Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

SohoNet: A Novel Social Honeynet Framework for Detecting Social Bots in Online Social Networks

Ong Yew Chuan[1,*], Stefania Paladini[2], Belal Alifan[3], Aceng Sambas[1,4], Sharifah Sumayyah Engku Alwi[1], Nur Syakirah Mohd Sedek[1]

[1] Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Besut Campus, 22200, Besut, Terengganu, Malaysia
[2] Queen Margaret Business School, Queen Margaret University, Musselburgh, EH21 6UU, Edinburgh, United Kingdom
[3] Faculty of Information Technology, Philadelphia University, Jarash Road, Amman, 19392, Jordan
[4] Department of Mechanical Engineering, Universitas Muhammadiyah Tasikmalaya, Tamansari Gobras, 46196, Tasikmalaya, Indonesia

ARTICLE INFO

Article history:
Received 23 May 2024
Received in revised form 21 September 2024
Accepted 5 November 2024
Available online 30 November 2024

Keywords:
Social media networks; anomaly detection; unsupervised learning

ABSTRACT

Online social networks (OSNs) are increasingly threatened by social bots – software-controlled accounts that mimic human users for various purposes. In this paper, we propose SohoNet, a novel social honeynet designed to identify, monitor, and detect these malicious entities. This innovative approach improves upon existing research by integrating multiple honeypots with a semi-automatic label engine, thereby significantly enhancing the accuracy of social bot detection. We deployed SohoNet on Platform X (formerly known as Twitter) to analyze activities during the 2022 Malaysian general election over a 14-day campaigning period. Our results show that the semi-automatic label engine successfully auto-labeled 73% of the profiles captured by SohoNet with a moderately high True Positive Rate (TPR) (0.75). Furthermore, SohoNet's overall performance (0.856), measured based on precision and capture rates, surpassed that of existing social honeypots. These findings demonstrate that SohoNet is an effective tool for detecting social bots, particularly in politically sensitive environments. However, the policy of cutting access to X API, along with the costly paid tiers introduced, poses significant challenges for future research as it restricts access to vital data and diminishes the ability to track and analyze bot behavior over time. Future work will aim to extend SohoNet's application across various OSNs to enhance its adaptability and utility.

1. Introduction

The proliferation of online social networks (OSNs) has led to an increase in the presence of social bots that impersonate real users and engage in malicious activities. These bots threaten the integrity and security of online communities by spreading misinformation [1], manipulating public opinion [2,3], and committing fraudulent acts, such as sharing deepfake content [4]. Therefore, understanding and detecting social bots is crucial to maintaining the authenticity and trustworthiness

* *Corresponding author.*
E-mail address: yewchuan@unisza.edu.my

https://doi.org/10.37934/ard.122.1.234248

of OSNs. Researchers have made significant strides in analyzing and detecting social bots, resulting in several bot detection tools such as Botometer X [5] and DeBot [6].

Developing an efficient bot detection system requires a thorough understanding of bot characteristics and behaviors. A common approach involves collecting publicly available social network data, annotating it to distinguish humans from bots, and identifying features that differentiate the two. However, the annotation process is challenging, especially with deceptive bots designed to mimic human behavior. An alternative method involves creating social honeypots – profiles designed to attract anomalies like spammers and bots. Analyzing data captured by social honeypots can reveal current threats to OSNs and provide ground truth data for building supervised anomaly detection systems [7,8]. Unfortunately, the precision of social honeypots is often low [9,10], casting doubt on their ability to produce high-quality data. While human verification of data captured by honeypots is possible, it is tedious and time-consuming [11].

This paper presents the design and implementation of SohoNet (short for Social Honeynet), a framework aimed at providing a systematic approach to detecting social bots in OSNs. SohoNet leverages the concept of social honeypots, introducing a collective approach where multiple honeypots operate in a coordinated manner. While by no means the only example of honeynet, SohoNet is markedly different in terms of conception and/or coding from famous instances as the ones from the HoneyNet project [12], or the well-known T_Po [13], a strong example of a multi-honeypot, powerful but not uniquely focused on OSNs.

Importantly, the use of these kinds of tools, and SohoNet in particular, is beneficial in the sense that it reduces the need for human annotation, by partially automating the bot labeling tasks. As explained later in the text, SohoNet's honeypots are divided into components with distinct functions (e.g., capturing, tracking, and labeling bots). For experimental purposes and proof of concept, SohoNet was deployed on Platform X (formerly known as Twitter) for 14 days, focusing on a case study of the 2022 Malaysian general election.

The rest of this paper is organized as follows: Section 2 provides background information on social honeypots and social bot detection. Section 3 discusses the design and implementation of SohoNet. Section 4 evaluates SohoNet, and conclusions are drawn in Section 5.

2. Literature Review

2.1 Social Honeypots

Social honeypots have emerged as valuable tools in social cybersecurity, particularly within OSNs. Researchers have explored their use to address various challenges related to malicious activities on these platforms. One key trend involves deploying social honeypots to capture evidence of spam profile behavior, aiding in the identification of spammers [14,15]. Frameworks for managing social network honeypots to detect advanced persistent threats (APTs) during the reconnaissance phase have also been proposed [16]. Additionally, integrating honeypots to detect and prevent social engineering attacks has been a topic of interest, leading to novel mechanisms for enhancing security systems [17]. Based on the reviewed literature, three main weaknesses of existing social honeypots can be summarized as follows:

i. Imprecision: Studies have shown that social honeypots perform poorly in capturing bots, with detection rates as low as 12.7% [9] and 38.2% [10]. This poor performance may be due to the difficulty humans face in distinguishing between real users and honeypots, which are also bots [18].

ii. Inefficiency: The inaccuracy of social honeypots leads to inefficiency. Since the collected data is not accurate, it must be manually labeled before being used to train a supervised anomaly detection system. This process is time-consuming and error prone.
iii. Safety concerns: The inaccuracy of social honeypots increases the risk for OSN users to connect with bots, exposing them to threats. Social honeypots may act as bridges connecting real users with bots, as they share the same social circle [19]. Furthermore, the deceptive design of existing social honeypots raises ethical concerns about the consequences of real users being deceived by the honeypots.

Despite these still significant limitations, social honeypots remain a pivotal tool against online threats, providing a foundational method for detecting and studying malicious entities. This motivated the proposal of SohoNet, which aims to address the issues in existing social honeypots.

