Identifiering av bottar inom sociala medier

Samuel Kouri
Åbo Akademi
Fakulteten för Naturvetenskaper och Teknik
Kandidatavhandling i Datateknik
Handledare: Mats Aspnäs

Abstrakt

Sociala medier har nyligen drabbats av stora mängder bottar som utvecklats att imitera mänskligt beteende för illvilliga ändamål. Sociala bottars aktivitet kan negativt påverka en tjänst och dess användare om aktiviteten inte förhindras. Olika metoder kan implementeras att identifiera och förhindra det här. Metoderna har sina egna fördelar och nackdelar på grund av hur de tar itu problemet. Granskning per användare är den mest vanliga metoden och har visat bra resultat, men drabbas ofta av integritetsproblem på grund av att an stor mängd data av riktiga användare bearbetas. Korrelationsalgoritmer som studerar användares aktivitet kan implementeras för effektiv mass-identifiering. Korrelationsalgoritmer har dock problem att identifiera sociala bottar som inte hör till ett nätverk av bottar eller sociala bottar som har en låg aktivitet. Nätverkstrafik kan också studeras för att identifiera bottar i realtid.

Bottar inom sociala medier utvecklas i olika slag till det specifika ändamål som krävs. Vissa slag är enklare att identifiera än andra, som till exempel kan de flesta systemen identifiera 100% av följar-bottar vars information matas in i systemet. Identifieringssystemen som utreds i denna avhandling har en total precision av 95 – 97% för identifiering av sociala bottar av olika slag. Sociala bottar utvecklas i snabb takt att bli mer sofistikerade, vilket betyder att undersökningar måste ständigt göras för att effektivt identifiera och förhindra bottar av olika slag.

Förkortningar

CNN	Convolutional Neural Network
DBSCAN	Density-based spatial clustering of applications with noise
DTW	Dynamic Time Warping
ENN	Edited Nearest Neighbor
GDPR	General Data Protection Regulations
GloVe	Global Vector
IRC	Internet Relay Chat
LSTM	Long short-term memory
MLP	Multilayer Perception
NLP	Natural Language Processing
SMOTE	Synthetic Minorty Over-Sampling Technique
SVMs	Support-Vector machine

Innehållsförteckning

1 Introduktion

En bot inom webben är ett datorprogram som utför automatiserade uppgifter inom ett nätverk, till exempel Googles *Googlebot* webcrawler som indexerar webbsidor genom att följa länkar inom webben [1]. De är oftast nyttiga för webben men det finns en mängd bottar som är programmerade med skadliga avsikter. Dessa skadliga bottar skapar risker för leverantören om deras aktivitet inte hindras. En leverantör av en webbtjänst måste iaktta dessa risker och implementera metoder för identifiering och förhindring av skadlig bot aktivitet, för att behålla tjänstens nätverk- och kundsäkerhet.

Bottar som försöker imitera människors beteende har ganska nyligen börjat förekomma inom sociala medier. Detta är på grund av att sociala medier har blivit en viktig aspekt i dagens samhälle och tillför en stor del av människors underhållning och en stor del av företages marknadsföring. Enligt *Statistia* hade Facebook 2,7 miljarder aktiva användare i slutet av år 2020 [2]. Ekonomiska och politiska tankar och idéer sprids i hög grad via sociala medier. Stora mängder följare tillför trovärdighet inom den sociala växelverkan, vilket har lett till en ny bransch av influensers på sociala medier. Monetära vinster kan åstadkommas med stora mängder följare med hjälp av sponsrat innehåll. Detta betyder att det finns möjligheter att utveckla bottar för att artificiellt åstadkomma trovärdighet.

Sociala bottar är programmerade för att växelverka med riktiga användare och använda plattformens algoritmer för att få stora mängder följare till sig och sitt nätverk av sociala bottar, på grund av de ovannämnda orsakerna. Sociala bottar har visat sig bli programmerade för även mer illvilliga ändamål, som att försöka styra användares politiska tankar eller aktiemarknaden. Varlor gjorde en undersökning som visade att 9 – 15% av aktiva Twitter användare är bottar eller uppvisar botliknande beteende [3].

Sociala bottar är väldigt sofistikerade och svåra att skilja från normala användare, på grund av att de klarar Turingtestet. Detta betyder att normala algoritmer ofta inte är lämpliga för problemet. Därmed måste mer sofistikerade och komplexa metoder ändvändas för identifiering. Många moderna identifieringsmetoder använder maskininlärning på grund av att en stor mängd data måste visualiseras och korrelationer mellan olika attribut måste hittas. Dessutom utvecklas sociala bottar att bli mer

sofistikerade med tiden och en maskininlärningsalgoritm kan utvecklas att lära sig medan den identifierar.

