國立臺灣大學電機資訊學院電機工程學系

碩士論文

Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

以統計方法進行Facebook盜用行爲偵測

Facebook Account Misuse Detection — A Statistical Approach

蔡佩眞

Christine Peijinn Chai

指導教授：雷欽隆 博士

Advisor: Chin-Laung Lei, Ph.D.

中華民國102年6月

June 2013

誌謝

這篇碩士論文的完成，要感謝很多幫助過或鼓勵過我的人。首先要感謝指導教授雷欽隆博士兩年來的指導，每次做研究定期報告的提供建議，讓我在研究過程中可以往正確的方向前進。也要感謝中研院資訊所的陳昇瑋博士，以及整個研究團隊，先行做實驗與收集資料樣本，讓我升碩二暑假去中研院實習，就能直接進行資料分析，以及開學後回到台大的電子郵件時相討論。同時也感謝台科大資訊工程系的李育杰博士，指導我機器學習的基礎知識。感謝台大寫作教學中心的邱崇賢老師，很高興可以選到他的學術英文寫作課，在論文撰寫上獲益良多。此外也謝謝整個網路計算暨安全實驗室的成員們，在各方面的協助，從論文格式到口試準備，以及男友的時相陪伴。最後還要感謝我的父母，這兩年的心靈支持與鼓勵。

摘要

社群網站上的個人資料是重要的課題，因爲一旦社群網站的個人帳號被盜用，所有在上面的個人資料都會被第三者取得，不論帳號擁有者做過任何隱私權設定。因此，本篇論文以統計方法並使用Support Vector Machine (SVM)，進行臉書的盜用行爲偵測。經由分析使用者在線上的瀏覽紀錄，可以發現正常的使用者在社群網站的行爲比較主動，盜用帳號者偏好閱讀私人訊息。

關鍵字：臉書、盜用帳號、統計方法、Support Vector Machine (SVM)、分類、交叉驗證

Abstract

Privacy of personal information on social networking websites has become an important issue, because when a social networking website account is used by a person other than the owner, all personal data stored on the website can be retrieved, no matter how the owner sets the privacy options. Therefore, this paper proposes a statistical approach with the use of Support Vector Machine (SVM) to detect whether the Facebook account user is the actual owner. By analyzing online browsing behavior features, it is found that the normal user tends to be more active and that the stealthy user prefers to read personal messages.

Keywords: Facebook, account misuse, statistical approach, Support Vector Machine (SVM), classification, cross validation.

Contents

List of Figures

List of Tables

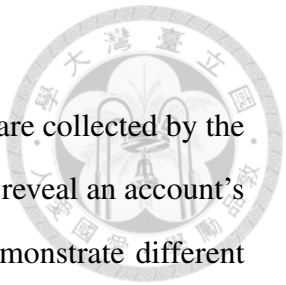
Chapter 1

Introduction

There is an increasing popularity on social networking websites, and people are spending more time interacting with each other via online social networks. In order to reach and to be reached by their friends, people tend to disclose some of their personal information publicly, but ”Despite concerns raised about the disclosure of personal information on social network sites, research has demonstrated that users continue to disclose personal information.” [36] Since people nowadays rely more on online social networking, the privacy issue has become more important.

1.1 Personal Information Online

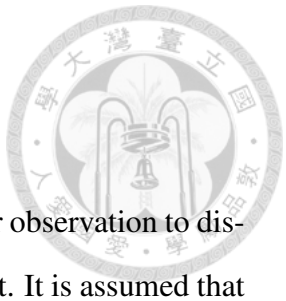
Personal information is the data related to a specific person. In the past, personal information generally included name, gender, contact number, date of birth, employment, and other information that could be found in one’s name card or identification card. Since the Internet became more widely used, more types of personal information have appeared in real life, such as online browsing behaviors and social networks profiles. Most kinds of personal information should not be disclosed without the owner’s approval, but accidental leakage of personal information can still happen due to private information inference from public data [24, 13]. What is worse, if an account on a social network site has been



used by someone other than the owner, all data stored on the account are collected by the stealthy user. Fortunately, the browsing habits on social networks can reveal an account's identity – whether the user is the true owner or not. Since users demonstrate different behavior patterns when logging in their own and others' accounts, the stealthy usage can be detected.

1.2 The Problem of Online Social Networks

Because the online mechanism of the social network server approves a user's login as long as the account name and password are correct, people other than the account owner can use or even steal the account. However, there are still some methods to tell the differences between stealthy users and actual account owners. Take Facebook for example. The device, web browser, IP address, and the Operating System used by the user are recorded on the server [10]. When the user logs in with a different IP or place, the server may ask him/her to identify some pictures of his/her friends [6]. If some friends are tagged on cartoon characters or even objects, it will be more difficult for the real account owner to pass the test. On the other hand, if one logs in his/her friend's account, it is very probable that he/she has many mutual friends with the account owner, and acquaintances have an advantage to pass the verification. Besides, modern Web browsers can remember users' passwords and keep logged-in sessions, and when users access the same online social network again, the login step is automatically bypassed. This is a big security threat especially for mobile devices because it is cumbersome to input passwords [16], and most people do not use PINs to protect their smart phones [25]. As a result, existing solutions cannot tell whether the user using the same web browser and device is actually "the same" or not.



