Detecting *In Situ* Identity Fraud on Social Network Services: A Case Study With Facebook

Shan-Hung Wu, Man-Ju Chou, Chun-Hsiung Tseng, Yuh-Jye Lee, and Kuan-Ta Chen, *Senior Member, IEEE*

***Abstract*—With the growing popularity of Social Networking Services (SNSs), increasing amounts of sensitive information are stored online and linked to SNS accounts. The obvious value of SNS accounts gives rise to the *identity fraud* problem—unauthorized, stealthy use of SNS accounts. For example, anxious parents may use their children's SNS accounts to spy on the children's social interaction; or husbands/wives may check their spouses' SNS accounts if they suspect infidelity. Stealthy identity fraud could happen to anyone and seriously invade the privacy of account owners. However, there is no known defense against such behavior when an attacker, possibly an acquaintance of the victim, gets access to the victim's computing devices. In this paper, we propose to extend the use of continuous authentication to detect the *in situ identity fraud incidents*, which occurs when the attackers use *the same accounts, the same devices, and IP addresses* as the victims. Using Facebook as a case study, we show that it is possible to detect such incidents by analyzing SNS users' browsing behavior. Our experiment results demonstrate that the approach can achieve higher than 80% detection accuracy within 2 min, and over 90% after 7 min of observation time.**

***Index Terms*—Authentication, continuous authentication, data privacy, fraud, information security, internet privacy, social computing, social security.**

I. Introduction

MANY PEOPLE use Social Networking Services (SNSs) daily, and link a lot of personal and sensitive information to their SNS accounts. The information generally includes friend lists, feeds from friends, nonpublic posts/photos, private interactions with acquaintances (such as chats and messages), and purchased apps/items. The obvious value of such information makes SNS accounts one of the most targeted online resources by malicious attackers. SNS sites have made significant efforts to prevent identity fraud and protect users' privacy. For example, Facebook records the regular IP addresses and devices used by each account. If an unusual IP address or device is used to log in to an account, the user may be required to answer some secret questions [1] or enter a security code sent to the account owner's mobile device [2] in order to verify if the login is authentic. Facebook also allows users to report account theft manually if they suspect that their accounts have been compromised.

Despite all the efforts to prevent identity fraud, user privacy can be compromised by another form of breach called *in situ identity fraud*—unauthorized, stealthy use of SNS accounts by attackers using the same device and network connection as the account owners. Different from other forms of identify fraud, anyone can perform *in situ* identity fraud without any technology hacks. For example, anxious parents may use their children's SNS accounts to spy on the children's social interactions; or husbands/wives may check their spouses' SNS accounts if they suspect infidelity. Similarly, colleagues, supervisors, friends, and siblings may use acquaintances' accounts for different reasons when there is a chance.

In situ identity fraud is widespread for a number of reasons. First, people tend to choose "YES" when the browsers on their own computers ask if they want to save their (SNS) passwords for automatic logins in the future. This is especially true when people use their mobile devices because inputting passwords is inconvenient [3], [4]. Mobile devices make *in situ* identity fraud easy in other ways, as they can be physically accessed by acquaintances or strangers [5], and most of them are not locked by PINs [6]. In addition, many SNS sites use cookies to avoid the need for account authentication within a short period of time. For example, once logged into Facebook, a user does not need to log in again for up to 60 days [7]. Given the above loopholes, if someone (usually an acquaintance) gets to access an SNS user's computer or mobile device, it is unlikely that he will need a technical background to obtain the information associated with the SNS account.

It is very difficult to detect *in situ* identity fraud of SNS accounts by traditional methods because attackers use the same IP addresses/devices and provide the same credentials to access the owners' accounts. Moreover, because users do not normally check the login logs, they cannot detect and report such incidents unless the attackers leave clear evidence. Thus, there is no defense against *in situ* identity fraud.

A. Contributions

In this paper, we investigate the *in situ* identity fraud problem on SNSs, and propose a scheme that analyzes users' browsing behavior to detect such incidents. Using Facebook as a case study, we show that it is possible to detect this type of attacks on SNS sites by analyzing users' browsing behavior, such as clicks on newsfeeds,[1] friend lists, profiles, likes, messages, photos/

Manuscript received October 26, 2014; revised October 08, 2015; accepted November 05, 2015. Date of publication December 22, 2015; date of current version November 22, 2017. This work was supported in part by the Ministry of Science and Technology under Grant MOST 103-2911-I-002-001, in part by the National Taiwan University under Grant NTU-ICRP-104R7501, and in part by Intel Corporation under Grant NTU-ICRP-104R7501-1.

S.-H. Wu is with the Department of Computer Science, National Tsing Hua University, Taipei, Taiwan (e-mail: shwu@cs.nthu.edu.tw).

M.-J. Chou and Y.-J. Lee are with the Department of Computer Science, National Taiwan University of Science and Technology, Taipei, Taiwan.

C.-H. Tseng is with the Department of Information Management, Nanhua University, Chiayi, Taiwan.

K.-T. Chen is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan.

Digital Object Identifier 10.1109/JSYST.2015.2504102

[1] A user's Facebook newsfeed is located in the middle column of his Facebook page. It is a constantly updated list that summarizes the status of people the user follows on Facebook.

videos, and comments. The results demonstrate that the proposed scheme can achieve more than 80% accuracy with a high degree of confidence within 2 min, and over 90% after 7 min of observation time.

B. Deployment

The proposed detection approach is designed to run on SNS servers and act as the first line of defense against *in situ* identity fraud. The deployment is straightforward because an SNS server simply collects information about the behavior of an account's session and feeds it into a detection model in real time. The model determines whether the current user is authentic. If there is any suspicion about the user, the SNS server can 1) apply more sophisticated analysis/monitoring techniques and/or 2) challenge the current user immediately by asking secret questions or via a second channel, such as mobile phone authentication [2]. Since this is the first line of defense, the detection model does not need to be 100% accurate. Reasonable detection power is sufficient and the key issue is to identify account fraud incidents as early as possible. Moreover, the proposed method is not limited to a specific SNS site or a certain learning technique because it is based on a standard supervised learning framework, such as the smooth SVM [8] adopted in this work. Thus, service operators like Facebook may choose the asymmetric SVM [9] or any other learning framework if they wish to further fine tune the identification performance.

C. Implications

We believe that the *in situ* identity fraud problem, which has not been widely studied, will become more critical in the future as people store increasing amounts of sensitive information online. In fact, the problem may also occur in email services such as Gmail and Outlook; time management services such as Google Calendar and Remember The Milk; and photo album services such as Instagram. Asking users to authenticate themselves repeatedly during the use of the services is infeasible in practice due to usability issues. Thus, implicit detection seems to be a reasonable way to prevent attacks of this kind.

This paper is organized as follows. Section II provides a review of related works; and Section III explains the rationale behind the proposed method, which exploits users' browsing behavior to detect *in situ* identity fraud. In Section IV, we discuss the user study based on Facebook and analyze users' behavior. We describe our detection methodology in Section V, evaluate the scheme's performance in Section VI, and consider security issues in Section VII. Section VIII contains some concluding remarks.

II. Related Work

In this section, we review existing studies on the privacy issues related to SNSs and intrusion/misuse detection.

A. SNS Privacy

A great deal of effort has been devoted to protecting users' privacy, which is always a concern for SNS users. He *et al.* [10], Zheleva and Getoor [11], and Tang *et al.* [12] observed a privacy loophole that allows attackers to infer private information about a user (such as sexual orientation) from his public SNS records/activities. In [13], Bilge *et al.* studied the feasibility of social engineering attacks over SNS, whereas the authors in [14] proposed a detection scheme against identity clone attacks that aim at creating fake identifies for malicious purposes. Felt and Evans [15] and Wishart *et al.* [16] developed methods that prevent privacy leaks from SNS developers' APIs and the software built based on them. Mahmood and Desmedt [17] identified on a type of privacy attack called frequent account deactivation. Meanwhile, in the SNS industry, Facebook uses a number of measures to protect users' privacy. For example, it provides an official page [18] to educate users about the recommended privacy and security settings, and it records the IP addresses, web browsers, and devices used by each account [19]. If an attempt is made to log into an account with unseen IP addresses or devices, Facebook challenges the user by asking secret questions [1] or via mobile phone authentication [2]. Users can also report suspicious identity fraud incidents manually.