Identifiering drabbas dock av problem inom etik och lag. Identifiering av sociala bottar kräver ofta stora mängder data över en användare inom en tjänst. Denna data a kan därmed länkas till användarnas riktiga identitet. Identifieringssystemen måste utvecklas med iakttagelse till EU:s GDPR (*General Data Protection Regulations*) lagar.

Denna avhandling kommer att utreda beteende av bottar inom sociala medier, risker som de medför och hur deras aktivitet kan förhindras. Avhandlingen kommer att studera och jämföra fem olika metoder för botidentifiering: Botometer, DeBot, S. Kuduguntas undersökning, DeBot, P. G. Efthimions undersökning och BotFlowMon. Dessa ovannämnda metoder använder sig av olika arkitekturer och metoder för identifiering av sociala bottar. Systemen och undersökningarna bygger ofta starkt på varandra, som till exempel referenser till tidigare utvecklade systems brister och test för precision genom korsundersökning. Metoderna och undersökningarna grupperas i denna avhandling i tre olika kategorier: *Granskning per användare*, *identifiering genom korrelation* och *trafikbaserad identifiering*.

2 Granskning per användare

Granskning per användare (engelska: *Per-User review*) är en metod där en enda användares aktivitet studeras. Användare hittas och markeras ofta via nyckelord i inlägg och sparas för senare undersökning.

Granskning per användare kräver ofta stora mängder data, tid och resurser. Detta är på grund av att botidentifiering är ett relativt nytt ämne inom informationsteknologi och klassificering för mest effektiva attribut måste undersökas. Integritet är också ett problem som ställer frågor för vilken information ett system kan använda för att inte bryta mot EU:s GDPR lagar [source].

2.1 Botometer

Botometer (tidigare känts som *BotOrNot*) är ett per användare övervakat maskininlärnings system för identifiering av sociala bottar inom Twitter [4], som är det första botidentifieringsramverk som utvecklats och publicerats för kommersiell användning. Botometer är utveklat av *OSoMe*, som är ett gemensamt projekt mellan Network Science Institute, Center for Complex Network and Systems Reasearch at SICE, och Media School at Indiana University [5].

2.1.1 Arkitektur

Systemet använder sig av slumpmässigt genererad skog (engelska: *Random Forest*) metoden för dess maskininlärningsmodell. Modellen är tränad med ett dataset av 5.6 millioner Twitterinlägg.

Slumpmässigt genererad skog är en arkitektur inom maskininlärning. Algoritmen baserar sig på att konstruera flera slumpmässigt genererade beslutsträd. Varje gren i ett träd representerar ett beslut, händelse eller reaktion. Varje träd ger en röst på vad den tror att rätta beslutet är. Hur beslutet fattas varierar beroende på appliaktion, men oftast väljs beslutet med flest röster. Slumpmässigt genererad skog arkitekturen är en av de snabbaste

maskininlärningsarkitekturerna att träna, är bra för stora mängder data och arkitekturen har visats ge resultat med hög precision. Detta har lett till att arkitekturen blivit populär inom dataanalys av stordata.

Botomeeters klassifikationssystem bearbetar över 1000 attribut från den markerade användaren och grupperar dem. Grupperingen består av 6 kategorier:

- ***Nätverks-attribut*** som är ett nätverk baserat på användarens retweets, nämningar och hashtags.
- ***Användar-attribut*** som är baserade på användarens Twitter metadata, vilket inkluderar språk och geografisk position, med mera.
- ***Vän-attribut*** som baserar sig på användarens sociala kontakter, som data över följarmängd och sociala inlägg.
- ***Tidsdrivna-attribut*** som inkluderar hur ofta användaren gör inlägg, gillar andra inlägg och följer andra användare.
- ***Innehålls-attribut*** som bildas från bearbetning av inläggens språk och mening.
- ***Sentiment-attribut*** som bildas från utredning av känslor i innehållet.

Den bearbetade data matas som input i den inlärda Random Forest maskinlärningsmodellen i gruppering enligt attributen. Systemet ger två så kallade CAP (*Complete Automation Probability*) värden för sannolikheten att användaren är en bot. CAP värdena består av ett engelska-värde och ett universal-värde. Engelska-värdet påverkas av mening och sentiment i inlägg, om användaren skriver på engelska. Universal-värdet är riktat mot användare som inte skriver på engelska och påverkas mer av användarens metadata.