2.2 Social Bot Detection

Social bot detection is crucial in social cybersecurity due to the increasing sophistication of bots and their potential to spread misinformation and manipulate online discourse. Researchers are actively developing techniques to identify and combat social bots. One prevalent trend involves utilizing advanced technologies such as machine learning [20] and deep learning [21] to enhance detection accuracy. These technologies facilitate the creation of more sophisticated detection mechanisms to keep pace with the evolving complexity of bots [22].

Despite advancements, researchers face several challenges in social bot detection. One significant challenge is bots' ability to mimic human behavior, making it difficult to differentiate them from legitimate users [23]. This challenge highlights the need for more robust detection methods that can effectively distinguish between bots and humans. Additionally, the deceptive nature of bots presents challenges in accurately identifying and characterizing their behavior [24,25]. Moreover, the substantial volume of social bots operating in OSNs poses a daunting task for detection efforts [26].

Given the identified limitations and challenges associated with traditional social honeypots and social bot detection techniques, there is a clear need for more effective and efficient solutions. Addressing the imprecisions, inefficiencies, and safety concerns of existing methods, this study introduces SohoNet. The following section details the design and implementation of SohoNet, explaining how it overcomes the weaknesses of prior approaches and meets the outlined requirements.

3. Methodology

3.1 Key Requirements of SohoNet

SohoNet is a network of social honeypots that work collectively, a key insight that differentiates it from existing social honeypots that operate independently. Motivated by the weaknesses of traditional social honeypots discussed in the previous section, we outlined three major requirements for SohoNet:

i. Precision: SohoNet must capture data with high precision. The honeypots should be designed to accurately capture bot profiles.
ii. Efficiency: SohoNet should operate with minimal manual effort, reducing the need for manual data labeling.
iii. Safety: SohoNet must not harm users of OSNs, including by exposing them to bots.

These three requirements are interrelated in ensuring the framework's overall effectiveness and user protection. High precision in capturing and identifying bot profiles is crucial for the system's reliability, directly impacting its ability to function efficiently by reducing the need for manual data labeling. This efficiency not only streamlines operations but also contributes to safety by minimizing human error and ensuring that users are not exposed to bots.

3.2 SohoNet Architecture

As illustrated in Figure 1, SohoNet comprises multiple components: trapper, interactor, explorer, and tracker. Each of these components contains one or more honeypots managed by the honeynet manager. The honeynet manager acts as the controller and coordinator, overseeing honeypot activities based on predefined configurations, facilitating interactions and information flow among components, and collecting captured data. A semi-automatic label engine partially automates the bot identification task, enhancing efficiency. SohoNet augments raw data with value-added information (e.g., bot responses to interaction, bot behavior over time) through data sharing among its components, thereby increasing the accuracy and utility of the captured data.

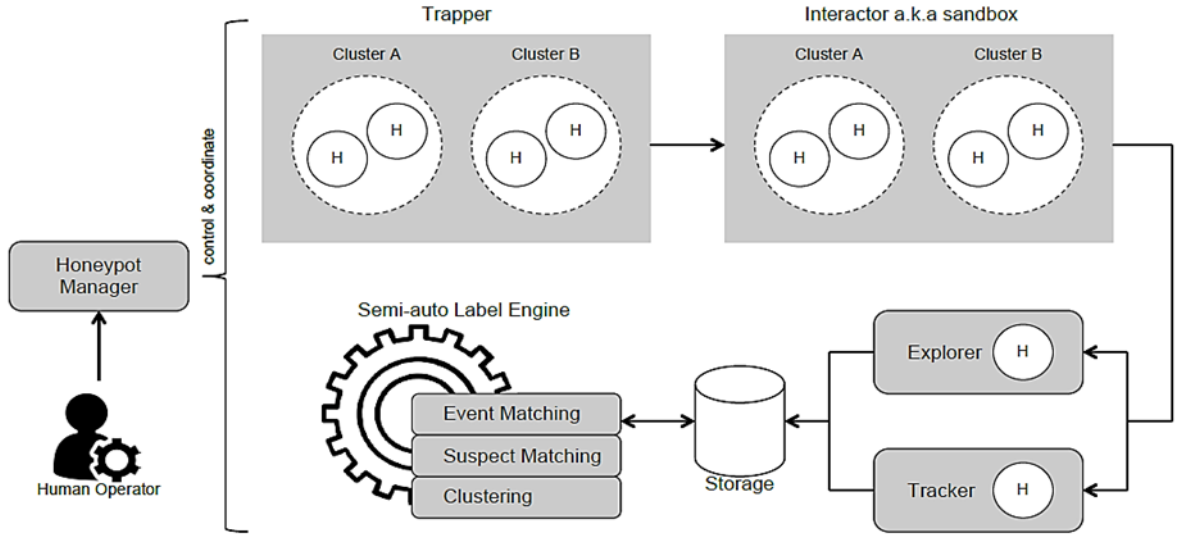

Fig. 1. The SohoNet architecture

3.3 Components of SohoNet

To understand how SohoNet operates, it is essential to examine the specific functionalities of each component in detail:

i. Trapper: This component consists of clusters of honeypots that employ the same type of engagement (e.g., posting the same content) to lure bots. Bots typically react automatically to specific activities within social networks [27]. The trapper honeypots operate passively, awaiting interactions, thereby minimizing the risk of exposing human users to bots.
ii. Interactor: Inspired by sandboxing [28] in network security, the interactor creates a safe, closed environment where honeypots interact with suspects captured by the trapper. The interactor monitors how suspects respond to interactions (e.g., follow-backs). Similar to

the trapper, honeypots in the interactor also operate in clusters to capture automated behaviors.

iii. Explorer: This component gathers basic information about suspects identified by the trapper and the interactor. This information includes profile metadata and a set of posts. For each post, a public search on the social network platform yields related profiles and posts.

iv. Tracker: This component records suspects' activities over time. As anomalies, such as social spammers, evolve to evade detection, bots likely exhibit similar behavior [29]. For each activity type (e.g., post, like, follow), the tracker collects a sequence of activity measures, including timestamps and activity values (e.g., 200 posts at 12:00 p.m.).

The four components of SohoNet work synergistically to enhance bot detection. The Trapper lures bots, the Interactor engages with them in a controlled environment, the Explorer gathers detailed information, and the Tracker monitors their activities over time. This systematic coordination between components ensures efficient and accurate identification of social bots.