However, none of the above measures can protect users' privacy from *in situ* identity fraud, i.e., if the attackers use the *identical devices* as the victims. Passwords, credentials, and cookies are usually stored in users' devices to avoid the need for repeated account authentication [3], [4], [7]. Thus, attackers who have physical access to the devices can easily bypass existing detection schemes and obtain sensitive information in users' SNS accounts.

B. Anomaly Detection and Misuse Detection

To our knowledge, the research on misuse detection can be traced back to Lunt *et al.* [20]. In [20], the authors described a system that monitors account logins, logouts, program executions, system calls, network activity, and so on in order to detect the misuse of an account by others. Similar strategies have been applied to the misuse of general computer systems [21]–[23], information retrieval systems [24], database systems [25], [26], transaction systems [27], email accounts [28], and so on. Furthermore, to enable real-time detection of misuse, continuous authentication (also known as reauthentication) [29] was commonly adopted to analyzes the activity of a system or a user to continuously verify the genuineness of the usage. This approach analyzes activities such as keyboard typing behavior [29], mouse movements [30], touch gestures on mobile devices [31], facial characteristics (if a webcam is available) [32], or any other soft biometric traits [33], [34].

The research done by Egele *et al.* [35] is probably the most similar to this paper. In the paper, the authors proposed a scheme to detect compromised accounts on online social networks and used Twitter to validate their approach. The key differences between [35] and the current paper are as follows: 1) the former requires relatively long-term observations, say, a few days or even longer time and 2) the compromised accounts are often used to post advertisement messages, whereas in our scenario (i.e., in-situ identity fraud), the stalker normally browse information without leaving trails. Thus, [35] cannot be applied to solve the *in situ* identity fraud problem.

To the best of our knowledge, this paper is the first to address the *in situ* identity fraud problem, a special case of account misuse, on SNS sites. As described in Section I, the growing popularity of SNSs and the increasing amounts of sensitive information online suggest the immediate demand for identity fraud detection for SNSs. Earlier proposals on account misuse detection cannot be applied to this scenario because 1) they are either based on system-level activities such as processes and network traffic, which are not informative indicators of *in situ* identity fraud as the same device and network connections are used or 2) they are based on domain-specific activities such as database queries and information search queries, which are clearly inapplicable to the context of SNS user behavior. Moreover, in many account misuse detection schemes, a detection model is required for each user. The cost may be prohibitive for SNS servers because an SNS site may have millions of users.[2] The scheme proposed in this paper only analyzes the Web browsing behavior of *three predefined user groups*. The detection model is universal, i.e., it can be applied to *all users*, and it incurs low data collection and computation overheads. Another advantage is that the scheme can be applied to a new account whose associated behavior is not yet clear. Note that the scheme is not a replacement for existing continuous authentication approaches. Rather, it can serve a low-cost filter for suspicious accounts, so that servers can apply more sophisticated, personalized analytical techniques when necessary.

III. Rationale Behind Our Approach

SNSs are not simply places for people to maintain their friend lists. They are more like platforms where people can engage various social activities, such as posting details of their own status, reading other users' comments on the news, chatting, and meeting new people. Some studies [36], [37] suggest that there is *no typical* user behavior pattern on a complicated, open platform like Facebook, as every user seems to behave differently on an SNS. For example, some people use SNS to satisfy their desire for self-promotion, so they spend most of their time sharing the latest information about their status and posting the latest photos/events. On the other hand, some people may want to make new friends online, chat with old friends, or spend time discovering new social games, while some may want to stalk certain other users.

Despite that users are driven by different personal intentions when they use SNS, we posit that SNS users would behave quite differently when they browse *SNS information of others* rather than those of their own, and we conjecture the resulting behavioral difference would be significant regardless of individual diversities. In other words, personal differences does certainly exist, but we consider that SNS users would exhibit relatively more or less similarly when they have opportunities to peep *SNS information of others*, such as the newsfeed of their friends on Facebook.

In the context of *in situ* identity fraud, an SNS user can be classified as one of the following roles: 1) an *owner*, which means the person uses his own account; 2) an *acquaintance* (as a stalker), who uses the account of someone he knows; or 3) a *stranger* (as a stalker), who uses the account of a person he does not know. Intuitively, when owners check their Facebook newsfeeds, they should focus more on the latest information posted by friends and use the "Like" or "Share" function to interact with others. By contrast, when a stalker (either an acquaintance or a stranger) browses a newsfeed, he may be more interested in historical information about the stalker and/or the account holder. Also, the stalker generally do not interact with others to avoid discovery by the account holder about the identity fraud. In summary, we believe that users' behavior varies in different roles for the following reasons.

1) The way people treat *familiar* information (and information from close friends) would be different than the way they treat *unfamiliar* information.
2) People in different roles may have different intentions.
3) To avoid detection by the account owners, stalkers are supposed not to make any interaction with others. Also, they may have limited time to commit in-situ identity fraud so their browsing behavior would be even more different.

We define the above differences in SNS users' browsing behavior as *role-driven behavioral diversity*, which serves the rationale behind the proposed detection scheme. In the next section, using a dataset collected on Facebook, we demonstrate the significance of the role-driven behavioral diversity. Then, in Section V, we explain how the detection scheme exploits this property to distinguish between account owners and stalkers.

IV. Browsing Behavior on Facebook

As mentioned earlier, we use Facebook as a case study of SNS users' role-driven behavior.

A. Data Collection

To capture the role-driven behavioral diversity of users, we asked a number of Facebook account owners to participate in experiments that involved browsing Facebook newsfeeds in different roles. Specifically, each subject browsed 1) his own newsfeed; 2) the newsfeed of a real-world acquaintance; and 3) the newsfeed of a stranger.

For the experiments, we hired *pairs* of subjects from an Internet community with more than one million members, where each pair satisfied at least one of the following relationship criteria: friends, family members, colleagues, classmates, or couples. The subjects were paid 10 USD each and gave permission for their actions (e.g., clicks, typing, and page views) to be recorded when they browsed newsfeeds. In addition, the subjects were only hired if they have been active Facebook users with over 50 friends and regularly using Facebook for more than 4 h per week.

Each experiment comprised three rounds. In each round, a subject was asked to browse the newsfeed of his account, a friend's account, or a stranger's account for 30 min. The subjects and accounts were paired *randomly*, but each subject was guaranteed to play all three roles in the three rounds, as

[2]For example, Facebook had more than one billion active users in December 2012.

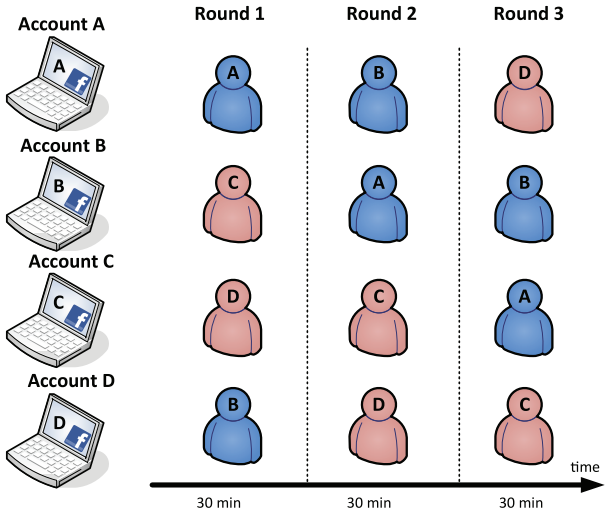


Fig. 1. (A, B) and (C, D) are pairs of acquaintances. Each experiment comprises three rounds. In each round, each subject is assigned an account *at random*. In the three rounds, a subject is guaranteed to browse his own account, an acquaintance's account, and a stranger's account in a randomized order.

TABLE I
SUMMARY OF THE EXPERIMENTS AND THE RAW DATASET

Property	Value
# Experiments	28
Trace length	9 302 min (155 h)
# Subjects	112
Male	56
Female	44
# Sessions	278
By owners	100
By acquaintances	81
By strangers	97
Avg. session length	33 min
Avg. action rate	3.0 action/min
Avg. page switching rate	0.7 page/min

shown in Fig. 1. During each round, the subjects could perform any action they wished (e.g., browsing photos or leaving comments), but they could not engage in sabotage activities (e.g., changing an account's password). Actions that followed external links (such as videos) were allowed, but the subjects were instructed not to browse the external content for longer than 1 min each time.

