaf, Konstantin Beznosov, and Matei Ripeanu. Graph-based sybil detection in social and information systems. In *Proceedings of the 2013*

IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '13, pages 466–473, New York, NY, USA, 2013. ACM. 3, 9

[10] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. The socialbot network: When bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 93–102, New York, NY, USA, 2011. ACM. 7

[11] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. Design and analysis of a social botnet. *Comput. Netw.*, 57(2):556–578, February 2013. 2

[12] Danah M. Boyd and N. B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1), 2007. 3

[13] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 197–210, San Jose, CA, 2012. USENIX. 7, 10, 13

[14] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. Fame for sale: efficient detection of fake twitter followers. *Decision Support Systems*, 80:56–71, December 2015. 2, 4, 7

[15] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*. San Diego, CA, 2009. 10, 13

[16] John R Douceur. The sybil attack. In *Peer-to-peer Systems*, pages 251–260. Springer, 2002. 4, 13

[17] Facebook. A continued commitment to security. `https://www.facebook.com/notes/facebook/a-continued-commitment-to-security/486790652130`, 2011. 7

[18] Peng Gao, Neil Zhenqiang Gong, Sanjeev Kulkarni, Kurt Thomas, and Prateek Mittal. Sybilframe: A defense-in-depth framework for structure-based sybil detection. *CoRR*, abs/1503.02985, 2015. 4, 7

[19] Neil Zhenqiang Gong, Mario Frank, and Prateek Mittal. Sybilbelief: A semi-supervised learning approach for structure-based sybil detection. *CoRR*, abs/1312.5035, 2013. 4, 7

[20] Síma Jirí and Elisa Schaeffer Satu. On the np-completeness of some graph cluster measures. *CoRR*, abs/cs/0506100, 2005. 13

[21] Steve Ressler. Social network analysis as an approach to combat terrorism: past, present, and future research. *Homeland Security Affairs*, 2(2):1–10, 2006. 2

[22] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 243–258, New York, NY, USA, 2011. ACM. 2

[23] Bimal Viswanath, M. Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 223–238, San Diego, CA, August 2014. USENIX Association. 4

[24] Bimal Viswanath, Muhammad Ahmad Bashir, Muhammad Bilal Zafar, Simon Bouget, Saikat Guha, Krishna P. Gummadi, Aniket Kate, and Alan Mislove. Strength in numbers: Robust tamper detection in crowd computations. In *Proceedings of the 2015 ACM on Conference on Online Social Networks*, COSN '15, pages 113–124, New York, NY, USA, 2015. ACM. 2

[25] Bimal Viswanath, Mainack Mondal, Allen Clement, Peter Druschel, Krishna P. Gummadi, Alan Mislove, and Ansley Post. Exploring the design space of social network-based Sybil defense. In *Proceedings of the Third International Conference on Communication Systems and Networking (COMSNETS'12)*, Bangalore, India, January 2012. 8

[26] Bimal Viswanath, Ansley Post, Krishna P. Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. *SIGCOMM Comput. Commun. Rev.*, 40(4):363–374, August 2010. 8, 9, 12, 13

[27] Wei Wei, Fengyuan Xu, Chiu C Tan, and Qun Li. Sybildefender: a defense mechanism for sybil attacks in large social networks. *Parallel and Distributed Systems, IEEE Transactions on*, 24(12):2492–2502, 2013. 10, 13

[28] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y. Zhao, and Yafei Dai. Uncovering social network sybils in the wild. *ACM Trans. Knowl. Discov. Data*, 8(1):2:1–2:29, February 2014. 7

[29] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 3–17, Washington, DC, USA, 2008. IEEE Computer Society. 13

[30] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: Defending against sybil attacks via social networks. *SIGCOMM Comput. Commun. Rev.*, 36(4):267–278, August 2006. 10, 13