1.3 Our Contribution

In the present paper, we propose an approach using statistical behavior observation to distinguish between the account owner and others using the same account. It is assumed that a social network website user has consistent usage behavior when he/she uses the same website. Various types of user behaviors as variables can tell whether the current account is used stealthily if the hypothesis holds. In our experiment, two pairs of acquaintances are recruited to login their Facebook accounts for three consecutive rounds, 30 minutes each. Afterwards, the subjects are instructed to use each other's Facebook account, and every subject has the chance to use his/her own, the partner's, and a stranger's account, while all statistical behaviors are recorded by a plugged-in program set beside the browser. Our observation reveals that if one is using his/her own account, he/she is more active in commenting and clicking on "like," and he/she views less personal messages or private clubs.

To the best of our knowledge, currently there is no other detection scheme for the usage stealing problem in Online Social Networks (OSNs). Thus we compared various statistical approaches to the model, and proved account misuse detection on social networking websites to be feasible. The approaches include 2-class SVM (Support Vector Machine), 2-class SVM with p-value variable selection, weight adjustment, 3-class SVM, and oversampling by using a sample dataset with ratio 1:2 between accounts used by self and others. Given leave-one-out cross validation, 3-class SVM and oversampling achieve more than 95% accuracy. For separate training and testing sets, 3-class SVM performs the best with 75% accuracy. Therefore, if a person uses someone else's Facebook account, it is very likely that the stealthy use will be caught. After a stealthy use of account is detected, the online social network server can issue a warning to the account owner via email or mobile phone to enhance his/her privacy awareness of personal information online.

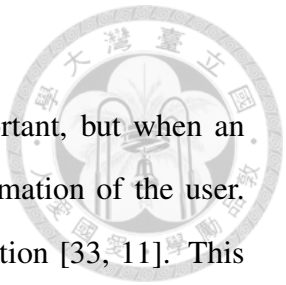
Chapter 2

Related Work

Since there is a growing demand of information privacy, prior work has been done in the information security of OSNs. The research papers listed in Section 2.1 discuss the attack and defense from users, but the server should also detect account misuse to be responsible for the privacy leakage of users. There are two kinds of authentication: static and continuous. Static authentication is about requiring the user to enter an account and the corresponding password when he/she logs in; as a result, if someone gains access to the password, the server will wrongly believe he/she is the actual user. On the other hand, continuous authentication monitors the whole user login session, so the server protects the user's account from login to logout [14]. However, continuous authentication should not interrupt the user unless the server begins to doubt the person's identity, or the measure will annoy most normal users [9]. Continuous authentication will be discussed in Section 2.2, and this leads to the importance of statistical behavior analysis of OSNs.

2.1 Security and Privacy for Online Social Networks

Some works center their attention on Facebook attack methods such as frequent account deactivation [17]. This kind of research fails to consider the possibility that the account is being attacked by itself instead of other accounts. Others, concentrate on the inference



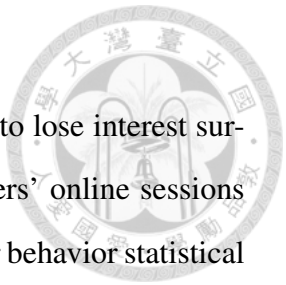
from public to private information [24, 13, 38]. This field is important, but when an attacker logs into a certain account, he/she can get all private information of the user. Still others focus on the monitoring of distributing personal information [33, 11]. This is mainly about third parties who have a desire to get personal data through software applications, but the personal privacy rights manager still cannot prevent the misused account from access of its own information.

2.2 Continuous Authentication

Continuous authentication has been discussed by many researchers for more than 10 years. In 1995, Shepherd proposed a method by analyzing keyboard typing characteristics, but this requires the user to type the word ”PASSWORD” five times, which is a severe interruption to the user [21]. Mobile devices like portable computers and smart phones are widely used in modern times, and Yazji came up with an implicit user re-authentication method based on electricity consumption due to filesystem activity and network access [35]. However, the action of an account in an OSN generally includes expanding page and commenting, no matter normal use or stealthy use, so Yazji’s scheme is not suitable for OSNs. Continuous authentication with biometrics is currently a popular issue, but each biometric has its own strengths and weaknesses, and using more features for detection is more expensive on the user side [34, 19, 20, 23].

2.3 The Importance of Statistical Behavior Analysis

However, currently existing methods for stealthy account usage detection are not directly related to the statistical behavior of true account owners and accounts used by other people. It is shown that statistical behavior of users assumed to be account owners can be observed. ”Interaction activity on Facebook is significantly skewed towards a small portion of each user’s social links.” and ”Users’ online time spending can be modeled with



Weibull distributions; soon after subscribing, a fraction of users tend to lose interest surprisingly fast; and the duration of OSN (Online Social Network) users' online sessions shows power law distribution characteristics." are the examples of user behavior statistical models [32, 12]. Therefore, a statistical approach can be an efficient way to distinguish between accounts used normally or stealthily because self-using accounts generally have similar behavioral patterns.

Chapter 3

Problem Description

The target of this research is to derive a statistical model for Facebook account misuse detection, in order to protect personal information. There are two major causes for accounts being used by people other than the owner. The first one is that the password has been stolen by a third party through online transmission interception. The second reason is that people generally allow the browser to remember their passwords for convenience, and that they leave the OSN logged on when they are temporarily away. In these cases, most access control mechanisms that require the correct password will not be functioning.

3.1 Assumptions

In this paper, a behavior analysis method is proposed, and a few assumptions are made:

1. Human beings generate consistent behavior when using the same OSN.

2. Accounts that have been used normally and those which have been used stealthily have significant differences in behavior.