3.4 Honeynet Manager and Honeypot

To construct SohoNet, the initial step involves designing and developing honeypots and the honeynet manager. We began by creating a X profile for each honeypot, ensuring each profile included essential information such as a name, profile image, and biography. Default profiles were avoided to prevent being flagged as fake accounts, which could lead to suspension by the social platform operator [30]. Specifically, names were generated using a publicly available random name generator, and screen names were manually created. Each profile description was manually crafted to indicate it was an automated agent, not a human. Unlike studies [31,32] that automate profile creation, we manually completed the process, including solving CAPTCHA challenges.

A honeypot performs three main operations through its social account:

i. update(t, d): Update its timeline with new posts.
ii. interact(t, s): Interact with other profiles by liking, commenting, and reposting.
iii. log(d): Capture data, including details of interactions with other profiles.

These functionalities were implemented using the X Application Programming Interface (API). However, due to API limitations, some tasks, such as activity logging (e.g., likes), were handled by processing email notifications.

Given that SohoNet consists of multiple honeypots, these operations must be automated. This automation is achieved using the X API. Importantly, web automation tools, as used by Boshmaf *et al.,* [33], were avoided to comply with social network operators' rules. The honeypots are controlled by the honeynet manager, which manages their operations based on predefined configurations. Table 1 lists the basic configuration of a honeypot.

Table 1
Basic configuration parameters for a honeypot

Type	Description
COM_ID	The components it belongs (e.g., trapper)
CL_ID	The cluster it belongs, each cluster has different target and content
ACT_TYPE	Types of activities it can perform (e.g., follow)
FREQ	The frequency of an activity (e.g., every 1 hour)

Honeypots delegate their authentication rights to the honeynet manager, which acts on their behalf. Specifically, the manager automates operations such as following accounts and posting updates, provides necessary data including post content and profiles to follow, and logs data by monitoring incoming activities and storing information for analysis. Built as a Python application, the honeynet manager utilizes API calls to automate honeypot operations and accesses their Gmail inboxes to extract notification emails from X. Additionally, it generates content for timeline updates by searching for public tweets with specific hashtags. The search keywords vary depending on the case study. For example, since our experiment focused on the Malaysian general election, some sample search keywords included #GE15, #PRU15, and #malaysiamemilih.

3.5 Semi-Automatic Label Engine

The distinguishing feature of SohoNet is its semi-automatic label engine, which enables the partial automation of social bot identification by leveraging data from four main components discussed in Section 3.3. Before we can understand how the semi-automatic label engine works (Section 3.5.2 to Section 3.5.4), it is essential to first grasp the concepts of events, suspects, and bots within SohoNet (Section 3.5.1).

3.5.1 Event, suspect, and bot

In SohoNet, each interaction between a profile and a honeypot is modeled as an event (*E*), defined as a tuple with five attributes: [*SID*, *HC*, *T*, *ET*, *EID*]. Table 2 details these attributes. Every profile captured by the honeynet is a potential bot suspect (*s*), each with a unique *SID*. Given a group of suspects, *S*, where $S = \{s_i, ..., s_n\}$, the semi-automatic label engine aims to classify each s_i as either a bot (*B*) or unknown (*U*).

Table 2
Attributes of an event

Attributes	Description
SID	Suspect ID
HC	The cluster which the honeypot belongs to
T	Timestamp of the event
ET	Type of the event (e.g., like, follow)
EID	Event ID

3.5.2 Phase 1: Event matching

The trapper and interactor component in SohoNet comprises pairs of honeypots that behave identically (e.g., following the same person). The semi-automatic label engine leverages this unique design to partially automate the bot labeling process. The engine begins by categorizing an event (*E*) into two types:

i. One-to-one: The suspect interacts with only one honeypot.
ii. One-to-many: The suspect interacts with multiple honeypots within the same cluster.

At this stage, the primary focus is on detecting one-to-many events, as these indicate a high level of automation. It is unlikely for a human to interact with multiple honeypots within a predefined time interval. To meet SohoNet's safety requirements, suspects involved in one-to-one interactions are not labeled as bots, as legitimate users might accidentally interact with a honeypot.

To detect all one-to-many events, the engine first identifies the matched events among all events triggered by SohoNet. Recall that an event (*E*) is a tuple, [*SID*, *HC*, *T*, *ET*, *EID*]. Two events are considered matched (≃) if they have the same attribute values for *HC* and *ET*, and if the event timestamps (*T*) are within the same window of a predefined interval (T_{int}, e.g., 15 minutes). A suspect (*s*) with the *SID* of the matched events is then assigned to a bot group (*B*). Formally, s_i, $s_j \in B$ if $E_i \simeq E_j$, where $E_i \simeq E_j$ if and only if the following conditions are met:

i. $[HC_i, ET_i] = [HC_i, ET_j]$
ii. $|T_i - T_j| \leq T_{int}$

To measure the similarity of event tuples, the Jaccard similarity coefficient [34] is employed. The Jaccard index ranges from 0 to 1, with 1 indicating an exact match. A strict threshold of 1 is applied to identify matched events, with the matching decision constrained by the time interval. Phase 1 results in a group of suspects (*s*) assigned to the bot group (*B*), while the remaining suspects are grouped in *U* and passed to Phase 2.

3.5.3 Phase 2: Suspect matching

Bots can exhibit varying degrees of automation and synchronization, working collectively to maximize impact. Previous studies [35,36] have highlighted how bots synchronize their profile information and behaviors. Detecting synchronized bots through clustering methods is challenging, as it requires analyzing large numbers of profiles. For instance, Chavoshi *et al.,* [35] identified 1,485 synchronized bots among one million X profiles using the DeBot API. Event matching in the first phase of the operation of the semi-automatic label engine also cannot detect these bots, as they appear as distinct bots interacting with different honeypots within the same cluster.