To capture the subjects' activities on Facebook, we used Fiddler, a Web debugging proxy, to monitor all the HTTP/HTTPS GET/POST requests issued by the Web browser. By parsing the HTTP/HTTPS request logs, we were able to capture every action performed by a user (as described later). Table I summarizes the experiments and the raw dataset we collected. Although our administrators monitored the experiments, some subjects did not follow the instruction to browse the specified Facebook newsfeeds. Out of 311 sessions, we removed 33 "noisy sessions" of subjects who obviously did focus on the newsfeeds, i.e., sessions with an idle or inactive period longer than 5 min. As a result, our 112 subjects (56 pairs of acquaintances) totally performed 278 Facebook browsing sessions. The trace contains 9302 min (155 h) and approximately 27 000 browsing actions.

TABLE II
18 COMMON USER ACTIONS WE COLLECTED FROM THE EXPERIMENT ON FACEBOOK

Actions	Account-relevant	Page-switching
Likes	√	
View cards	√	
View likes	√	
View messages	√	
View photos	√	
To friend list page	√	√
To note page	√	√
To photo page	√	√
To wall page	√	√
To fan page		√
To feed page		√
To group page		√
To message page		√
Add comments	√	
Delete comments	√	
Click hyper-links		
Expand comments	√	
Expand page		

TABLE III
EXAMPLES OF USER ACTIONS COLLECTED FROM FACEBOOK

Time stamp	Action	Target person
1345837539249.47	Likes	Friend A
1345837568519.15	View cards	Account owner
1345837586398.26	Add comment	Friend A
1345837732512.73	Group page	
1345837756445.03	Likes	Friend B
1345837770260.55	View cards	Nonfriend C
1345837773293.04	View message	Friend A
1345837828598.01	Likes	Nonfriend C
1345837875240.45	Expand page	

B. Defining Features

We identified 18 common user actions on Facebook, as shown in Table II. Each of the actions can be associated to two attributes: 1) *account-relevant action*, which refers to an action used to interact with another person (account) and 2) *page-switching action*, which causes the browser to load another Facebook page. By parsing the HTTP/HTTPS request logs, we obtained a chronological list of actions for each session, as shown by the examples in Table III. Each record on the list contains the name of the action, the occurrence time stamp, and the target person the user interacted with if the action was interactive. Based on the account owner's profile and friend list, we annotated the target person as either the "account owner," a "friend," or a "nonfriend."

Next, we define a number of features and extract their values for each session we collected in order to characterize users' browsing behavior and apply machine learning algorithms for predicting *in situ* identity fraud incidents. Even if there is a perfect learning algorithm, without features that encode information about who is initiating a session, the algorithm will not be able to distinguish between account owners and stalkers. How to define discriminative features is a key issue in this work, and it is usually challenging because it requires insight, domain knowledge, creativity, and even "black arts" [38].

We interviewed heavy users of Facebook about their regular usage patterns and the ways they discovered and explored interesting information. Based on the results, we defined 139 features. All the features of a particular session can be extracted

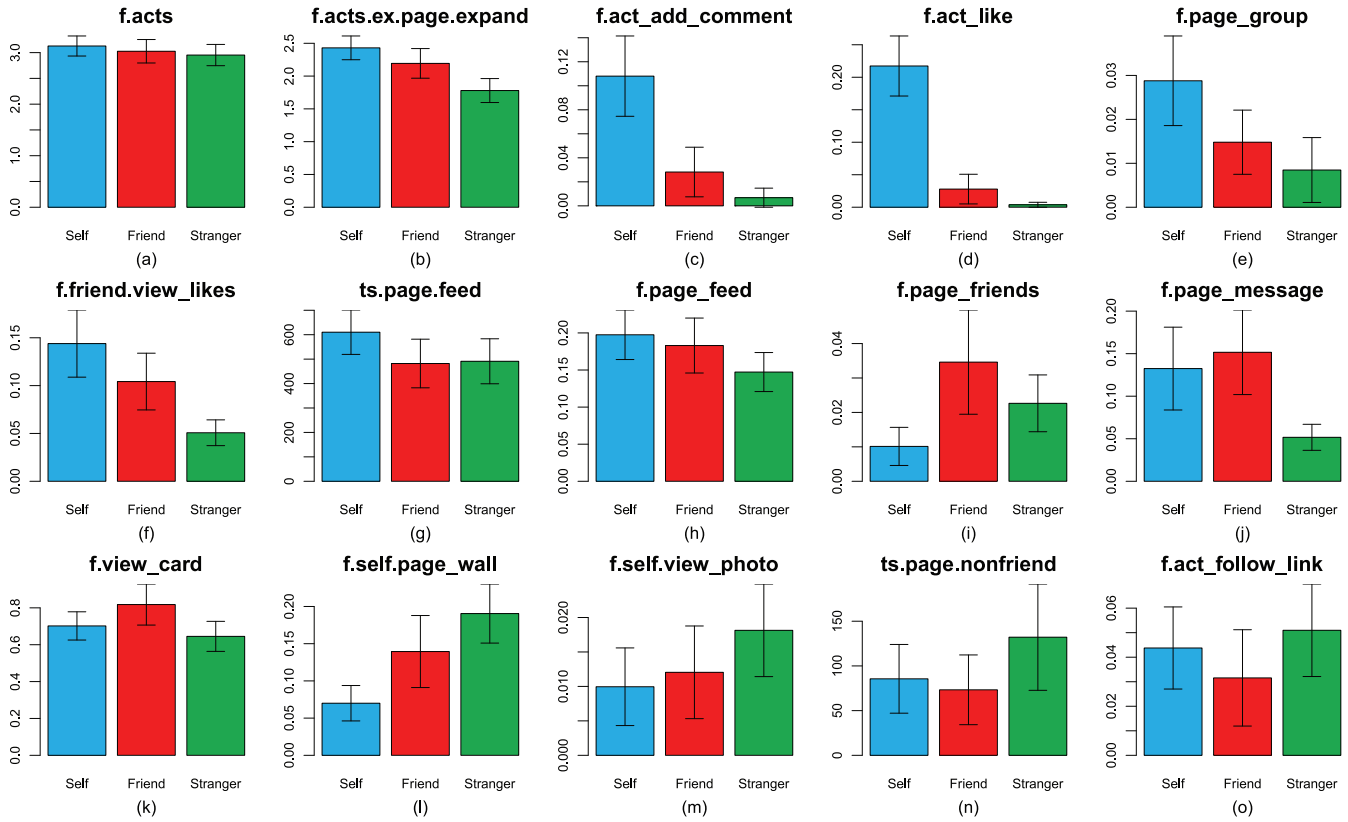


Fig. 2. Evidence of role-driven behavioral diversity.

from the session's action list (see Table III). We summarize the features as follows.

1) `f.<action>`: The frequency of a certain action (per minute), where `<action>` refers to an action defined in Table II. Meanwhile, `f.acts` denotes the frequency of all the user actions by dividing the number of actions by the session duration. In addition, we also define `f.acts.excluding.page.expand` to represent the frequency of all but the "expand page" action.[3] This feature allows us to determine how fast a user perform actions in addition to browsing information.
2) `f.<target_type>.<action>`: The frequency of a certain action that targets a certain user type. The `<action>` is an account-relevant action in Table II, and `<target_type>` can be `self` (if the target person is the account owner), `friend` (if the target person is a friend of the account owner), or `nonfriend` (if the target person is not a friend).
3) `b.<xxx>`: The binary version of all the above features; i.e., `b.<xxx>=1` iff `f.<xxx>` is greater than 0. For example, `b.<action>` indicates if a certain action occurs during the session.
4) `f.act.<target_type>`: The frequency of all the account-relevant actions performed in relation to a certain target user type.
5) `ts.page.<page_type>`: The time the session user spends on a certain page after performing a page-switching action. The `<page_type>` can be `feed` (the account owner's newsfeed), msg (the account owner's message box), `self` (pages, such as the wall/friend list/note/photos, of the account owner), `friend` (pages of friends), `nonfriend` (pages of nonfriends), or `public` (pages belonging to fans or groups).
6) `f.act.page.<page_type>`: The frequency of all the actions the users perform on a certain page type. We also define `f.act.expand.page.<page_type>` and `f.act.non.expand.page.<page_type>` to take account of the frequency of the "expand page" action and that of the rest actions on a certain page type, respectively.
7) `n.act.person`: The number of target people the user interacts with (via an account-relevant action) during the session.
8) `n.act.person.<statistics>`: The statistics of the number of visits made to different users' pages during the session. The `<statistics>` include the `mean`, `std_dev`, `median`, and `maximum`. For example, if the user visits the account owner's pages once, friend A's pages thrice, friend B's pages once, and nonfriend C's pages twice, we obtain `mean` = 1.75, `std_dev` = 0.96, `median` = 1.5, and `maximum` = 3. We capture these features because we want to determine if a user focuses on specific person(s).