2.1.2 Fördelar och nackdelar med Botometer

Botometer har enligt OSoMe en *yta under kurvan* (AUC, engelska: *Area Under the Curve*) precision av 0.95 för att skilja sociala bottar från riktiga användare [4]. Dock enligt underökningar av Torusdağ från Department of Computer Engineering, TOBB ETÜ, Ankara, Turkiet, är dessa siffror inte korrekta [6]. Torusdağ utveckalde i sin undersökning bottar av fem olika slag: *beundrarbot*, *trendämnebot*, *trendämnebot med slumpmässigt*

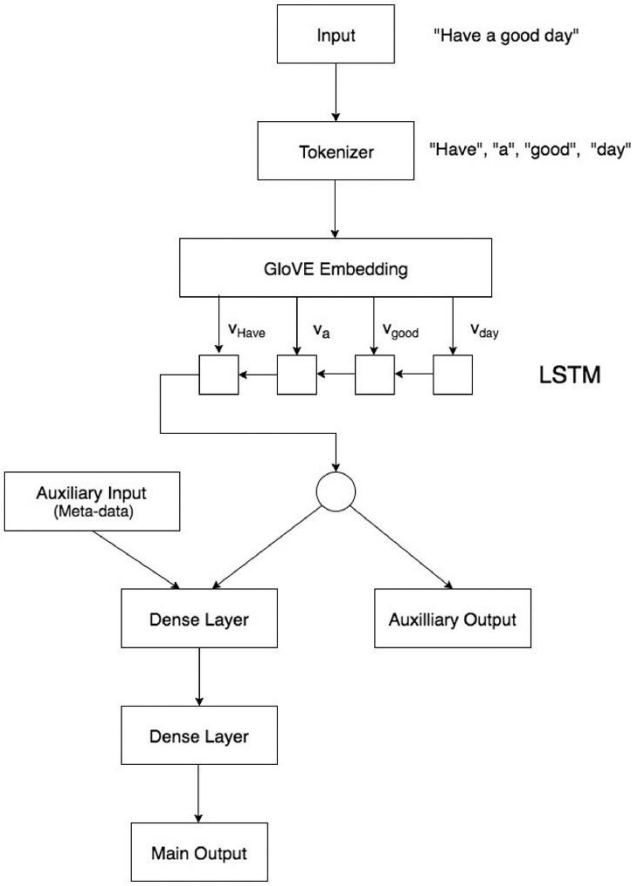
ordval, *propagandabot* och *reklambot*. Enligt Torusdağ kunde Botometers API endast ständigt identifiera *beundrarbottar* och framhäver att mera undersökningar måste göras för att utveckla effektiva botidentifierings system [6].

Feng (Utvecklare av BotFlowMon) kritiserar också Botometer för allvarliga integritetsproblem [7], på grund av att Botometer använder och bearbetar stora mängder data och av riktiga användare.

2.2 Förbättringsförslag av Kudugunta

Problemet med botidentifiaktionsystem av kategorin *granskning per användare*, är att de kräver stora mängder data för validering av en enda användare. Sneha Kudugunta et.al. i artikeln *Deep neural networks for bot detection* [8] undersöker om en social bot kan skiljas från en människa genom att studera ett enda inlägg. Detta skulle leda till att mängden data som krävs kommer drastiskt minska. Kuduguntas djupinlärningsmodell använder sig av kontextuell *Long Short-Term Memory* (LSTM) arkitektur för naturlig språkbehandling (NLP, engelska: *Natural Language Processing*) med en 200 dimensionell GloVE-inbäddning (*Global Vector*).

LSTM är en arkitektur för neurala nätverk. Idén med LSTM är att systemet sparar tidigare inmatningar och ger dessa variabler vikter vilket nätverket läser i ordning. Arkitekturen ger möjligheter att uppfatta mening och sentiment av en skriven text, vilket gör att LSTM är en populär arkitektur för textbearbetning inom maskininlärning. LSTM har använts inom textgeneration, sentiment analys och analys av aktiemarknaden. GloVe-inbäddning är en teknik som används inom LSTM textbearbetning. GloVE-filer innehåller vikter för ord som baserar sig på studier av distanser mellan ord i meningar, vilket ger LSTM arkitektur möjligheten att studera mening i text.



Figur 1: Arkitektur förslag av Kudugunta [8]

Enligt Kudugunta finns det stora brister med att endast studera texten i inlägget [8]. Specifikt valda metadata från inlägget krävs för bra resultat. Kudugunta valde att minska på mängden kännetecken från metadata för att minska på mängden data. Det visade sig att med endast sex kännetecken kan nästan lika bra resultat åtkommas som med hela samplet. Kännetecken består av antalet retweets, svar till inlägget, favoriseringar, hashtags, länkar och nämningar.

En mindre mängd kännetecken ökar neurala nätverkets effektivitet på grund av att systemet bearbetar en mindre mängd data. Dessutom ökar en mindre mängd kännetecken också nätverkets tolkbarhet, vilket ökar utvecklingens effektivitet.