To identify synchronized suspects, three pieces of information are considered: the suspect's URL, description, and content. The matching process runs sequentially, with each unmatched suspect evaluated using the next criterion. Suspects that do not match any of these criteria at the end of the process will be assigned to the unknown group (*U*).

i. Criterion 1 URL: The URL refers to the profile's shared URL. Initially, any shortened URLs are expanded, and the Jaccard similarity coefficient is used to find matched URLs. URL matching is a straightforward process compared to existing methods for clustering profile names, which require a set of labeled profiles to generate a Markov chain for similarity comparison [37].
ii. Criterion 2 description: The description is a short text describing a profile. Unlike event and URL matching, which seek exact matches, description matching aims to find nearly identical texts. The correlation of the overlap coefficient proposed by Lee *et al.,* [38] is used for this purpose. Following the recommendations by Lee *et al.,* [38], 4-shingling is used to split the description, and a typical threshold of 0.6 or above is applied to consider two descriptions as matched.
iii. Criterion 3 content: Content refers to the posts produced by a suspect (e.g., tweets). Instead of matching suspects' posts directly, data collected by the Explorer is used for a broader matching range. For each suspect s_i, a set of posts, $C(s_i)$ is extracted, where $C(s_i) = \{c_i, \ldots, c_n\}$. For each s_i, the Explorer performs a public search within the social network, yielding a set $R(c_i)$. $R(c_i)$ contains pairs $\{(p_i, tw_i), \ldots (p_n, tw_n)\}$ where p represents a profile and tw (short for tweet) represents a post produced by p. Since not all search results are relevant, we again use the correlation of the overlap coefficient to measure the similarity

between c_i and r in $R(c_i)$. This method is effective since posts are short texts (e.g., 280-character tweets). The computation results in a new list of profiles, each having at least one post matching with c_i. This is an indication that these profiles are bots involved in a coordinated campaign. All profiles in this list are labeled as bots by the engine.

3.5.4 Phase 3: Clustering and manual labeling

In the final phase, we apply k-means clustering [39] to classify the remaining suspects in the unknown group (*U*) into either the bot group (*B*) or the human group (*H*). Clustering is chosen because it is an unsupervised method that does not require pre-labeled data, making it ideal for identifying patterns in unlabeled datasets.

To perform clustering, we utilize several features derived from the activity sequence collected by the Tracker component. The activity sequence is selected because it captures a suspect's behavior across various activities, including likes and follows, more comprehensively than the suspect's timeline. The features used are as follows:

i. Ratio of activity types: This includes the ratios of static, add, and delete activities.
   (a) Static activity refers to no changes within a one-hour interval.
   (b) Add activity denotes positive changes.
   (c) Delete activity indicates negative changes.
   These features help identify the like-unlike and follow-unfollow strategies used by bots to attract attention while masking aggressive behavior, such as an unusually high number of average likes per day.
ii. Maximum active hours: This metric measures the maximum number of hours a profile is active, with a profile considered active if an add or delete event is detected. Bots generally exhibit higher numbers of active hours since automated programs can operate continuously, unlike humans.
iii. Mean and standard deviation: These statistics measure the average and variability of fluctuation counts within the tracking timeline. Bots are expected to have higher mean fluctuation counts.
iv. Fano factor: This factor identifies potential bulk activities by measuring the dispersion of Fano noise [40]. A high Fano factor suggests automated activities.

As for the labeling process, it was carried out by three undergraduate computer science students. They were first trained by the authors to analyze and identify social bots. To support this task, an annotation dashboard (see Figure 2) was created, displaying detailed information about each profile, including tweet types, interfaces, and volume, to enhance the accuracy of the annotations.

For this labeling task, we used a majority voting method after completing each set of annotations. When the three annotators disagreed, we consulted an expert annotator (a member of the author team) to make the final decision. To ensure the quality of the annotations, we calculated the inter-annotator agreement (IAA) using Fleiss' Kappa score [41]. The IAA score was 0.82, indicating satisfactory quality of the annotations.

Operationally, our annotation process included both suspects and bots, as the labels were also required for evaluation purposes in Section 4. From the SohoNet operational perspective, we only needed to label a sample of profiles from each cluster to identify the bot group (*B*) and the human group (*H*).

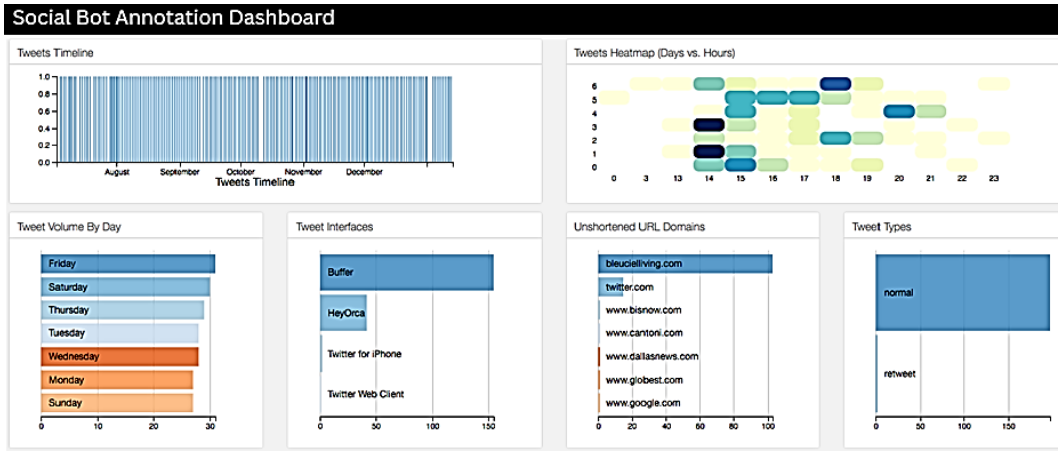

Fig. 2. The social bot annotation dashboard

3.6 Ethical Principles of SohoNet

For SohoNet to function effectively, it must also comply with a set of ethical guidelines to minimize risks to stakeholders within the online social platform. By referring to existing ethical guidelines on developing social honeypots and automated agents [42-44], we outlined three major ethical principles for SohoNet.

3.6.1 Adherence to terms of service

Every social platform has its own terms of service, which must be respected to ensure ethical decisions during the creation and operation of SohoNet. While there are various methods to create social profiles, some may raise ethical concerns. For instance, Stringhini *et al.,* [9] partially automated the profile creation process by automatically filling out 300 registration pages and manually solving CAPTCHAs. In contrast, Zhang *et al.,* [32] fully automated the process by purchasing 1,000 Twitter profiles. Accessing a social platform through automated means or purchasing fake profiles is generally considered unethical without strong justification. Therefore, we manually handle profile registration and CAPTCHA solving when deploying social honeypots in SohoNet.