After extracting the features for each session, we obtain a dataset for further analysis. Each session is labeled as either

[3]In Facebook, some pages (e.g., newsfeeds and walls) and page items (e.g., comments, notifications lists, etc.) can be expanded to show earlier/more information via an "expand page" action.

"owner," "acquaintance," or "stranger," depending on the user's role during the session in the experiment.

C. Role-Driven Behavioral Diversity

To verify the existence of role-driven behavioral diversity between account owners, acquaintances, and strangers, we analyze users' behavior patterns in different roles. Our observations are summarized below.

1) General Diversity: As shown in Fig. 2(a), all the sessions for the three user roles have similar values in `f.acts`. However, in `f.acts.excluding.page.expand` [Fig. 2(b)], the sessions controlled by account owners have the highest values followed by those of acquaintances, and sessions controlled by strangers have the lowest values. This implies that acquaintances and strangers usually pay more attention to reading/searching for interesting information and care more about historical information, as the content hidden by expandable pages/items is older in terms of posting time.

The sessions used by acquaintances/strangers also yield much lower values in `f.act_add_comment` [Fig. 2(c)] and `f.act_like` [Fig. 2(d)] than those controlled by account owners. The reason is obvious: normally, acquaintances and strangers do not want to leave clues about their prying behavior.

2) What Stalkers Do Not Care About: Although acquaintances/ strangers expand pages more frequently [Fig. 2(b)], they do not expand the comment lists as often as account owners. This is because they may not know the people who left the comments, and therefore show less interest in them unless the comments are relevant to people they know. For similar reasons, acquaintances/strangers show less interest in the pages of fans and groups [Fig. 2(e)] and people who like a post [Fig. 2(f)]. They also spend relatively less time on the accounts' newsfeeds [Fig. 2(g)] but spend more time in checking the account owners' personal wall and photos [Fig. 2(l) and (m)].

3) What Acquainted Stalkers Care About: Among the three roles, acquaintances pay the most attention to the friend lists of the account owners [Fig. 2(i)]. This is because an acquaintance may be interested to know the account owner's social connections, especially people who are not friends of the acquainted stalker. For similar reasons, acquaintances generally show the most interest in the message boxes [Fig. 2(j)] and profile cards[4] of the accounts' friends [Fig. 2(k)].

4) What Stranger Stalkers Care About: Interestingly, strangers view account owners' profiles [Fig. 2(l)] and photos [Fig. 2(m)] more often than the account owners and their friends. The reason is that strangers do not know the account owners, so they are usually curious about the owners and how they look like. Stranger stalkers' actions are also less relevant to account owners' social relationships, as shown by the fact that they are more interested in nonfriends [Fig. 2(n)] and external links [Fig. 2(o)].

We believe that the above observations manifest the prominence of role-driven behavioral diversity. Next, we show how this diversity can be exploited to build a low-cost model for detecting *in situ* identity fraud on social network services.

[4]A profile card on Facebook provides a badge-like summarization of an account's basic information.

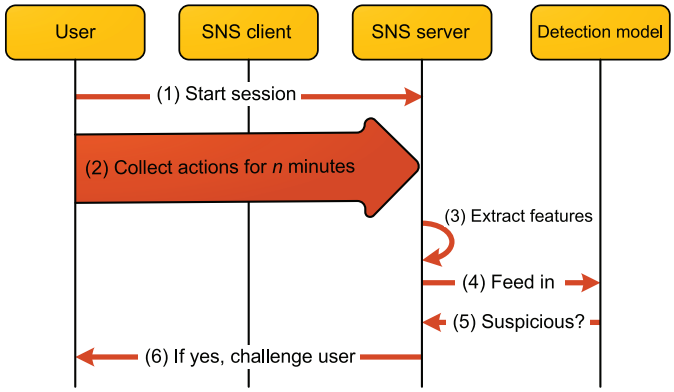


Fig. 3. Flowchart of the detection scheme.

V. Detection Scheme

In this section, we describe the scheme for detecting *in situ* identity fraud on SNS sites. Recall that each session in our dataset is labeled as either "account owner," "acquaintance," or "stranger." Because our goal is to distinguish stalkers from account owners, in the following, we replace the "acquaintance" and "stranger" labels with "stalker."

Fig. 3 provides an overview of the detection scheme. After a user logs in with a stored credential or existing authentication cookies (Step 1), the SNS server monitors and records the user's actions for an *observation period* of n minutes, where n is a configurable parameter (Step 2). At the end of the observation period, the server extracts the features of the monitored session based on the recorded actions (Step 3). It then feeds the session features, as defined in Section IV-B, into a detection model (Step 4), which determines if the session owner is suspicious by predicting the label of the session (Step 5). If the predicted label is "stalker," the SNS server can challenge the user by asking secret questions or via a second channel, such as the account owner's mobile phone (Step 6). Alternatively, the server can implement a more sophisticated, but costly, detection scheme.

For the server, the runtime cost of the scheme is low because it exploits the role-driven behavioral diversity property. As a result, only *one* detection model is needed for all SNS users. Note that although we utilize a two-class detection model to distinguish stalkers from account owners, the scheme can be easily extended to identify account owners, acquaintances, and strangers in a multiclass detection model. We train the detection model with the labeled sessions collected earlier. Clearly, the effectiveness of the detection scheme depends to a large extent on the quality of the predictions made by the detection model. Thus, to obtain high-quality predictions, we take the following rigorous steps to train the model.

A. Model Development

To facilitate the following description, we label the sessions browsed by stalkers and account owners as 1 and -1, respectively. In a training dataset D of size n, $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a labeled instance (i.e., session) with d features and $y_i \in \{1, -1\}$ is the

corresponding label. Our objective is to obtain a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ such that given a new instance $\mathbf{x}'$ with an unknown label y', we have $f(\mathbf{x}') > 0$ iff $y' = 1$, where the function f denotes the detection model in our scheme. In Step 4 in Fig. 3, the SNS server feeds the session $\mathbf{x}'$ into f; then, in Step 5, the SNS server gets $f(\mathbf{x}')$ and determines whether the session is suspicious or not by computing $\text{sgn}(f(\mathbf{x}'))$.

To obtain f, we use the support vector machine (SVM), a widely used machine learning algorithm for binary classifications. The objective functions of conventional SVMs (either linear or nonlinear) can be solved by standard quadratic programming algorithms. However, when applied to scenarios like we have for now, the solver may need to deal with an extremely large D due to the huge user base of the SNS service. To speed up the training process, we adopt smooth SVM (SSVM) [8], which is a variant of SVM, instead. SSVM adds $b^2/2$ to the objective of SVM and exploits the square of the slacks ξ_i^2 to penalize the noises and outliers. It utilizes the Karush–Kuhn–Tucker (KKT) optimization condition to convert the conventional SVM to an unconstrained minimization problem which can be solved efficiently by the Newton's method with an Armijo step size. The kernel trick also applies to SSVM as well. Here we pair up the nonlinear SSVM with the RBF kernel, which is defined as $K(\mathbf{a}, \mathbf{b}) = e^{-\gamma\|\mathbf{a}-\mathbf{b}\|_2^2}$. There are two hyper-parameters we have to determine in the nonlinear SSVM: the penalty coefficient C and γ in the RBF kernel function. We use the two-stage uniform design model selection method [39] with 13 runs and 9 runs in the first and second stages, respectively (according to [39]), to search for the best combination of both hyper-parameters.