3. The selected features, such as giving a ”like” and commenting on a post, can effectively reflect the user’s behavior.

4. The selected features can be observed and recorded by intercepting HTTP/HTTPS traffics.

5. The selected features are ubiquitous in most OSNs; that is, they are not limited to the characteristics of Facebook.

3.2 Limitations

The privacy issues about social networks can be discussed in various aspects, so we would like to exclusively focus on account misuse detection. Therefore, the listed topics below are out of the scope for this research project:

1. Behavioral differences related to age and gender in OSNs.

2. Privacy settings such as which should (not) be set visible to the public.

3. Relationship network analysis and how misused friends' accounts affect the user.

4. Encryption, decryption, and interception through the transmission channel.

3.3 Goals

The results are expected to be feasible and should satisfy the following conditions:

1. It can be applied to most OSNs; i.e. the results are general enough.

2. The OSN server catches accounts not used by their owners as many as possible, and sends false alarms to account owners as less frequently as possible. The latter is more important because false alarms annoy the users a lot.

3. After an adequate period of training the model, the accuracy of testing unseen data should be acceptable, to ensure the success of account misuse detection.

Chapter 4

Data Collection and Processing

It is extremely difficult to acquire stealthily-used Facebook accounts in real life, because asking users to login others' account in real life is almost impossible. Therefore, "Facebook User Behavior" experiments to actively collect stealthily-used account data were conducted, and account users were classified into three groups: self, acquaintance, and stranger; this is shown in Fig. 4.1.

We posted the recruitment on the largest online bulletin board system in Taiwan (PTT.cc), and 15 experiments were successfully conducted from May to August 2012. A total of 60 subjects (38 males and 22 females, average 23.5 years old), and we completed 172 successful rounds in total. Females are generally more concerned about privacy [24], so they may be less willing to allow others to use their Facebook accounts.

4.1 Experiment Conduction

In each experiment, two pairs of acquaintances are recruited, and subjects between pairs should be mutual strangers. "Acquaintance" means that two people know each other in real life, and that they may be colleagues, friends, classmates, and family members. If we arrange different accounts to different subjects, each subject may browse his/her own account, an acquaintance's, or a stranger's. Moreover, to avoid the effect of light users,

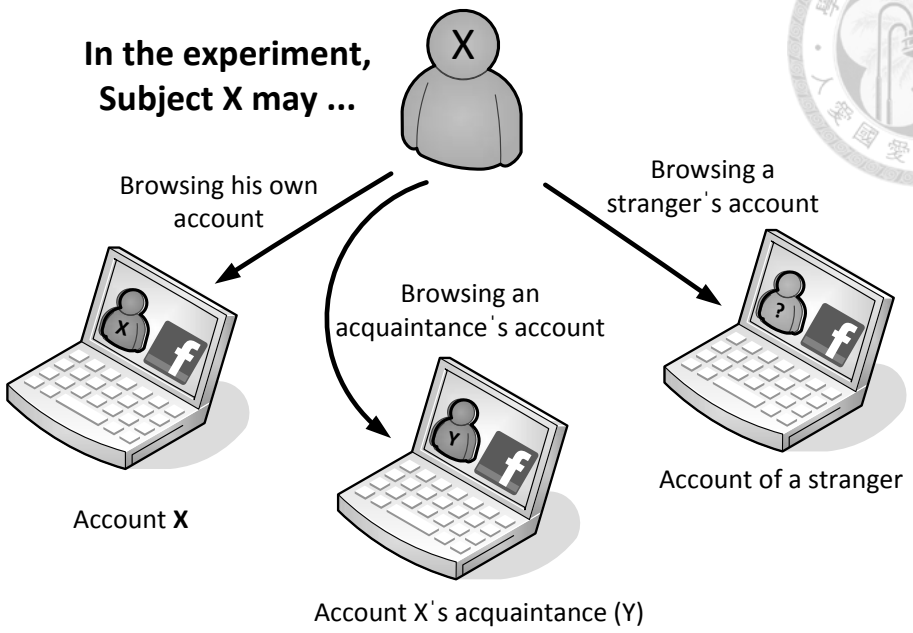


Figure 4.1: An account can be used by its owner, an acquaintance, or a stranger

all subjects attending the experiment should have more than 50 friends and spend at least four hours per week on Facebook.

There are four seats for the experiment, and each seat is equipped with one Windows 7 computer with a Google Chrome browser. The four subjects were assigned to a corresponding seat and instructed to login their Facebook accounts. They are required to browse their friend list page first because we need to crawl information about every subject's friends.

The experiment consists of three rounds, and each rounds lasts for 30 minutes. After a round ends, we shuffle the seat for every subject, and keep all accounts logged in. The experimental setup is shown in Fig. 4.2. All subjects are instructed to use the account directly on the computer, and every subject has the chance to use his/her own, an acquaintance's, and a stranger's account.