Identifieringssystemet implementerar en algoritm som de benämnt SMOTENN. SMOTENN är en kombination av SMOTE (*Synthetic Minority Over-Sampling*

Technique) for att översampling och ENN (*Edited Nearest Neighbor*) för undersampling. SMOTENN används för att balansera tränings-dataset och har visat sig vara en viktig komponent för bra resultat.

Förbättringsförslaget drar drastiskt ner på mängden data och resurser som krävs för identifiering av sociala bottar. Enligt Kuduguntas undersökningar och tester har systemet en AUC precision av 96% [8] och systemet kan ännu utvecklas vidare för implementation av djupare granskning efter identifieringen för högre precision.

2.3 Efthimions undersökning

Efthimion undersöker i artikeln "*Supervised Machine Learning Bot DetectionTechniques to Identify Social Twitter Bots*" vilka attribut som är mest effektiva och om en mer effektiv algoritm kan implementeras för identifiering av sociala bottar inom Twitter [9]. I undersökningen analyserar och använder Eftimion Cresci-2017 dataset för träning av maskininlärnings modell. Detta dataset innehåller en stor mängd data över användare Användarna grupperas in i tre olika kategorier: *sociala spam-bottar*, *traditionella spam-bottar*, *följarbottar* och *riktiga användare* (Se Tabell 1).

Sociala spam-bottar indelas i tre kategorier: bottar som retweetade politiska kandidater i Italien, bottar som försöker få användare att installera mobil-applikationer och bottar som lägger inlägg av reklam för produkter på webbutiken Amazon.

Traditionella spam-bottar indelas också i tre kategorier: spam-bottar utan ett klart ändamål, spam-bottar som gör reklam och få klicks på en URL och spam-bottar som gör reklam för jobbplatser och få klicks på en URL.

Följarbottar försöker få stora mängder följare för en specifik användare för att få användaren att verka mer populär. Följare kan enkelt köpas som en service via webben av personer som har ett stort nätverk av bottar inom sociala medier.

Gruppering	Antalet användare	Totala antalet inlägg
Sociala spam-bottar	4,912	3,457,344
Traditionella spam-bottar	1,533	6,014,982
Följarbottar	3,351	196,027
Riktiga användare	3,474	8,377,522

Tabell 1: *Distribution av antalet användare och inlägg för Cresci-2017 dataset kategorier* [9]

Enligt Efthimions analys av Cresci-2017 dataset är största indikatorerna att en användare är en bott [9]:

- Standard profilbild
- Ingen beskrivning i användarbeskrivningsfältet eller en länk i fältet
- Samma skärmnamn som Twitter ID
- En vän- till följare-förhållande av högre än 50:1
- Ingen information av geografisk position
- Ett ständigt tidsintervall för inlägg
- Låg mängd inlägg för följarbottar

2.3.1 Arkitektur

Efthimions metod bearbetar information av användare och bildar en matris av sann eller falsk värden (1 = sant, 0 = falskt). Variablerna består av analys av användarens profil-information och inlägg (Se Tabell 2). Matrisen krävs för att Efthimions maskininlärnings arkitektur är baserad på *stödvektormaskin* (SVMs, engelska: Support-vector machine) arkitektur.

SVMs är en algoritm som används inom maskininlärning för en binär linjär klassifikation och bildning av kluster. Idéen med SVMs är att maximera distansen mellan punkter av olika kategorier i träningsläge för att mer exakt placera ny data i dessa kategorier.

Analysområde	Variabel
Profil	Ingen Twitter ID
	Ingen profilbild
	Inget skärmnamn
	Mindre än 30 följare
	Ingen information av Geografisk position
	Språket är inte på engelska
	Användarbeskrivningen innehåller en länk
	Mindre än 50 Inlägg
	2:1 vän/följare förhållande
	Över 1000 följare
	Standard profilbild
	Noll inlägg
	50:1 vän/följare förhållande
	100:1 vän/följare förhållande
	Ingen användarbeskrivning
Text Analys	Levenshtein distans mellan användarens inlägg mindre än 30

Tabell 2: *Efthimions botklassifikations variabler* [9]

Enligt Efthimion har systemet en precision av 95,77% för identifiering av sociala spambottar, 96,25% traditionella spambottar, 100% för följarbottar och 99,87 för ryska NBC News bottar [9]. Systemet har en total precision av 97,75% för identifiering av sociala bottar [9].