B. Feature Selection

The training of SSVM is preceded by a feature-selection process [40] that only selects a subset of features in D for training. The process is important for three reasons: 1) Given the large number of sessions that SNS servers must monitor (Step 1 in Fig. 3), a small set of features helps the SSVM *scale up in making predictions*. 2) The selected features help us determine the actions that are useful in distinguishing stalkers from account owners. By ignoring the features that are not helpful, we can collect fewer actions (Step 2 in Fig. 3) and save the cost of feature extraction (Step 3 in Fig. 3) on the servers. 3) The process improves the prediction accuracy of the final SSVM.

The feature-selection process is divided into two stages, as shown in Fig. 4. In the first stage, we use the 1-norm SVM [41] to obtain a set of candidate features. Then, in the second stage, we use the forward feature selection [40] algorithm to select the best features from the candidate set for training the detection model.

Unlike 2-norm SVM, which minimizes $\|\mathbf{w}\|_2^2$ in its objective, 1-norm SVM minimizes $\|\mathbf{w}\|_1^2$ (called the LASSO penalty [42]). We utilize 1-norm SVM to derive the candidate set because it usually finds a sparse $\mathbf{w}$ (i.e., a $\mathbf{w}$ that tends to contain zeros) thanks to its "compressed sensing" interpretation [43]. To compile the candidate set, we only keep features that correspond to the nonzeros in $\mathbf{w}$, as the features that correspond to zeros are usually redundant or noisy [41]. Next, we use the forward feature-selection algorithm to select the final features from the candidate set. Initially, the set for storing the final features is empty. In each step, the algorithm selects one feature from the candidate set that yields the best improvement in SSVM's prediction accuracy[5] of SSVM and adds it to the feature set. The above step is repeated until the candidate set is empty, or there are no features in the candidate set can further improve the prediction accuracy.

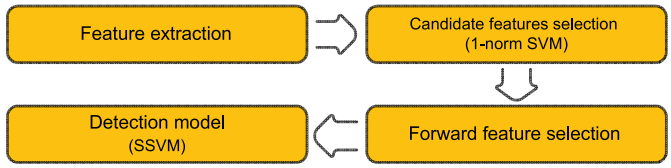


Fig. 4. Steps in training a detection model.

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed detection model.

A. Configuration

After the data cleaning step (described in Section IV-A), the set D contains 278 instances (i.e., sessions) of which 178 are positive (i.e., labeled $+1$, which denotes stalkers) and 100 are negative (i.e., labeled -1, which denotes account owners). Each instance is represented by an 139-dimension feature vector.

To the best of our knowledge, there are no other schemes for detecting in-situ identity fraud on SNS services. Thus, we evaluate the detection model by simulating different observation periods and compare the results. Specifically, given an observation period L minutes, we extract the behavioral features of a session from those actions performed within the first L minutes after the session starts. Then, the performance of the detection model is evaluated for $L = 1, 2, \ldots, 25$ min, respectively. Although the subjects were asked to browse an SNS account for 30 min in each round (see Section IV-A), we choose the maximum of L to be 25 rather than 30 because some sessions ended prematurely due to subjects' requests and resulted in a slightly shorter trace. Therefore, to ensure a comparable evaluation across all the sessions, we consider $L \leq 25$ here.

As described in Sections V-B and V-A, to construct the detection model, we first use an 1-norm SVM to derive the candidate features, and then use forward feature selection and SSVM with 10-fold cross validation to select the most distinguishing features as well as the hyper-parameters C and γ. Here, we use the leave-one-out cross validation [40] on D to evaluate the detection performance of our model.

B. Detection Performance at the 25th Minute

First, we consider the detection model's performance when $L = 25$ min. Table IV shows the results achieved by the model *with* and *without* feature selection. As we can see, feature

[5]We use 10-fold cross validation [40] to measure the accuracy.

TABLE IV
RESULTS ACHIEVED UNDER VARIOUS CONDITIONS

	With Oversampling	Without Oversampling
With feature selection	Acc: 93.53% FPR: 5.00% FNR: 7.30%	Acc: 90.29% FPR: 18.00% FNR: 5.06%
F-score	0.9483	0.9260
Without feature selection	Acc: 91.37% FPR: 6.00% FNR: 10.11%	Acc: 87.77% FPR: 22.00% FNR: 6.74%
F-score	0.9302	0.9071

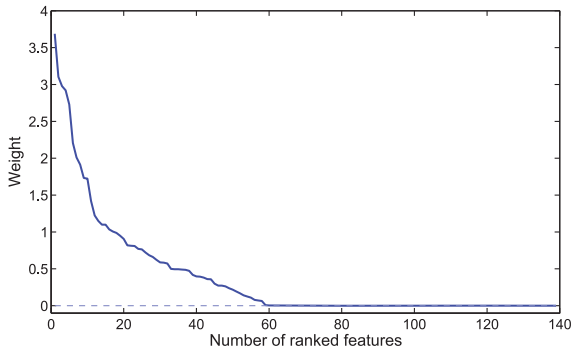


Fig. 5. Weights in $\mathbf{w}$ found by the 1-norm SVM over the corresponding features, which are ranked by their weights. Only 60 features (out of 139) remain in the feature set.

selection improves the performance because it yields higher accuracy/F-scores and lower FPRs/FNRs. The reason is that feature selection eliminates. As shown in Fig. 5, only 60 features (out of 139) remain in the feature set after using the 1-norm SVM for candidate selection.

We observe that the ratio of positive instances to negative instances in the dataset D is 1.78:1. The imbalance tends to yield a higher FPR. To resolve this issue, we use an *oversampling* approach to randomly select and then duplicate 78 negative instances to balance the ratio between positive and negative instances. The effect of duplicating an instance is to double the penalty if we misclassify the instance. Therefore, by duplicating the negative instances in D, we can avoid aliasing and reduce the FPR. Note that because the oversampling technique causes randomness, we train 10 models and average their results. Table IV shows the results achieved by our model *with* and *without* oversampling. We can see that the oversampling can control the tradeoff between FPR and FNR.

Fig. 6 shows the ROC curve and AUC of our model when feature selection and oversampling are applied. The AUC is fairly high (0.962), while the ROC curve shows that the model can achieve a TPR of 90% TPR and a FPR of 4.5%.

C. Early Detection Performance

To prevent theft of sensitive information, we should apply the *in situ* identity fraud detection scheme as early as possible in each session. In order to determine how our model performs with different time limits, we vary L from 1 to 25 min and train a model for each value of L with feature selection and oversampling. Fig. 7 shows the accuracy achieved by the models. When $L \geq 7$ min, the results are stable and reasonably good, and the accuracy rate is higher than 90%. Even at the 2nd minute,

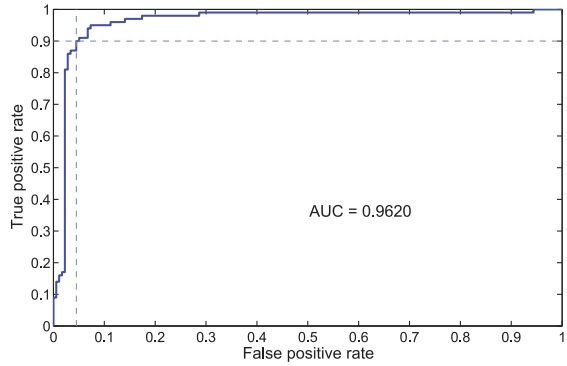


Fig. 6. ROC curve and AUC of the model at 25 min.

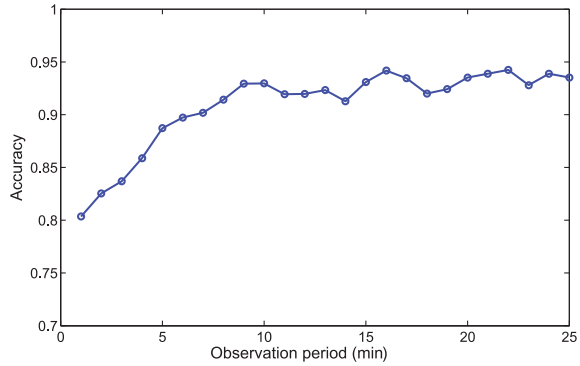


Fig. 7. Accuracy of the detection scheme with different observation periods. The graph shows that the detection model can achieve stable and reasonably good results after 7 min.