3.6.2 Non-deceptive design

Deceptiveness is a hallmark of traditional honeypots, designed to attract attackers into the system. However, such design can potentially harm users by causing feelings of deception or exposing personally identifiable information [45-48]. Even though debriefing users at the end of a deceptive experiment is ethically preferable, it cannot undo any harm already inflicted on real users. Therefore, SohoNet ensures a non-deceptive design by clearly indicating on our profile that we are social honeypots. While using deception might prevent SohoNet from being detected by some sophisticated bots, it is not worth compromising the rights of real users to achieve better research results.

3.6.3 Respect for copyright and privacy

The collection and use of content and information within a social networking platform involve critical copyright and privacy issues. Creating and operating SohoNet requires various types of content, such as profile images during registration and posts to update the profile's timeline, as well

as data collected from suspects. It is essential to handle the source, copyright status, and privacy of this content with care to avoid unethical actions. For example, using profile images for honeypots may violate copyright laws if their source or copyright status is unclear. To address this, we follow the approach of Paradise *et al.,* [16], who used non-copyrighted images in their research involving social honeypots. SohoNet has adopted this same ethical process.

4. Results

To demonstrate the practical application of the proposed SohoNet, we deployed it on Platform X. Unlike the previous study that focused on the 2016 US Presidential Election [49], our research centers on the 2022 Malaysia General Election. This political case study was chosen due to the documented involvement of bots in political activities worldwide [50].

Our SohoNet consisted of 18 honeypots. The distribution of these honeypots, based on their components and types of engagements, is detailed in Table 3. The SohoNet operated from November 5th to 18th, covering the entire campaigning period of the general election. During this period, SohoNet triggered 911 events from 263 unique suspects. These events included 375 follows, 287 likes, 109 retweets, 88 messages, and 52 mentions and replies, averaging 65 events per day with 18 suspects identified daily. The discrepancy between the number of events and suspects was due to the clustering design, where a single suspect interacted with multiple honeypots within the same cluster, triggering multiple events. Figure 3 illustrates the timeline of various types of engagements detected by SohoNet throughout its operation.

Table 3
Engagement types and quantities in SohoNet

Components	Types of engagement	Quantity
Trapper	Tweet	8
Interactor	Like	2
	Follow	2
	Retweet	2
Explorer		2
Tracker		2

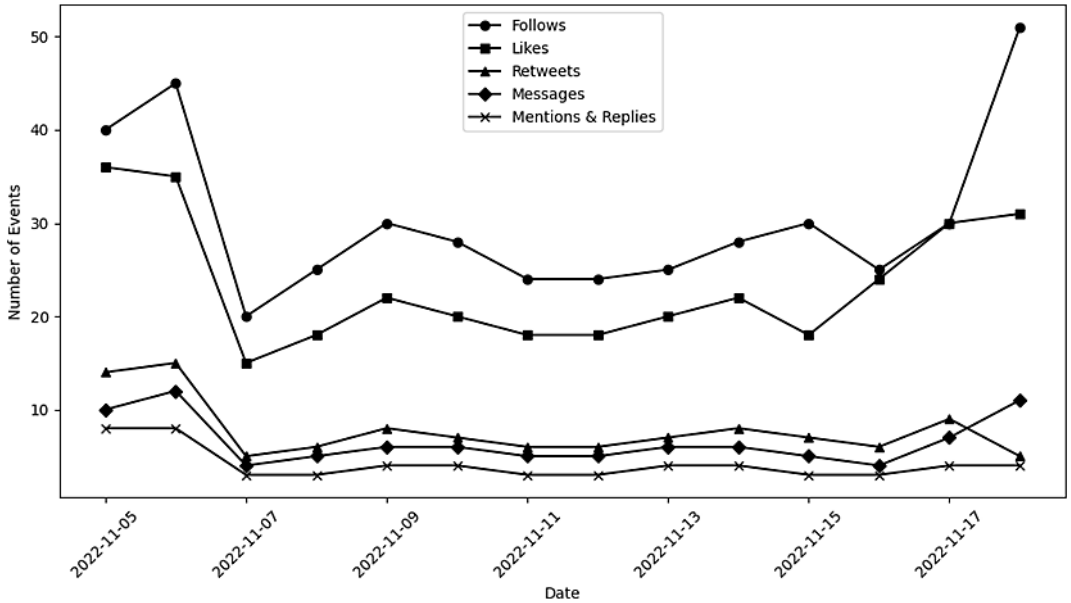

Fig. 3. Daily engagement events on SohoNet

The timeline reflects the dynamics of an election, with spikes during the nomination day and increased engagement as the campaign progresses, particularly towards polling day. The graph shows significant initial spikes in follows and likes. These activities decline sharply after the initial peak but gradually rise again towards the end, peaking on the day before polling. Retweets follow a similar but less pronounced pattern, while messages and mentions & replies maintain a lower, more stable level of activity throughout the period.

It is noteworthy that the Trapper captured 183 unique suspects, while the Interactor attracted 80 new suspects. This indicates that bots employ strategies beyond keyword searches, such as follower and liked tweet searches [51]. The semi-automatic labeling engine identified 171 (65%) of the 263 suspects as bots in Phase 1. In Phase 2, 21 suspects (8%) matched one or more of the three criteria, bringing the total labeled to 73%, with 71 suspects remaining. Through the manual annotation process, we identify there are 64 bots and 7 humans among these remaining suspects.

4.1 Evaluating the SohoNet

We evaluated the overall performance of SohoNet by introducing a new performance metric that balances quality and quantity. Previous studies primarily used capture rate (CR), which measures the average number of profiles (or engagements) captured by each honeypot per day [10]. However, relying solely on capture rate is insufficient, as high capture rates accompanied by false positives are not useful. It is impractical for social honeypots to capture many profiles as suspects when only a small number of these suspects are social bots.

We term this new metric the overall performance (OP) metric, which incorporates both precision (Prec.) and CR, with adjustable weights:

$$OP = (Prec. \times weight_1) + (\frac{1}{1+ e^{-CR}} \times weight_2) \quad (1)$$

An ideal honeypot will have an overall performance of 1, indicating perfect precision and capture rate. For our study, we set both $weight_1$ and $weight_2$ to 0.5 to achieve a balance between these evaluation metrics. Table 4 summarizes the precision, capture rates, and overall performance of existing social honeypots. We compared our work with two other studies because these are the only ones that provide false positive values, enabling us to calculate precision. The results show that SohoNet's strategy, which collectively lures bots, achieves positive outcomes.