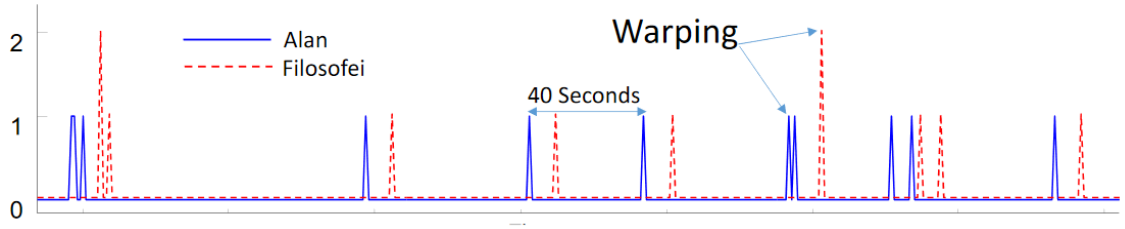
3 Identifiering genom korrelation

Identifiering genom korrelation är en metod som effektivt får och bearbetar data på en stor mängd användare. Bottar av en och samma utvecklare beter sig oftast på samma eller liknande sätt. Detta leder till att man kan studera aktiviteten, som till exempel hur ofta användaren sätter in inlägg, och hitta korrelationer mellan olika användare. Identifiering genom korrelation strävar efter att hitta ställen i tidsutrymmet där en mängd användares aktivitet synkroniseras. Riktiga mänskliga användares aktivitet kan och kommer att slumpmässigt synkroniseras vid vissa instanser då ett långt tidsbelopp studeras. Dock kan ett tal ges för sannolikheten av att detta har slumpmässigt uppstått, vilket är nyckeln för att urskilja falskpositiva instanser från positiva.

3.1 DeBot

DeBot är ett system för identifiering av sociala bottar inom Twitter. DeBot är utvecklat av Nikan Chavoshi och Abdullah Mueen från Department of Computer Science i University of New Mexico, och Hossein Hammooni. Systemet skiljer sig från andra identifieringsmetoder på grund av dess användning av dynamisk tidsförvrängning (DTW, engelska Dynamic time warping) för att räkna ut korrelationer mellan användare av tjänsten [10]. På grund av att systemet baserar sig på att studera aktiviteten, kan systemet hitta korrelationer mellan olika användare oberoende om användarna är korrelerade med varandra genom konventionella metoder. Till exempel behöver bottarna inte följa varandra eller skriva på samma språk.

Enligt Chavoshi är traditionella korrelationsmetoder, som till exempel Pearsons korrelation eller korskorrelation, inte lämpliga för validering [10]. Detta är på grund av nätverksfördröjningar och Twitters interna systemfördröjningar, som orsakar latens. Därför måste DTW implementeras. DTW är en korrelationsalgoritm som har använts inom taligenkänning och online handsignatur kontroll med mera [11]. Algoritmen minskar förvrängning genom att tillåta en elastisk transformation av tidsutrymmet.



Figur 2: *Graf av två användares starkt korrelerad aktivitet under 6 minuters tid* [10]

3.1.1 Arkitektur

DeBot är ett oövervakat system som studerar stora mängder av användares aktivitet, som inlägg, vad användaren gillat och vem användaren följt och att hitta korrelationer inom aktiviteten. DeBot har fyra komponenter*: samlare*, *indexerare*, *lyssnare* och *validerare*.

Samlaren använder Twitters API för att samlar inlägg som innehåller visa onämnda nyckelord och markerar användarna. Samlaren bildar en tidsserie för aktiviteten av alla markerade användares handlingar och filtrerar ut irrelevanta instanser, som till exempel användare med en enda handling.

Indexeraren grupperar de samlade användarnas data med en fördröjningssensitiv hashing metod. För att gruppera användarna genererar systemet ett slumpmässigt tidsintervall och räknar ut korskorrelation mellan aktiviteterna med ett specifikt tal för maximalt tillåten fördröjning mellan handlingarna i beaktande.

Lyssnaren fungerar på samma sätt som samlaren, men i detta fall samlar lyssnaren data på aktiviteten från de redan misstänkta användarna och inte med användning av nyckelord.

Valideraren läser lyssnarens data och hittar korrelationer inom aktiviteterna i tidsutrymmet, bildar kluster av användarna genom warping-korrelation och eliminerar falskpositiva instanser.

3.1.2 Fördelar och nackdelar med DeBot

DeBot har enligt uppgifter från 2016 hittat sammanlagt 540 000 unika sociala bottar och visats ha en AUC precision av 0.94 för identifiering [10]. Precisionen har räknats ut genom korsundersökning med tidigare identifieringsmetoder och Twitters upphävningsprocess.

DeBot kan, på grund av korrelationsalgoritmerna, identifiera nätverk av sociala bottar, oberoende om användarna är korrelerade med varandra genom konventionella metoder. Detta betyder att systemet är väldigt effektiv att hitta stora grupper av bottar inom sociala medier med bearbetning av en låg mängd data per identifiering. DeBot kan dock inte identifiera sociala bottar med låg aktivitet, på grund av att det inte finns en tillräcklig mängd data för att korrelera dem till andra användares aktivitet.