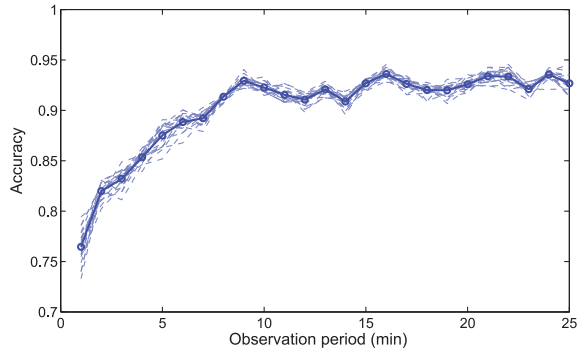


Fig. 8. Accuracy achieved by 20 models on 20 randomly permuted datasets. The thick line represents the average accuracy.

the accuracy is above 80%. This would help significantly as the scheme is used as a trigger for more sophisticated analysis.

To verify the robustness of the model, we randomly permute D for 20 times and use 10-fold cross validation [40] to train one model for each of the 20 permutations. Fig. 8 and Table V show the mean accuracy and standard deviation of the 20 models. The results indicate that the standard deviation of the accuracies is very low regardless of L. In addition, Figs. 8 and 7 show that the model performs consistently well using cross validations. Therefore, the performance of the detection scheme is very robust.

TABLE V
MEAN AND STANDARD DEVIATION OF THE ACCURACY GIVEN BY THE MODELS TRAINED USING THE 10-FOLD CROSS VALIDATION ON 20 RANDOMLY PERMUTED DATASETS

Minute	2	3	4	5	6	7
Mean	81.9%	83.2%	85.3%	87.5%	88.8%	89.2%
Std.	0.7%	0.8%	0.5%	1.0%	0.8%	0.5%
	8	9	10	11	12	13
Mean	91.3%	92.9%	92.2%	91.5%	91.0%	92.0%
Std.	0.3%	0.4%	0.4%	0.6%	0.5%	0.4%
	14	15	16	17	18	19
Mean	90.8%	92.7%	93.6%	92.6%	92.0%	92.0%
Std.	0.5%	0.3%	0.4%	0.5%	0.4%	0.6%
	20	21	22	23	24	25
Mean	92.5%	93.3%	93.3%	92.1%	93.5%	92.6%
Std.	0.4%	0.5%	0.7%	0.4%	0.4%	0.8%

VII. DISCUSSION

In this section, we discuss the representativeness of dataset collected from our controlled experiments and analyze the security robustness of the proposed scheme.

A. Dataset Validity

We acknowledge that while we did our best to mimic *in situ* identity fraud incidents in our controlled experiment (c.f., Section IV-A), a simulated environment is certainly not real-life scenarios and some behavioral differences would exist. Theoretically, a stalker in a real *in situ* identity fraud incident would 1) act secretly and not leave any evidence that can be traced by the account owners and 2) act under time pressure as account owners may come back to their devices depending on the situation. We consider that our dataset have captured, to some degree, the stalkers' realistic behavior for the following reasons.

1) The sessions used by acquaintances/strangers also yield much lower values in `f.act_add_comment` [Fig. 2(c)] and f.act_like [Fig. 2(d)] than those controlled by account owners. This shows that most stalkers' behavior is realistic as the stalkers would not like the owners to know of their usage, even we did not create a stealthy usage scenario for them.
2) Stalkers behave under time pressure, as shown in Fig. 9(a) that the accumulated average action rate decreases over time. The decrease in action rate implies that stalkers tend to explore the information they are most interested as early as possible, even we provided a full 30 min for their "stalking."
3) We inspect how the numbers of frequent action patterns (among all the collected stalker sessions) change with time. We first apply the cSpade algorithm [44] with a minimum support of 0.1 to identify the frequent action patterns shared by all the users in the stalker role. Then we measure the maximum support, i.e., the proportion of sessions sharing a frequent pattern, of the identified patterns over time. Fig. 9(b) shows that the accumulated maximum support is logarithmically increasing with time, which indicates that initially, stalkers perform similar sets of actions in the first few minutes. The commonality of actions gradually decreases in the latter stages of the session. This phenomenon manifests that stalkers tend to check more "important" information they are interested and after that their surfing behavior diversifies due to difference in individual preferences.

In sum, as collecting stalkers' behavior in real-life *in situ* identify fraud is extremely challenging, if not impossible, we believe that our controlled experiments provide a reasonable approximation of the real-life scenario and, to some extent, capture the essential characteristics of stalkers' behavior, as indicated by the significant role-driven behavioral diversity (Section IV-C).

Although our dataset is an approximate of the reality, we consider the proposed scheme can well serve a *bootstrapping* role for *in situ* fraud detection. Once the proposed scheme is deployed by SNS operators and real-life *in situ* identify fraud incidents have been identified, the model could be further revised by learning from the newly captured realistic stalking behavior and further enhance the utility of the fraud detection scheme.

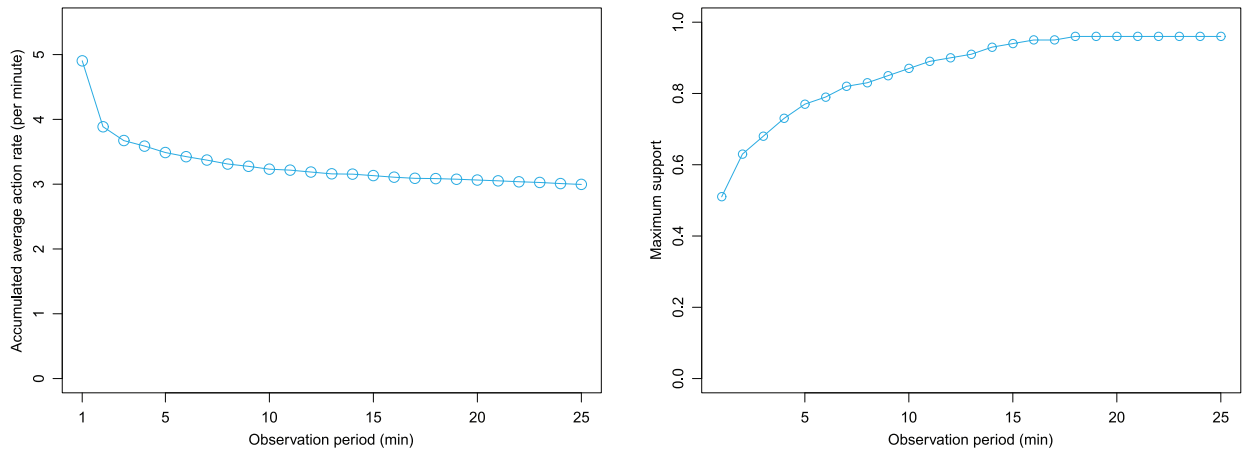
B. Security Analysis

As shown in Fig. 3, the data collection, processing, decision-making, and follow-up actions (such as challenging the users if they are genuine account holders) in the proposed scheme are all performed on the server side. Hence, it is impossible for attackers to compromise the scheme by interfering the fraud detection process from the clients sides.

As the detection scheme in running on the server side (i.e., by the SNS operators), an attacker cannot avoid detection once he logs in because all users are monitored by the server. The only way for an attacker to continue using the compromised account is to evade the detection model.