Specifically, we attained the highest precision compared to Stringhini *et al.,* [9] and Yang *et al.,* [10]. Regarding capture rate, we ranked third among the works reviewed. While studies by Lee *et al.,* [52] and Lee *et al.,* [53] show notably high capture rates, the precision of their work remains questionable. This is because they assume all profiles captured by their honeypots are anomalies. However, this assumption is flawed since research has shown that some legitimate human users reciprocate interactions with anyone, including honeypots, without careful examination, simply to increase their social capital [54,55]. Nevertheless, it is important to highlight that such comparisons may carry some inherent bias, as the social honeypots were deployed at different times on different social network platforms.

Our engine achieved a TPR of 0.75. While this is moderately high and acceptable, it suggests there is room for improvement. To gain deeper insights into the performance, we manually analyzed the events and suspects that our engine failed to identify as bots. Our investigation revealed the presence of synchronized bots that engage with our honeypots in diverse ways within the same cluster. For instance, two bots responded to the same tweets from our honeypots with different actions: one by

liking and another by retweeting [56]. Our engine did not flag these as bots because we set a strict threshold that expects an exact match in the detected events.

Table 4
Performance comparison of SohoNet and existing social honeypots

	Platform	Prec.	CR	OP
Our SohoNet	X	0.973	1.04365	0.856
Webb *et al.,* [7]	MySpace	-	0.23514	-
Lee *et al.,* [52]	X	-	1.74667	-
Lee *et al.,* [53]	X	-	1.85031	-
Stringhini *et al.,* [9]	MySpace, Facebook, X	0.127	0.00170	0.314
Zhou *et al.,* [54]	Sina Weibo & QQ	-	0.05605	-
Yang *et al.,* [10]	X	0.382	0.04014	0.446
Bardi *et al.,* [14]	Instagram	-	0.56916	-

We also identified that the predefined interval set in our engine could have impacted its performance. In our experiment, we set this interval to 15 minutes. Our analysis showed that some bots, even those from the same group, did not interact with our social honeypots within this timeframe. This discrepancy may be because these bots operate on longer cycles to evade detection or follow different engagement schedules based on their programmed objectives [6].

Additionally, the engine faced challenges in identifying standalone bots that do not participate in any groups or campaigns. Our engine primarily relies on detecting synchronized behaviors of multiple bots through event and profile matching. Research indicates that standalone bots are particularly challenging to identify, as they do not display the same network behavior patterns as group-operated bots. Standalone bots often mimic legitimate user behavior more closely, complicating detection using traditional methods focused on group behaviors and interactions [57]. Going forward, the idea is to expand SohoNet's features, testing it in different contexts to gather additional insight and refine its engine and capabilities.

5. Conclusions

In this study, we introduced SohoNet, an innovative social honeynet framework designed to detect and analyze social bots in OSNs. Our work demonstrates the effectiveness of SohoNet in identifying bots by leveraging collective intelligence from multiple honeypots and the semi-automatic label engine. The results indicate that SohoNet successfully meets three key requirements: precision in capturing bots, efficiency in labeling them, and ensuring the safety of the online social network community involved.

While this study marks a significant advancement in the field of social honeypots and social bot detection, it is essential to acknowledge certain limitations. The focus on a specific case study, the Malaysian general election, may limit the generalizability of the findings to other contexts. Additionally, limitations in accessing the X API and the associated costs pose challenges for future research by restricting access to crucial data and limiting the ability to monitor bot behavior over time. Future research should explore applying SohoNet across various online social platforms to overcome these challenges.

We believe SohoNet makes a substantial contribution to the fields of social cybersecurity and online social network analysis. By merging innovative approaches with ethical considerations, this research aims to set a new standard for detecting and combating social bots, ultimately protecting the integrity and reliability of OSNs in the digital era.

Acknowledgement
This research was not funded by any grant.