4 Trafikbaserad identifiering

Trafikbaserad identifiering baserar sig på att studera nätverksflöden för att hitta avvikelser i en användares handlingar. Fördelar med trafikbaserad identifiering är att systemet kan studera aktiviteten i realtid och blockera IP-adresser från misstänkta användare snabbare än de tidigare nämnda kategorierna.

4.1 BotFlowMon

BotFlowMon är ett trafikbaserat botidentifieringssystem utvecklat av Yebo Feng, Jun Li och Lei Jiao från University of Oregon och Xintao Wu från University of Arkansas [7]. Systemet baserar sig på att studera tjänstens nätverkstrafik med att agera som en mellanhand för tjänsten och användarna. BotFlowMon baserar sig på tidigare undersökningar inom trafikbaserad identifiering av bottar, som till exempel botnet identifiering och IRC-protokoll (*Internet Relay Chat*) baserade bottars identifiering. BotFlowMon systemet fungerar på nätverk av *NetFlow* formatet. En nätoperatör kan implementera BotFlowMon på vilken som helst router som ligger mellan användarna och tjänsten för identifiering.

Systemet registrerar endast tidsstämplar, IP-adresser, portnummer och paket per sekund. BotFlowMon registrerar ingen information av registrerade pakets innehåll för att skydda paketens och användarnas integritet.

4.1.1 Arkitektur

BotFlowMon systemet har två lägen: Träningsläge och identifieringsläge. Processen består av fem steg, som i systemet representeras av fem moduler (Se Figur 3):

Förbehandlingsmodulen filtrerar bort all trafik som inte är relaterad till den valda sociala medietjänsten och irrelevanta flöden som till exempel flöden med noll bytes. Webbtjänster kan ha flera aktiva IP prefix, därför implementerats *BGPStream* [12] i

förbehandlingsmodulen för att få en lista av alla aktiva IP prefix för tjänsten. Med hjälp av detta kan Systemet i realtid filtrera bort den irrelevanta datan.

Flödesaggregationsmodulen bearbetar NetFlow informationen på en kollektiv nivå och grupperar flöden. Flöden grupperas med att bilda kluster genom en modifierad DBSCAN (*Density-based spatial clustering of applications with noise*), som tar i beaktande positionen av handlingarna i tidsutrymmet. Dessa kluster analyseras för att bilda en uppfattning om vilka flöden hör till samma transaktion.

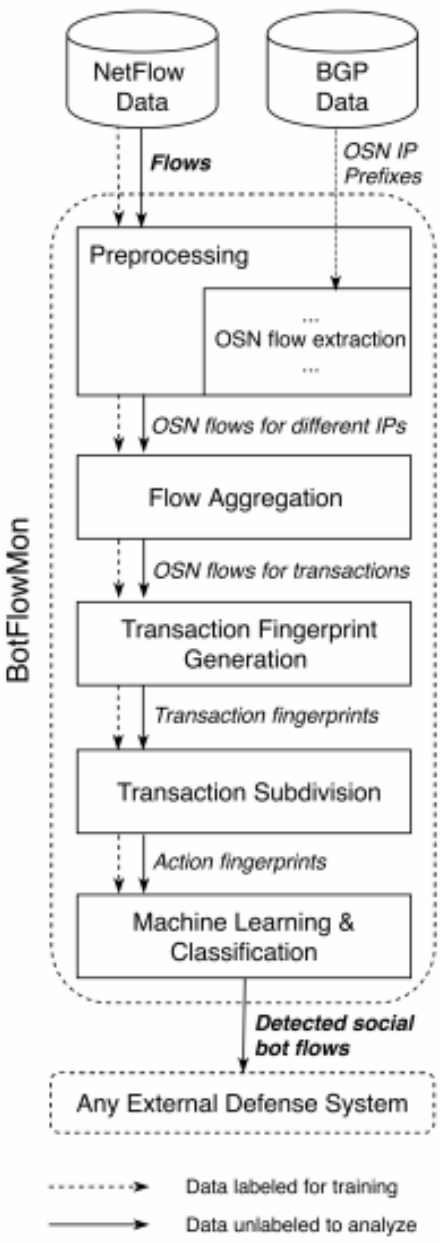
Modul för generering av transaktionsfingeravtryck som bearbetar och extraherar attribut från givna dataset, normaliserar värden och visualiserar flöden.

Modul för indelning av transaktioner till dess handlingar som delar transaktionerna till dess handlingar, som till exempel ett inlägg, kommentar eller klick av gilla-knappen. Feng et.al har utvecklat en algoritm som de kallar *Clustering Based on Density Sort and Valley Point Competition* för att dela transaktionerna till dess handlingar [7].