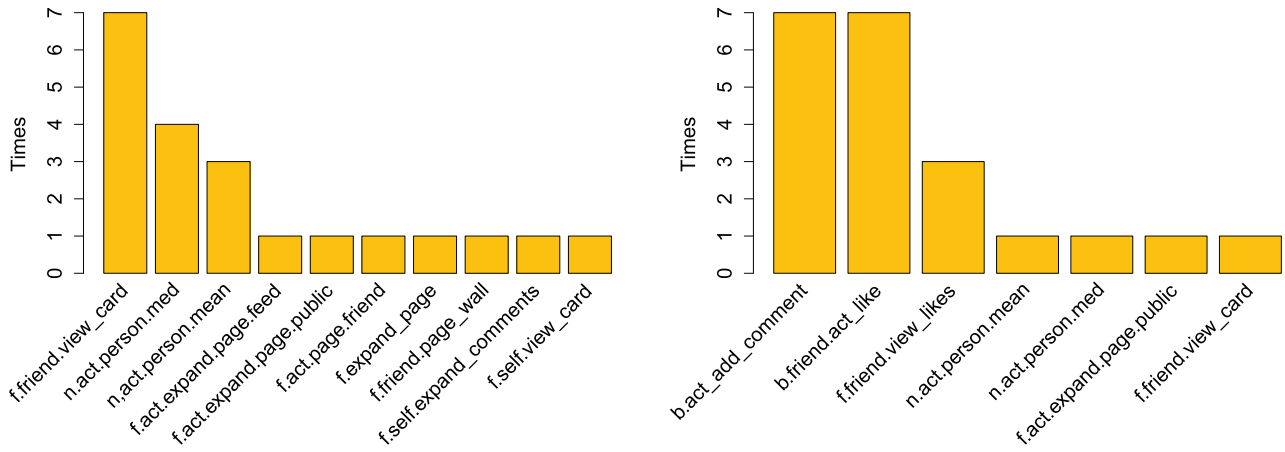
The detection model does not rely on any cryptographic techniques and it is based completely on user behavior. Hence, to avoid detection by the model, an attacker must 1) mimic the account owner's actions or 2) perform as few actions as possible and exit the site. Attacks of the first type are less likely because account owner's behavior patterns are not well documented [37]. Even if some attackers could successfully mimic owners' actions, they would be forced to spend time on something that is of little interest to them and may miss some desirable information. This makes the attack less harmful. In the second type of attacks, attackers are under time pressure because the detection model can achieve close to 80% accuracy even if attackers only browse victims' newsfeeds for 1 min. The time pressure makes the attacks less harmful because attackers may not be able to find the information they want in such a limited period of time.

In addition, the detection scheme is not tied to a specific detection model. For example, a personalized detection model may be particularly helpful in identifying the first type of attacks because it is even harder to imitate an individual's behavior. Moreover, a detection model that considers the timestamp of each action may help identify attacks of the second type, as users (both account owners and stalkers) often perform actions in the some order based on habit. While this paper points out the effectiveness of continuous authentication on detecting in-situ identity fraud, it would certainly be



(a) (b)

Fig. 9. Evidence that stalkers in our experiments tend to finish their favored activities in the early stages of the sessions. (a) Action rate of stalkers. (b) Maximum support of frequent action patterns of stalkers.



(a) (b)

Fig. 10. Frequency of the topmost positive and negative significant features that distinguish stalkers from account owners in the first 7 min. (a) Significant features for stalkers. (b) Significant features for account owners.

possible to develop more sophisticated detection models to defeat increasingly smart attackers.

One might also suspect that the model training process could be contaminated by malicious attackers. We consider the risk of this attack to be negligible due to the following reasons.

1) The *in situ* identify fraud issue we discuss in this paper is "not scalable" since it normally involves physical access of computer devices; in other words, unlike other types of Internet frauds, such as phishing [45], a malicious user cannot easily extend in-situ identify fraud attacks on dozens or even hundreds of victims. Thus, it is of little incentives for a malicious user to manage to compromise the model training process of an *in situ* identify fraud detection scheme.
2) The collection of the dataset does not necessarily involve uncontrolled crowds where malicious attackers could infiltrate. One option is to only collect training data from in-house, controlled crowds, like what we did in Section IV. One another option is to reassure the identity of the users via a separate authentication scheme, such as challenging the users with secret personal questions that are often used by web services when users retrieve lost passwords. In this way we can collect the training dataset only from trustworthy users whose identities are double confirmed.

VIII. Conclusion

In this paper, we have proposed a low-cost detection scheme for SNSs that analyzes users' browsing behavior to detect *in situ* identity fraud incidents. Using Facebook as a case study, we show that 1) the *role-driven behavioral diversity* property does exist; 2) the property can be exploited to design a low-cost detection scheme that is applicable to all users; and 3) the scheme is hard to evade and it renders a reasonable detection performance after an observation period of 2 min.

In our future work, we will study the browsing behavior of individuals and develop personalized detection models. Such models can only be activated when needed and provide detailed analysis for suspicious sessions. We will also improve

the current low-cost detection model to achieve higher detection accuracy within the first few minutes. Such improvements would be possible because we see different user behavior patterns in a variety of time scales. As an example, by counting the occurrences of significant features, i.e., those with the three most positive and three most negative weights, in SSVM with the observation period L varies from 1 to 7 min, as shown in Fig. 10(a) and (b) shows, respectively, we find that some of the significant features in the first 7 min are not prominent in the full 25-min model derived in Section IV-C. This exhibits that users' behavior may change over time and it hints the possibility of more accurate early detection by utilizing such time-dependent browsing actions. We hope that this work will motivate in-depth studies on developing more sophisticated models to prevent *in situ* identity fraud in general.

REFERENCES

[1] J. Constine. (2010). *Facebook has Users Identify Friends in Photos to Verify Accounts, Prevent Unauthorized Access* [Online]. Available: http://www.insidefacebook.com/2010/07/26/facebook-photos-verify/, accessed on 2012.

[2] J. Constine. (2012). *Facebook Asks Every User for a Verified Phone Number to Prevent Security Disaster* [Online]. Available: http://techcrunch.com/2012/06/14/facebook-security-tips/, accessed on 2012.

[3] C. Technologies. (2012). *Phone Data Makes 4.2 Million* Brits Vulnerable to Id Theft* [Online]. Available: http://www.credant.com/news-a-events/press-releases/69-phone-data-makes-42-million-brits-vulnerable-to-id-theft.html, accessed on 2012.

[4] P. Mah. (2011). *Stored Passwords Add to Mobile Security Risks* [Online]. Available: http://www.itbusinessedge.com/cm/blogs/mah/stored-passwords-add-to-mobile-security-risks/?cs=47183, accessed on 2012.

[5] R. Yu. (2012). *Lost Cellphones Added Up Fast in 2011* [Online]. Available: http://usatoday30.usatoday.com/tech/news/story/2012-03-22/lost-phones/53707448/1, accessed on 2012.

[6] E. Hansberry. (2011). *Most Consumers Don't Lock Mobile Phone Via Pin* [Online]. Available: http://www.informationweek.com/mobility/security/most-consumers-dont-lock-mobile-phone-vi/231700155, accessed on 2012.

[7] Facebook. (2012). *Removal of Offline_Access Permission* [Online]. Available: https://developers.facebook.com/roadmap/offline-access-removal/, accessed on 2012.

[8] Y.-J. Lee and O. Mangasarian, "SSVM: A smooth support vector machine for classification," *Comput. Optim. Appl.*, vol. 20, no. 1, pp. 5–22, 2001.

[9] S.-H. Wu, K.-P. Lin, C.-M. Chen, and M.-S. Chen, "Asymmetric support vector machines: Low false-positive learning under the user tolerance," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2008, pp. 749–757.

[10] J. He, W. Chu, and Z. Liu, "Inferring privacy information from social networks," *Intell. Secur. Informat.*, vol. 3975, pp. 154–165, 2006.

[11] E. Zheleva and L. Getoor, "To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles," in *Proc. 18th Int. Conf. World Wide Web (WWW)*, 2009, pp. 531–540.

[12] C. Tang, K. Ross, N. Saxena, and R. Chen, "What's in a name: A study of names, gender inference, and gender behavior in facebook," *Database Syst. Adv. Appl.*, pp. 344–356, 2011.

[13] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All your contacts are belong to us: Automated identity theft attacks on social networks," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 551–560.

[14] L. Jin, H. Takabi, and J. B. Joshi, "Towards active detection of identity clone attacks on online social networks," in *Proc. 1st Conf. Data Appl. Security Privacy (CODASPY'11)*, 2011, pp. 27–38.

[15] A. Felt and D. Evans, "Privacy protection for social networking APIs," Web 2.0 Security and Privacy (W2SP), 2008.

[16] R. Wishart, D. Corapi, A. Madhavapeddy, and M. Sloman, "Privacy butler: A personal privacy rights manager for online presence," in *Proc. 8th IEEE Int. Conf. Pervasive Comput. Commun. (PERCOM Workshops)*, 2010, pp. 672–677.