The subjects are allowed to do anything on Facebook except for playing online games and sabotage activities like changing the account password. In order to keep the subjects focused on the experiment, they are allowed to follow external links but for less than 1 minute, and they will be reminded by a browser warning message when browsing the

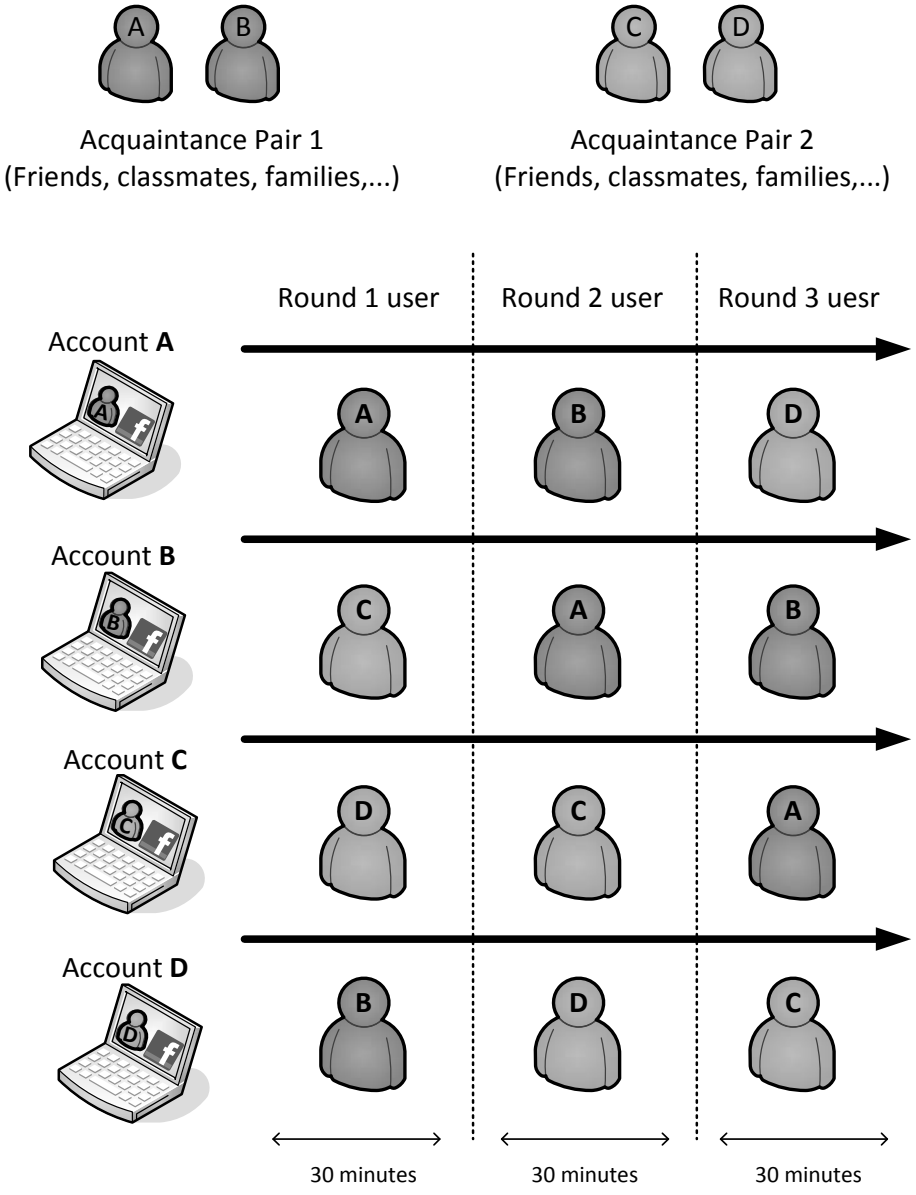
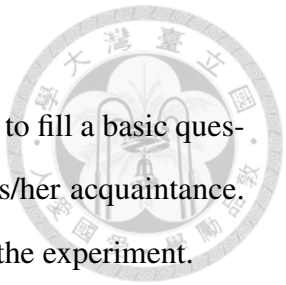


Figure 4.2: Experimental setup for three rounds



external webpage. After the experiment ends, each subject is required to fill a basic questionnaire about his/her gender, age, address, and relationship with his/her acquaintance. A gift certificate of 300 NTD is given to each subject who completes the experiment.

4.2 Data Recording

To record the statistical behaviors of the user, we used Fiddler [26], a free Web debugging proxy, to monitor all the HTTP/HTTPS GET/POST requests by a browser. By parsing HTTP/HTTPS request logs, we were able to capture every action performed by a user on Facebook. We also developed a Google Chrome extension to record all new tab actions, in order to find parallel browsing during the experiment. We even customized the Google Chrome extension to detect the url of the current tab; if a subject watches a non-Facebook website, a warning message will show on the screen.

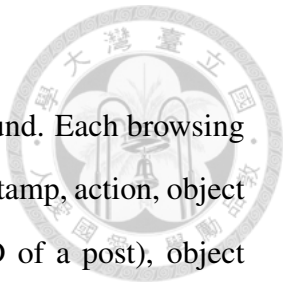
After recording every request of experiment computers, we parsed logs and derived different types of browsing. From the experiment logs, we found more than 20 request patterns corresponding to specific browsing actions, and they are classified into four groups:

1. **Page group**: Any action directing to a new page or refreshing a page, such as a message page and the news feed page of the account.

2. **Action group**: Actions about user-generated content, such as giving or canceling a ”like” to a post and making comments on a photo, and following the link on a post.

3. **Viewing group**: Actions like viewing a hover card of an entity and watching a photo.

4. **Expand group**: Actions which do not result in changing a page, including expanding newsfeed page or comments of a post.



We also derived a browsing action log of each subject in every round. Each browsing action log contains six columns of every action record, including timestamp, action, object owner (e.g. the author of a post), object ID (e.g. the Facebook ID of a post), object description, and the tab number in the browser.