1: Input: dataset D, radius threshold value r
2: Initialize set C to store clusters
3: Use r to calculate the density of each data point in D
4: $D := Sort(D)$
5: ▷ Sort data in D in the decreasing order of density
6: **for** e in D **do**
7: **if** e is not a potential point of any cluster **then**
8: Label e as a member of a new cluster c_e
9: Add cluster c_e to C
10: **else if** e is the potential point of only one cluster c_a **then**
11: Label e as a member of cluster c_a
12: **else if** e is the potential point of two clusters $\{c_i, c_{i+1}\}$ **then**
13: ▷ e is a valley point, and e can only be two clusters' valley in two-dimensional space
14: $competition(e, \{c_i, c_{i+1}\})$
15: ▷ Start the valley point competition mechanism
16: **end if**
17: **end for**
18: **return** C

Figur 3: *Pseudokod för algoritmen "Clustering Based on Density Sort and Valley Point Competition"* [7]

Manskininlärnings- och klassifikationsmodul som bearbetar handlingarna för att bilda en klassifikationsmodell för identifiering av sociala bottar. Feng et.al har använt *Keras* med *TensorFlow* konstruktion av maskininlärningsmodellen [7]. TensorFlow är ett programbibliotek för maskininlärning och Keras är en ett programbibliotek för att effektivt utveckla program i TensorFlow. BotFlowMons maskininlärningsarkitektur har i största del använt Multilayer Perception (MLP) och Convolutional Neural Network (CNN) i maskininlärningsmodellen för träning och testning.



Figur 4: BotFlowMon process visualisering [source]

4.1.2 Fördelar och nackdelar med BotFlowMon

BotFlowMon har enligt Fengs tester en uppskattad precision av 92,33–93,61 % [7]. Systemet kan implementeras enkelt inom ett nätverk för en effektiv realtidsidentifiering. BotFlowMon har relativt låga krav för hårdvara, enligt Fengs undersökningar kan en laptop med 2.7-Ghz CPU och 16-GB av RAM köra systemet på ett campusnivås nätverk [7].

BotFlowMon kan dock inte urskilja nyttiga sociala bottar från sociala bottar utvecklat med illvilliga ändamål. Feng påpekar att detta är svårt att göra utan innehållet av data i flöden, som systemet inte läser på grund av dess integritetsproblem [7]. Enligt Feng kommer systemet utvecklas vidare för att identifiera olika kategorier av sociala bottar [7]. BotFlowMon är för tillfället begränsad till en specifik miljö av låg trafik och flöden av formatet NetFlow. Systemet grupperar också transaktioner endast enligt IP, vilket betyder att om en riktig användare och en bot är på samma maskin och anslutna till tjänsten på samma gång, kommer systemet att läsa handlingarna att höra till samma transaktion. Detta kan dock enligt Feng utredas med att gruppera transaktionerna enligt IP och portnummer.

5 Sammanfattning

Identifiering av sociala bottar är ett svårt problem att lösa. Bottar utveckals ständigt att bli mer sofistikerade och svårare att urskilja från riktiga användare. Metoder som implementerar maskininlärning har haft bästa resultaten, på grund av att maskininlärningssystem kan lära sig av ny data med medan systemet identifierar. Skillnaden mellan nyttiga bottar och sociala bottar utvecklat med illvilliga ändamål är oftast flummig, därmed har identifieringssystem svårt att kategorisera dem enligt syfte. En balans mellan mängden data som bearbetas och precision måste ännu undersökas och systemen måste ta i beaktande problem inom tjänstens riktiga användares integritet. Integritetsproblem uppkommer på grund av att en stor mängd riktiga användares data bearbetas, vilket kan länkas till användarnas riktiga identitet.

Granskning per användare är den vanligaste metoden för identifiering. Systemen kräver ofta stora mängder data över en användare för att få en bra precision, men undersökningar görs över vilka attribut är de viktigaste inom identifiering. Botometer är det första kommersiella systemet som utvecklats för detta ändamål, men har en låg effektivitet och integritetsproblem. Kuduguntas förbättringsförslag har visat att mängden attribut, och därmed data, kan drastiskt minskas utan att minska på precisionen. Minskning av data leder till en mer effektiv identifiering av sociala bottar.

Undersökningar inom identifiering genom korrelation har gjort stora framsteg för massidentifiering av sociala bottar. Korrelationsalgoritmer kan identifiera sociala bottar vars aktivitet synkroniserar inom tidsutrymmet, därmed kräver identifieringen bearbeta en relativt låg mängd data per användare för en hög precision. Korrelationsalgoritmer har dock problem med att identifiera bottar som inte hör till ett nätverk av bottar eller har låg aktivitet.

Trafikbaserad identifiering av sociala bottar kan i realtid identifiera bottar. Utvecklarna av BotFlowMon har visat att en effektiv identifiering kan implementeras på ett nätverk med låg trafik för realtids identifiering. Systemet kan dock ännu inte lämpligt implementeras på stora nätverkssystem och lider av brister inom generalisering för flera nätverksformat. Dessutom drabbas systemet av problem inom identifiering av sociala bottar med låg aktiviet.