[17] S. Mahmood and Y. Desmedt, "Your facebook deactivated friend or a cloaked spy," in *Proc. IEEE Int. Workshop Secur. Soc. Netw. (SESOC'12)*, 2012, pp. 367–373.

[18] Facebook. (2015). *Facebook Security* [Online]. Available: http://www.facebook.com/security, accessed on 2012.

[19] Facebook. (2009). *Facebook's Privacy Policy—2. Information We Receive* [Online]. Available: http://www.facebook.com/note.php, accessed on 2012.

[20] T. F. Lunt *et al.*, "IDES: The enhanced prototype-a real-time intrusion-detection expert system," Tech. Rep. SRI Project 4 185-010, SRI-CSL-88, 1988.

[21] G. K. Kuchimanchi, V. V. Phoha, K. S. Balagani, and S. R. Gaddam, "Dimension reduction using feature extraction methods for real-time misuse detection systems," in *Proc. IEEE Inf. Assur. Workshop*, 2004, pp. 195–202.

[22] D.-K. Kang, D. Fuller, and V. Honavar, "Learning classifiers for misuse detection using a bag of system calls representation," in *Intelligence and Security Informatics*. New York, NY, USA: Springer, 2005, pp. 511–516.

[23] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *J. Netw. Comput. Appl.*, vol. 28, no. 2, pp. 167–182, 2005.

[24] R. Cathey, L. Ma, N. Goharian, and D. Grossman, "Misuse detection for information retrieval systems," in *Proc. 12th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2003, pp. 183–190.

[25] C. Y. Chung, M. Gertz, and K. Levitt, "DEMIDS: A misuse detection system for database systems," in *Integrity and Internal Control in Information Systems*. New York, NY, USA: Springer, 2000, pp. 159–178.

[26] M. Meier, "A model for the semantics of attack signatures in misuse detection systems," in *Information Security*. New York, NY, USA: Springer, 2004, pp. 158–169.

[27] P. Helman and G. Liepins, "Statistical foundations of audit trail analysis for the detection of computer misuse," *IEEE Trans. Software Eng.*, vol. 19, no. 9, pp. 886–901, Sep. 1993.

[28] S. J. Stolfo, S. Hershkop, K. Wang, O. Nimeskern, and C.-W. Hu, "Behavior profiling of email," in *Intelligence and Security Informatics*. New York, NY, USA: Springer, 2003, pp. 74–90.

[29] S. Shepherd, "Continuous authentication by analysis of keyboard typing characteristics," in *Proc. Eur. Conv. Secur. Detect.*, 1995, pp. 111–114.

[30] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proc. ACM Workshop Visual. Data Min. Comput. Secur.*, 2004, pp. 1–8.

[31] T. Feng *et al.*, "Continuous mobile authentication using touchscreen gestures," in *Proc. IEEE Conf. Technol. Homeland Secur.*, 2012, pp. 451–456.

[32] K. Niinuma and A. K. Jain, "Continuous user authentication using temporal information," in *Proc. Biom. Technol. Hum. Ident. VII*, 2010, p. 76670L.

[33] K. Niinuma, U. Park, and A. K. Jain, "Soft biometric traits for continuous user authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 4, pp. 771–780, Dec. 2010.

[34] R. H. Yap, T. Sim, G. X. Kwang, and R. Ramnath, "Physical access protection using continuous authentication," in *Proc. IEEE Conf. Technol. Homeland Secur.*, 2008, pp. 510–512.

[35] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "COMPA: Detecting compromised accounts on social networks," in *Proc. Symp. Netw. Distrib. Syst. Secur. (NDSS)*, 2013.

[36] K. Hampton. (2011). *Social Networking Sites and Our Lives Part 2: Who are Social Networking Site Users?* [Online]. Available: http://pewinternet.org/Reports/2011/Technology-and-social-networks/Part-2/Facebook-activities.aspx, accessed on 2012.

[37] A. N. Joinson, "Looking at, looking up or keeping up with people? Motives and use of Facebook," in *Proc. SIGHI Conf. Hum. Factors Comput. Syst. (CHI'08)*, 2008, pp. 1027–1036.

[38] P. Domingos, "A few useful things to know about machine learning," *Communi. ACM*, vol. 55, no. 10, pp. 78–87.

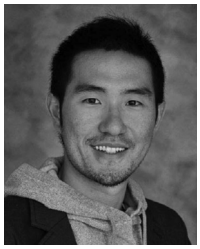
[39] C.-M. Huang, Y.-J. Lee, D. Lin, and S.-Y. Huang, "Model selection for support vector machines via uniform design," *Comput. Stat. & Data Anal.*, vol. 52, no. 1, pp. 335–346, 2007.

[40] I. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2005.

[41] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, vol. 16, pp. 49–56.

[42] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy.*

Stat. Soc. B (Methodological), vol. 58, pp. 267–288, 1996.
[43] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.
[44] M. J. Zaki, "Sequence mining in categorical domains: Incorporating constraints," in *Proc. 9th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2000, pp. 422–429.
[45] K.-T. Chen, J.-Y. Chen, C.-R. Huang, and C.-S. Chen, "Fighting phishing with discriminative keypoint features," *IEEE Internet Comput.*, vol. 13, no. 3, pp. 56–63, May/Jun. 2009.

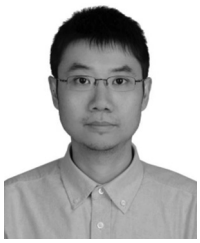


Shan-Hung Wu received the Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan.

He is currently an Associate Professor with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Before joining the National Tsing Hua University, he was a Senior Research Scientist with Telcordia Technologies Inc, Piscataway, NJ, USA. He has authored many research papers in top-tier conferences, such as ICML, KDD, INFOCOM, Mobihoc, ICDE, and ICDCS. His research interests include machine learning, data mining, database systems, and mobile applications.

Man-Ju Chou received the B.S. and M.S. degrees in computer science from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2011 and 2013, respectively.

Since then, she has been a Data Engineer with Yahoo APAC Data Team, Taipei, Taiwan, working on business intelligence and CRM system.

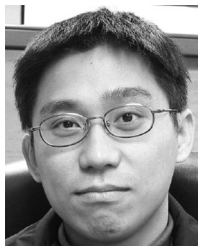


Chun-Hsiung Tseng received the B.S. degree in computer science from the National ChengChi University, Taipei, Taiwan, and the M.S. and Ph.D. degrees in computer science from the National Taiwan University, Taipei, Taiwan.

He was a Research Assistant with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, from 2003 to 2010. He was a Faculty Member with the Department of Computer Information and Network Engineering, Lunghwa University of Science and Technology, Taoyuan, Taiwan, from 2010 to 2013. He is currently a Faculty Member with the Department of Information Management, Nanhua University, Chiayi, Taiwan. His research interests include big data analysis, crowd intelligence, e-learning systems, and Web information extraction.

Yuh-Jye Lee received the Ph.D. degree in computer science from the University of Wisconsin–Madison, Madison, WI, USA, in 2001.

He is currently a Professor of Computer Science and Information Engineering with the National Taiwan University of Science and Technology, Taipei, Taiwan. He also serves as a Principal Investigator with the Intel–NTU Connected Context Computing Center, Taipei, Taiwan. His research interests include optimization theory, network and information security, machine learning, big data, data mining, numerical optimization, and operations research. His recent major research is applying machine learning to information security problems such as network intrusion detection, anomaly detection, malicious URLs detection and legitimate user identification. Currently, he focus on online learning algorithms for dealing with large-scale datasets, stream data mining, and behavior-based anomaly detection for the needs of big data, Internet of Things data analytics, and machine-to-machine communication security problems.



Kuan-Ta Chen (a.k.a. Sheng-Wei Chen) (S'04–M'06–SM'15) received the Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 2006, and the B.S. and M.S. degrees in computer science from the National Tsing-Hua University, Hsinchu, Taiwan, in 1998 and 2000, respectively.

He is a Research Fellow with the Institute of Information Science and the Research Center for Information Technology Innovation (joint appointment), Academia Sinica, Taipei, Taiwan. His research interests include quality of experience, multimedia systems, and social computing. Dr. Chen has been an Associate Editor of *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* since 2015. He is a Senior Member of ACM.