4.3 Data Processing

To maintain validity, all data should be cleaned up before doing analysis. Invalid traces should be removed, and there are 172 successful rounds in total. The datapoints showing the user was idle more than five minutes are rejected, since these imply the subject may be doing something other than using Facebook, and there are 163 rounds left.

The valid traces were processed into feature variables, classifying into four types:

1. **Ratio/Rate**: It is defined as the total actions divided by the total time span (minutes). For example, the feature ratio.expand_page indicates the average time a user spends on expanding pages. In the beginning, the feature type was set to ”ratio,” and the type name was changed to ”rate” afterwards.

2. **Binary**: If a feature value is nonzero, its binary value is 1; otherwise, the binary value is 0. For example, the feature b.expand_page represents whether a user expands a page.

3. **Number**: Number is the total number of observed actions within the time span. For example, the feature n.act.page.feed captures how many browsing actions a user does in the allotted time.

4. **Time span**: It is defined as the time spent for a single action, and it can be expanded to the sum, mean, standard deviation, maximum, and minimum values in a group of actions. Note that this category does not apply to all behaviors, and we only record

the time span for actions that take more than a second to perform, such as reading a certain page.

Chapter 5

Models for Data Analysis

Support Vector Machine (SVM) is one of the most suitable classification algorithms for this experiment for two reasons. First, it works well with small-scale data [5, 8], and we only have 163 samples. Second, in another cybercrime detection model based on Facebook data, Deylami showed that SVM is better than AdaBoostM1 [2] and Naive Bayes [7]. Therefore, we selected SVM as our main account misuse detection scheme.

SVM was adopted for classification of account users, and it was implemented on R version 2.12.1 with an x86 64-bit Unix computer. The primary methods include discovering new features, using the basic 2-class SVM, and using SVM with p-value variable selection. The secondary methods involve more statistical knowledge, consisting of weight adjustment, oversampling, and 3-class SVM. Finally, cross validation was done to verify the effectiveness of the results. The results comprise a confusion matrix and three values: accuracy, false positive rate (FPR), and false negative rate (FNR).

5.1 Support Vector Machine

”Support Vector Machines are among the most robust and success classification algorithms. They are based upon the idea of maximizing the margin i.e. maximizing the minimum distance from the separating hyperplane to the nearest example.” [4] The clas-

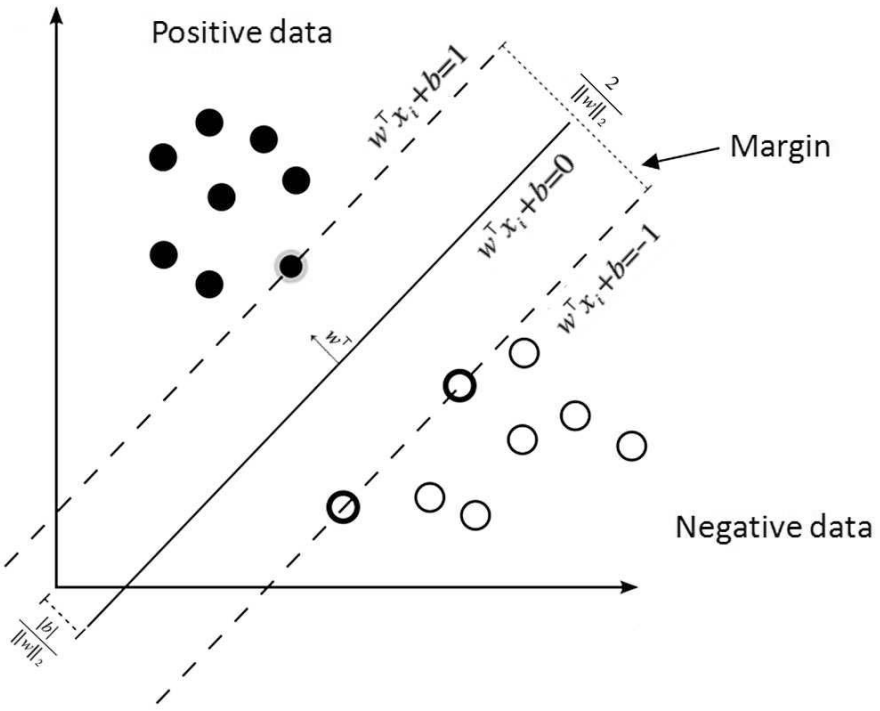


Figure 5.1: The separating hyperplane and margin of SVM

sification of positive and negative data is shown in Fig. 5.1.

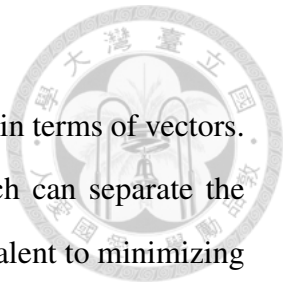
Given a training dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ with size n, where $x_i \in R^d$ is the training instance and $y_i \in \{1, -1\}$ is the corresponding label. For an account not used by its owner, $y_i = 1$; otherwise, $y_i = -1$. Since our goal is to distinguish accounts used by non-owners from account normally used, the labels "acquaintance" and "stranger" are combined into a single label "other." The SVM can be formulated as the following quadratic programming problem:

$$\min_{(\mathbf{w},b,\xi)\in R^{d+1+n}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \tag{5.1}$$

For C-classification, the constraint is

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \text{ for } \xi_i \geq 0, \text{ and } i = 1, 2, \ldots, n \tag{5.2}$$

For nu-classification, the 1 on the right side of inequality is replaced with 0.