References
[1] Ruffo, Giancarlo, Alfonso Semeraro, Anastasia Giachanou, and Paolo Rosso. "Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language." *Computer Science Review* 47 (2023): 100531. https://doi.org/10.1016/j.cosrev.2022.100531
[2] Rum, Siti Nurulain Mohd, Raihani Mohamed, and Auzi Asfarian. "Identifying political polarization in social media: A literature review." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 34, no. 1 (2024): 80-89. https://doi.org/10.37934/araset.34.1.8089
[3] Zhang, Yaozeng, Jing Ma, and Fanshu Fang. "How social bots can influence public opinion more effectively: Right connection strategy." *Physica A: Statistical Mechanics and its Applications* 633 (2024): 129386. https://doi.org/10.1016/j.physa.2023.129386
[4] Ghani, Miharaini Md, Wan Azani Wan Mustafa, Mohd Ekram Alhafis Hashim, Hafizul Fahri Hanafi, and Durratul Laquesha Shaiful Bakhtiar. "Impact of generative AI on communication patterns in social media." *Journal of Advanced Research in Computing and Applications* 26, no. 1 (2022): 22-34.
[5] Yang, Kai-Cheng, Onur Varol, Pik-Mai Hui, and Filippo Menczer. "Scalable and generalizable social bot detection through data selection." In *Proceedings of the AAAI conference on artificial intelligence*, 34, no. 01, p. 1096-1103. 2020. https://doi.org/10.1609/aaai.v34i01.5460
[6] Chavoshi, Nikan, Hossein Hamooni, and Abdullah Mueen. "On-demand bot detection and archival system." In *Proceedings of the 26th International Conference on World Wide Web Companion*, p. 183-187. 2017. https://doi.org/10.1145/3041021.3054733
[7] Webb, Steve, James Caverlee, and Calton Pu. "Social honeypots: Making friends with a spammer near you." In *CEAS*, p. 1-10. 2008.
[8] Lee, Kyumin, James Caverlee, and Steve Webb. "Uncovering social spammers: social honeypots+ machine learning." In *Proceedings of the 33rd international ACM SIGIR conference on Research and Development in Information Retrieval*, p. 435-442. 2010. https://doi.org/10.1145/1835449.1835522
[9] Stringhini, Gianluca, Christopher Kruegel, and Giovanni Vigna. "Detecting spammers on social networks." In *Proceedings of the 26th Annual Computer Security Applications Conference*, p. 1-9. 2010. https://doi.org/10.1145/1920261.1920263
[10] Yang, Chao, Jialong Zhang, and Guofei Gu. "A taste of tweets: Reverse engineering twitter spammers." In *Proceedings of the 30th Annual Computer Security Applications Conference*, p. 86-95. 2014. https://doi.org/10.1145/2664243.2664258
[11] Bindu, P. V., Rahul Mishra, and P. Santhi Thilagam. "Discovering spammer communities in twitter." *Journal of Intelligent Information Systems* 51 (2018): 503-527. https://doi.org/10.1007/s10844-017-0494-z
[12] The Honeynet Project. "The honeynet project workshop 2024." 2024.
[13] Telekom Security. "T-pot version 24.04 released." 2024.
[14] Bardi, Sara, Mauro Conti, Luca Pajola, and Pier Paolo Tricomi. "Social honeypot for humans: luring people through self-managed Instagram pages." In *International Conference on Applied Cryptography and Network Security*, p. 309-336. Cham: Springer Nature Switzerland, 2023. https://doi.org/10.1007/978-3-031-33488-7_12
[15] El Mendili, Fatna, Mohammed Fattah, Nisrine Berros, Youness Filaly, and Younès El Bouzekri El Idrissi. "Enhancing detection of malicious profiles and spam tweets with an automated honeypot framework powered by deep learning." *International Journal of Information Security* (2023): 1-30. https://doi.org/10.1007/s10207-023-00796-7
[16] Paradise, Abigail, Asaf Shabtai, Rami Puzis, Aviad Elyashar, Yuval Elovici, Mehran Roshandel, and Christoph Peylo. "Creation and management of social network honeypots for detecting targeted cyber attacks." *IEEE Transactions On Computational Social Systems* 4, no. 3 (2017): 65-79. https://doi.org/10.1109/TCSS.2017.2719705
[17] Abualhija, Mwaffaq, Nid Al-Shaf'i, Nidal M. Turab, and Abdelrahman Hussein. "Encountering social engineering activities with a novel honeypot mechanism." *International Journal of Electrical & Computer Engineering (2088-8708)* 13, no. 6 (2023). https://doi.org/10.11591/ijece.v13i6.pp7056-7064
[18] Kenny, Ryan, Baruch Fischhoff, Alex Davis, Kathleen M. Carley, and Casey Canfield. "Duped by bots: why some are better than others at detecting fake social media personas." *Human Factors* 66, no. 1 (2024): 88-102. https://doi.org/10.1177/00187208211072642
[19] Romero, Daniel, and Jon Kleinberg. "The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter." In *Proceedings of the International AAAI Conference on Web and Social Media*, 4, no. 1, p. 138-145. 2010. https://doi.org/10.1609/icwsm.v4i1.14015

[20] Aljabri, Malak, Rachid Zagrouba, Afrah Shaahid, Fatima Alnasser, Asalah Saleh, and Dorieh M. Alomari. "Machine learning-based social media bot detection: A comprehensive literature review." *Social Network Analysis and Mining* 13, no. 1 (2023): 20. https://doi.org/10.1007/s13278-022-01020-5
[21] Hayawi, Kadhim, Susmita Saha, Mohammad Mehedy Masud, Sujith Samuel Mathew, and Mohammed Kaosar. "Social media bot detection with deep learning methods: a systematic review." *Neural Computing and Applications* 35, no. 12 (2023): 8903-8918. https://doi.org/10.1007/s00521-023-08352-z
[22] Ferrara, Emilio. "Social bot detection in the age of ChatGPT: Challenges and opportunities." *First Monday* (2023). https://doi.org/10.5210/fm.v28i6.13185
[23] Kolomeets, Maxim, Olga Tushkanova, Vasily Desnitsky, Lidia Vitkova, and Andrey Chechulin. "Experimental evaluation: Can humans recognise social media bots?." *Big Data and Cognitive Computing* 8, no. 3 (2024): 24. https://doi.org/10.3390/bdcc8030024
[24] Pozzana, Iacopo, and Emilio Ferrara. "Measuring bot and human behavioral dynamics." *Frontiers in Physics* 8 (2020): 125. https://doi.org/10.3389/fphy.2020.00125
[25] Guo, Zhen, Jin-Hee Cho, Ray Chen, Srijan Sengupta, Michin Hong, and Tanushree Mitra. "Online social deception and its countermeasures: A survey." *IEEE Access* 9 (2020): 1770-1806. https://doi.org/10.1109/ACCESS.2020.3047337
[26] Chen, Xiujuan, Shanbing Gao, and Xue Zhang. "Visual analysis of global research trends in social bots based on bibliometrics." *Online Information Review* 46, no. 6 (2022): 1076-1094. https://doi.org/10.1108/OIR-06-2021-0336
[27] Shao, Chengcheng, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. "The spread of low-credibility content by social bots." *Nature Communications* 9, no. 1 (2018): 1-9. https://doi.org/10.1038/s41467-018-06930-7
[28] Wright, William, David Schroh, Pascale Proulx, Alex Skaburskis, and Brian Cort. "The Sandbox for analysis: concepts and methods." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 801-810. 2006. https://doi.org/10.1145/1124772.1124890
[29] Cresci, Stefano. "Detecting malicious social bots: story of a never-ending clash." In *Disinformation in Open Online Media: First Multidisciplinary International Symposium, MISDOOM 2019, Hamburg, Germany, February 27–March 1, 2019, Revised Selected Papers 1*, p. 77-88. Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-39627-5_7
[30] Umbrani, Kunal, Deven Shah, Amit Pile, and Anamika Jain. "Fake Profile Detection Using Machine Learning." In *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, p. 966-973. IEEE, 2024. https://doi.org/10.1109/ICETSIS61505.2024.10459570
[31] Freitas, Carlos, Fabrício Benevenuto, Adriano Veloso, and Saptarshi Ghosh. "An empirical study of socialbot infiltration strategies in the Twitter social network." *Social Network Analysis and Mining* 6 (2016): 1-16. https://doi.org/10.1007/s13278-016-0331-3
[32] Zhang, Jinxue, Rui Zhang, Yanchao Zhang, and Guanhua Yan. "The rise of social botnets: Attacks and countermeasures." *IEEE Transactions on Dependable and Secure Computing* 15, no. 6 (2016): 1068-1082. https://doi.org/10.1109/TDSC.2016.2641441
[33] Boshmaf, Yazan, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. "Design and analysis of a social botnet." *Computer Networks* 57, no. 2 (2013): 556-578. https://doi.org/10.1016/j.comnet.2012.06.006
[34] Hasan, Md Ahsan Ul, Azuraliza Abu Bakar, and Mohd Ridzwan Yaakub. "Detecting community through user similarity analysis on Twitter." In *2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, p. 1-6. IEEE, 2024. https://doi.org/10.1109/IMCOM60618.2024.10418381
[35] Chavoshi, Nikan, Hossein Hamooni, and Abdullah Mueen. "Debot: Twitter bot detection via warped correlation." In *Icdm*, 18, p. 28-65. 2016. https://doi.org/10.1109/ICDM.2016.0096
[36] Cresci, Stefano. "A decade of social bot detection." *Communications of the ACM* 63, no. 10 (2020): 72-83. https://doi.org/10.1145/3409116
[37] Ahmed, Faraz, and Muhammad Abulaish. "An mcl-based approach for spam profile detection in online social networks." In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, p. 602-608. IEEE, 2012. https://doi.org/10.1109/TrustCom.2012.83
[38] Lee, Kyumin, James Caverlee, Zhiyuan Cheng, and Daniel Z. Sui. "Campaign Extraction from Social Media." *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, no. 1 (2014): 1-28. https://doi.org/10.1145/2542182.2542191
[39] Khalil, Hunia, Muhammad US Khan, and Mazhar Ali. "Feature selection for unsupervised bot detection." In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, p. 1-7. IEEE, 2020. https://doi.org/10.1109/iCoMET48670.2020.9074131