Kategori	System	Precision
Granskning per användare	Botometer	95%* [4]
	Kuduguntas undersökning	96% [8]
	Efthimion undersökning	97,75% [9]
Identifiering genom korrelation	DeBot	94% [10]
Trafikbaserad identifiering	BotFlowMon	92,33–93,61% [7]

Tabell [n] *Nämnda ramverks och undersökningars precision för identifiering av sociala bottar.*

* Torusdağ påstår att detta värde inte är korrekt enligt hans undersökning om Botometers precision [6]

6 Referenser

[1] Google, ”Overview of Google crawlers (user agents),” 10 mars 2021. [Online]. Hämtat från: https://developers.google.com/search/docs/advanced/crawling/overview-google-crawlers. [Använd 15 mars 2021].

[2] Statistia, ”Number of monthly active Facebook users worldwide as of 4th quarter 2020,” januari 2021. [Online]. Hämtat från: https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/. [Använd 22 mars 2021].

[3] O. Varol, E. Ferrara, C. A. Davis, F. Menczer och A. Flammini, ”Online Human-Bot Interactions: Detection, Estimation, and Characterization,” i *ICWSM*, Montréal Québec Canada, 2017.

[4] D. A. Clayton, O. Valor, E. Ferrara, A. Flammini och F. Menczer, ”BotOrNot: A System to Evaluate Social Bots,” i *WWW '16: 25th International World Wide Web Conference*, Monréal Québec Canada, 2016.

[5] OSoMe, ”Botometer,” [Online]. Hämtat från: https://botometer.osome.iu.edu/.

[6] B. Torusdağ, M. Kutlu och A. A. Selçuk, ”Are We Secure from Bots? Investigating Vulnerabilities of Botometer,” i *2020 5th International Conference on Computer Science and Engineering (UBMK)*, Diyarbakir, Turkey, 2020.

[7] Y. Feng, J. Li, L. Jiao och X. Wu, ”BotFlowMon: Learning-based, Content-Agnostic Identification of Social Bot Traffic Flows,” i *2019 IEEE Conference on Communications and Network Security (CNS)*, Washington, DC, USA , 2019.

[8] S. Kudugunta och E. Ferrara, ”Deep neural networks for bot detection,” ScienceDirect, oktober 2018. [Online]. Hämtat från: https://www.sciencedirect.com/science/article/pii/S0020025518306248?casa_token=Bou

_T9wT7NIAAAAA:vrW3VHvatBaSW6bNxkDSe1xBTMMS1QvipZC39x7WwHrc6sVe9ibJ4W05uKD9CRtdgbTCbF3r. [Använd 19 februari 2021].

[9] P. G. Efthimion, S. Payne och N. Preferes, ”Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots,” 2018. [Online]. Hämtat från: https://scholar.smu.edu/cgi/viewcontent.cgi?article=1019&context=datasciencereview. [Använd 18 februari 2021].

[10] N. Chavoshi, A. Mueen och H. Hamooni, ”DeBot Twitter Bot Detection via Warped Correlation,” i *ICDM*, Barcelona, 2016.

[11] P. Senin, ”Dynamic Time Warping Algorithm Review,” december 2008. [Online]. Hämtat från: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.716.1867&rep=rep1&type=pdf. [Använd 18 februari 2021].

[12] C. Orsini, A. King, D. Giodano, V. Giotsas och A. Dianotti, ”BGPStream: A Software Framework for Live and Historical BGP Data Analysis,” i *IMC '16: Proceedings of the 2016 Internet Measurement Conferenc*, New York, NY, United States, 2016.

[13] gdpr.eu, ”What is GDPR, the EU’s new data protection law?,” [Online]. Hämtat från: https://gdpr.eu/what-is-gdpr/. [Använd 27 mars 2021].

[14] M. Orabi, D. Mouheb, Z. Al Aghbari och I. Kamel, ”Detection of Bots in Social Media: A Systematic Review,” ScienceDirect, juli 2020. [Online]. Hämtat från: https://www.sciencedirect.com/science/article/abs/pii/S0306457319313937?casa_token=VlLxWoxMwg8AAAAA:4xG6VUJSW8atspm9UUeuoqkMJfuvDNHhsPQiyFcQfPsWMDPfYxwMlcOhxw992fKl_Jbx0VlS. [Använd 18 februari 2021].

[15] H. Tankovska, ”Number of monthly active Facebook users worldwide as of 4th quarter 2020,” Statistia, 2 februari 2021. [Online]. Hämtat från: https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/. [Använd 18 februari 2021].