Note that $\|\mathbf{w}\|^2$ is also written as $\mathbf{w}^\top\mathbf{w}$ in matrix notations; $\langle \mathbf{w}, \mathbf{w} \rangle$ in terms of vectors.

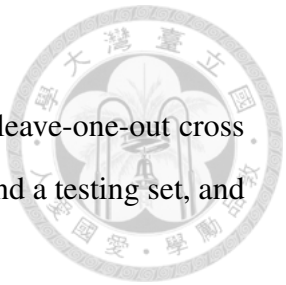
The idea is to find a separating hyperplane, $\mathbf{w}^\top\mathbf{x}_i + b = 0$ which can separate the positive and negative instances. Maximizing the margin $\frac{2}{\|\mathbf{w}\|_2}$ is equivalent to minimizing $\frac{1}{2}\|\mathbf{w}\|^2$. The variables $\mathbf{w}, b$ are subject to the constraint equation (5.2), which aims to classify every data correctly if possible. For every misclassified data point $\mathbf{x_i}$, a penalty $\xi_i \geq 0$ is added to the corresponding equation. In this way, the objective function maximizes the margin and minimizes the sum of penalties simultaneously.

For the hyperplane $\mathbf{w}^\top\mathbf{x}_i + b = 0$, the inner product $\langle \mathbf{w}, \mathbf{x} \rangle$ can be defined as a kernel function. Set $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x_i}$, and $\langle \mathbf{w}, \mathbf{x} \rangle = \sum_{j=1}^{n}\sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x_i}, \mathbf{x_j} \rangle$. There are various kernel functions, and the radial basis function allows us to replace $\langle \mathbf{x_i}, \mathbf{x_j} \rangle$ with kernel $K(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma\|\mathbf{x_i}-\mathbf{x_j}\|_2^2}$. "The radial basis function (RBF) is by far the most popular choice of kernel types used in Support Vector Machines. This is mainly because of their localized and finite responses across the entire range of the real x-axis." [22]

In the experiment, the options of SVM are set to nu-classification and RBF kernel. Other parameters are automatically tuned by the R software.

5.2 Cross Validation

After generating a statistical model, we implemented cross validation to avoid overfitting. Cross validation is the technique where a sample is randomly divided into at least two subsets, and test results are validated by comparing across sub-samples. The goal of this method is to verify if the result is replicable when the model tests some unseen datapoints [37]. One round of cross validation involves testing one subset and training the model with the remaining ones. To reduce variability, multiple rounds of cross validation are performed by allowing the subsets to "take turns" being the testing set [28]. Finally, every sample has been tested by the model, and we can analyze the results of the whole dataset.



Two commonly used techniques are 10-fold cross validation and leave-one-out cross validation. The two techniques both separate data into a training set and a testing set, and are introduced as follows:

1. **10-fold cross validation**: Randomly separate the data into 10 parts of the same size; one part is for testing, and the others are for training. Repeat the training and testing process for 10 times, so each part has the chance to be tested by the model generated by other parts.

2. **Leave-one-out cross validation**: This method is also known as n-fold cross validation, which is similar to the 10-fold one, but the data is separated until every part contains only one sample. The training and testing process is repeated until every sample has already been tested.

Since the dataset contains only 163 samples, we used leave-one-out cross validation to check the accuracy of the model. ”This procedure is especially useful when the dispersion of the distribution is wide or extreme scores are present in the data set.” [37] Details of model validation will be discussed in the next chapter.

5.3 Structure of Data Results

The data results are listed in a confusion matrix, and an example is shown in Table 5.1. When a sample is ”positive,” it means that the corresponding account is actually used by a non-owner, and a ”negative” sample implies the account is used by the owner himself/herself.

Taking the testing results into consideration, each sample falls in one of the four types listed below [18]:

1. True Positive (TP): An account sample is used by a person other than its owner and there is account misuse detected by the system.

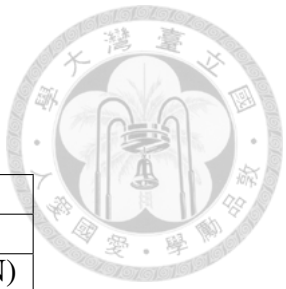


Table 5.1: The confusion matrix

	Prediction	
Truth	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

2. True Negative (TN): An account sample is used by its owner and no misuse is detected by the system.

3. False Positive (FP): An account sample is used by its owner, but the system mistakenly detects it to be used by others.

4. False Negative (FN): An account sample is used by a person other than its owner, but the system wrongly regards it as a normal user.

Moreover, the three statistical indices are used to describe the data results:

1. Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

   Accuracy is the overall performance of the model, considering all positive and negative instances.

2. False Positive Rate (FPR) = $\frac{FP}{TN+FP}$

   False positive rate is the likelihood of misclassifying a negative sample, i.e. an account used by its owner, into a positive one. When this rate is too high, the users receiving false alarms all the time will be annoyed.

3. False Negative Rate (FNR) = $\frac{FN}{TP+FN}$

   False negative rate is the likelihood of misclassifying a positive sample, i.e. an account used by a non-owner, into a negative one. When this rate is too high, the system cannot catch most of the misused accounts.