[40] Zhou, Lu, Wenbo Wang, and Keke Chen. "Tweet properly: Analyzing deleted tweets to understand and identify regrettable ones." In *Proceedings of the 25th International Conference on World Wide Web*, p. 603-612. 2016. https://doi.org/10.1145/2872427.2883052
[41] Gwet, Kilem L. *Handbook of inter-rater reliability: The Definitive Guide to Measuring the Extent of Agreement Among Raters*. Advanced Analytics, LLC, 2014.
[42] Elovici, Yuval, Michael Fire, Amir Herzberg, and Haya Shulman. "Ethical considerations when employing fake identities in online social networks for research." *Science and Engineering Ethics* 20 (2014): 1027-1043. https://doi.org/10.1007/s11948-013-9473-0
[43] Dittrich, David. "The ethics of social honeypots." *Research Ethics* 11, no. 4 (2015): 192-210. https://doi.org/10.1177/1747016115583380
[44] de Lima Salge, Carolina Alves, and Nicholas Berente. "Is that social bot behaving unethically?." *Communications of the ACM* 60, no. 9 (2017): 29-31. https://doi.org/10.1145/3126492
[45] Rowe, Neil C. "The ethics of deception in cyberspace." In *Handbook of Research on Technoethics*, p. 529-541. IGI Global, 2009. https://doi.org/10.4018/978-1-60566-022-6.ch034
[46] Alosefer, Yaser, and Omer Rana. "Honeyware: a web-based low interaction client honeypot." In *2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, p. 410-417. IEEE, 2010. https://doi.org/10.1109/ICSTW.2010.41
[47] Anwar, Ahmed H., Charles A. Kamhoua, Nandi O. Leslie, and Christopher Kiekintveld. "Honeypot allocation for cyber deception under uncertainty." *IEEE Transactions on Network and Service Management* 19, no. 3 (2022): 3438-3452. https://doi.org/10.1109/TNSM.2022.3179965
[48] Javadpour, Amir, Forough Ja'fari, Tarik Taleb, Mohammad Shojafar, and Chafika Benzaïd. "A comprehensive survey on cyber deception techniques to improve honeypot performance." *Computers & Security* (2024): 103792. https://doi.org/10.1016/j.cose.2024.103792
[49] Ong, Yew Chuan. "Characterising and Detecting Social Bots." PhD diss., University of Sheffield, 2020.
[50] Martini, Franziska, Paul Samula, Tobias R. Keller, and Ulrike Klinger. "Bot, or not? Comparing three methods for detecting social bots in five political discourses." *Big Data & Society* 8, no. 2 (2021): 20539517211033566. https://doi.org/10.1177/20539517211033566
[51] Chu, Zi, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. "Detecting automation of twitter accounts: Are you a human, bot, or cyborg?." *IEEE Transactions on Dependable and Secure Computing* 9, no. 6 (2012): 811-824. https://doi.org/10.1109/TDSC.2012.75
[52] Lee, Kyumin, Brian Eoff, and James Caverlee. "Devils, angels, and robots: Tempting destructive users in social media." In *Proceedings of the International AAAI Conference on Web and Social Media*, 4, no. 1, p. 275-278. 2010. https://doi.org/10.1609/icwsm.v4i1.14044
[53] Lee, Kyumin, Brian Eoff, and James Caverlee. "Seven months with the devils: A long-term study of content polluters on twitter." In *Proceedings of the International AAAI Conference on Web and Social Media* 5, no. 1, p. 185-192. 2011. https://doi.org/10.1609/icwsm.v5i1.14106
[54] Zhou, Yi, Kai Chen, Li Song, Xiaokang Yang, and Jianhua He. "Feature analysis of spammers in social networks with active honeypots: A case study of chinese microblogging networks." In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, p. 728-729. IEEE, 2012. https://doi.org/10.1109/ASONAM.2012.133
[55] Ghosh, Saptarshi, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. "Understanding and combating link farming in the twitter social network." In *Proceedings of the 21st International Conference on World Wide Web*, p. 61-70. 2012. https://doi.org/10.1145/2187836.2187846
[56] Cresci, Stefano, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race." In *Proceedings of the 26th International Conference on World Wide Web Companion*, p. 963-972. 2017. https://doi.org/10.1145/3041021.3055135
[57] Zeng, Ziming, Tingting Li, Jingjing Sun, Shouqiang Sun, and Yu Zhang. "Research on the generalization of social bot detection from two dimensions: feature extraction and detection approaches." *Data Technologies and Applications* 57, no. 2 (2023): 177-198. https://doi.org/10.1108/DTA-02-2022-0084