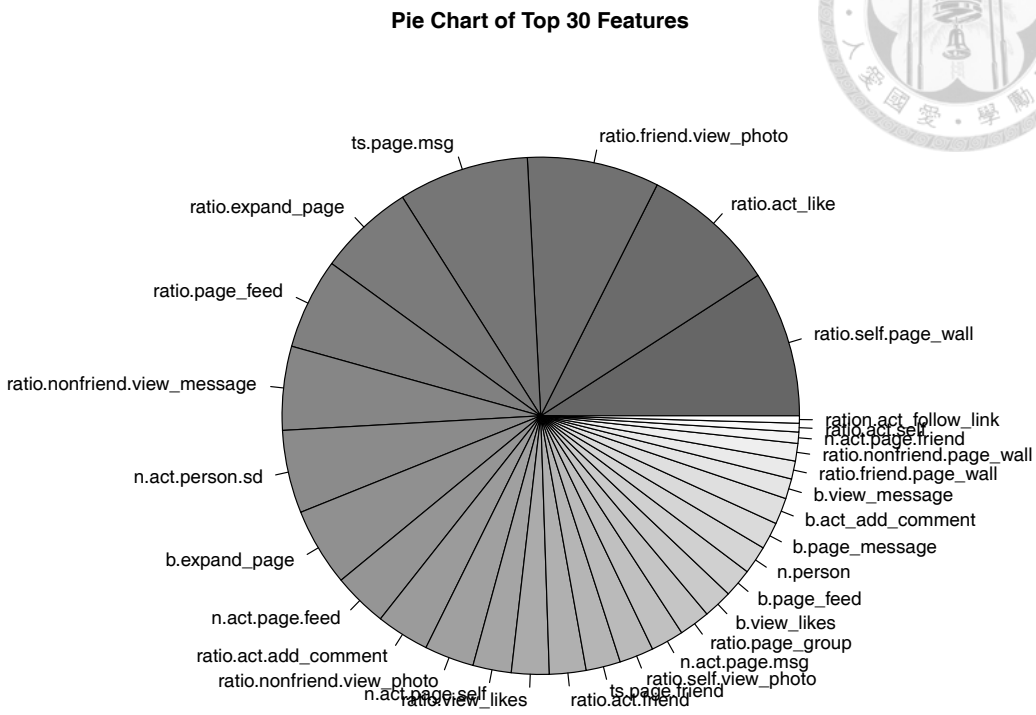


Figure 5.2: Top 30 significant features

5.4 Primary Methods and Results

In the beginning, we were curious about which few variables are more dominant in the whole dataset, but there seems to be no such group. Each feature contributes to the result (an account being classified as used by self or other) with a small percentage, and from the pie chart of top 30 significant features in Fig. 5.2 (omitting the remaining features), none of them accounts for more than 25% of the decision value. For the definition of the features, please refer to Appendix C. "Significance" is defined as the absolute value of the coefficient to the feature variable with scaling. Therefore, instead of seeking for single feature explanation, we decided to concentrate on discovering new features and improving the accuracy of model validation results.

In this research, SVM is adopted for classification, with parameters set to nu-classification and RBF kernel. Leave-one-out cross validation is used for accuracy and false positive/negative rate measurements.

5.4.1 Discovering New Features

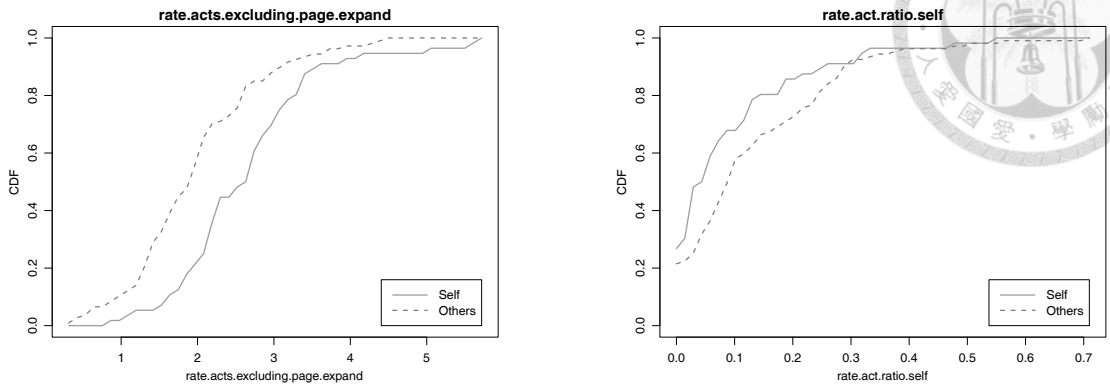
Since the account behavior can be explained in various ways, there always exist new features outside the feature set. A good feature should not only have nonzero values in most of the data, but also have significantly different CDF (Cumulative Distribution Function) curves when the account is used by the owner or others. In this subsection, two new features are introduced based on nonlinear calculation of existing features.

1. **rate.excluding.page.expand** = $\frac{\textbf{the number of actions other than page expansions}}{\textbf{ts.span.minutes}}$

   Page expansion is performed by almost every Facebook account, no matter which webpage the account is browsing. In fact, requesting a new page from the server and expanding a page simply imply that the user is interested in the particular content. In both actions, the user just clicks on the button designed by the server, so they are not directly related to user behavior. Therefore, the feature rate.excluding.page.expand is defined to be the number of actions per minute other than page expansions for the user. The denominator ts.span.minutes is the time span (minutes) of each round, which is set to 30 in this experiment. The CDF is shown in Fig. 5.3(a).

2. **rate.act.ratio.self** = $\frac{\textbf{rate.act.self}}{\textbf{rate.act.self+rate.act.friend+rate.act.nonfriend}}$

   Most Facebook users perform actions mostly on other people's accounts, such as liking a friend's status, commenting on a status, sending message to another person. However, people who do not use their own account tend to get acquainted with it, so they do more actions on the account itself. Due to the previous observations, the feature rate.act.ratio.self is defined to be the time-spending ratio of performing actions on the account's personal page. To avoid division by zero, the denominator is set to the rate of all actions, including actions on self, friends, and non-friends. The CDF of the feature is shown in Fig. 5.3(b).



(a) rate.acts.excluding.page.expand (b) rate.act.ratio.self

Figure 5.3: The CDF of the two new features

Table 5.2: Basic 2-class SVM results

	Prediction		
Truth	Other	Self	Total
Other	103 (63.2%)	4 (2.5%)	107 (65.7%)
Self	21 (12.9%)	35 (21.4%)	56 (34.3%)
Total	124 (76.1%)	39 (23.9%)	163 (100%)

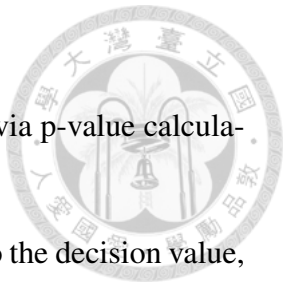
5.4.2 Basic 2-class SVM

The data classification results of using the basic 2-class SVM are summarized in Table 5.2 and the following three indicators.

- Accuracy = 85%
- False positve rate = 38%
- False negative rate = 4%

5.4.3 SVM with P-value Variable Selection

The result in the previous subsection is not satisfying due to the high false positive rate, and one of the most likely reasons is the curse of dimensionality in Machine Learning [27] — some features may be unrelated to the classification. There are 129 features in total



with 86 non-binary ones, and which variables to remove are decided via p-value calculation.

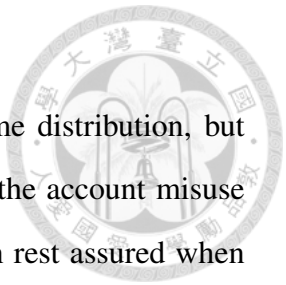
It is observed that different features make different contributions to the decision value, so if a feature shows the same CDF pattern in self-using accounts and other-using ones, the feature should be eliminated. The Wilcoxon Rank-Sum test (a.k.a. Mann-Whitney U test) was performed to test if the feature value of accounts used by self and others are not equally distributed, so we applied this method to calculate the p-value. Note that Wilcoxon signed-rank test is a paired difference test, and K-S (Kolmogorv-Smirnov) test cannot compute duplicate values in the samples, so both of them are not suitable for the data analysis of this experiment.

The null hypothesis H_0 and the alternative hypothesis H_1 are defined as follows:

- H_0: The distributions of both groups of feature values are equal.
- H_1: The probability of an observation from one population exceeding an observation from the second population is not equal to 50% [29].

If H_1 actually holds, the corresponding feature is very likely to have different behavior distributions in self-used and other-used accounts. "In statistical significance testing, the p-value is the probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true." [30]

For each feature, if the corresponding p-value is less than a certain threshold, the null hypothesis H_0 is rejected and the alternative hypothesis H_1 is accepted. It is inferred that the feature shows different patterns in accounts used by its owner or other people. Due to the small sample size, keeping more features is desirable in order to avoid overfitting in future datasets. Therefore, the upper limit of p-values was set to 12% instead of 10% or 5%. Although there may be more features staying in the dataset, only 36 of the 86 non-binary features had p-values below the threshold. In fact, some features have p-values as high as 80%, so they are obviously not related to the account misuse detection. A



low p-value does not imply that the two groups come from the same distribution, but a high p-value is a good indicator for the feature to be unrelated to the account misuse classification. For the features with extremely high p-values, we can rest assured when removing them.

Table 5.3 lists these useful features in ascending order of p-values, and the definitions are listed in Appendix A. It is observed that if people are using their own accounts, they are more active in clicking on ”like” or commenting, and they view less personal messages or private clubs. Note that the extra features added in Section 5.4.3 both have low p-values, implying great statistical significance.

By using SVM with p-value variable selection, the accuracy was improved from 85% to 90%, and the false positive rate was reduced from 38% to 23%. The results are shown in Table 5.4 and the following three indicators. This 79-variable set consisting of 36 non-binary variables will be used in the all the succeeding sections. For the remaining 43 binary variables, the definitions are listed in Appendix B.

- Accuracy = 90%
- False Positive Rate = 23%
- False Negative Rate = 3%

5.5 Secondary Methods and Results

The 23% false positive rate is not satisfying, because this implies one out of five on average Facebook account owners receive a stealthy use notice from the server, while nothing serious has happened. Accordingly, secondary methods are needed. The variable set with 36 non-binary features is preserved, but more statistical knowledge is involved in this stage. Weight adjustment, oversampling, and 3-class SVM were implemented on the original dataset. Oversampling and 3-class SVM both perform very well in accuracy with leave-one-out cross validation.

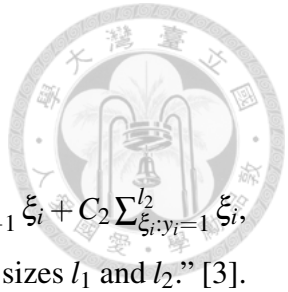
Table 5.3: The 36 non-binary features with p-values less than 12% (in ascending order of p-values)

Number	Feature Name	Number	Feature Name
1	rate.**act_like**	19	rate.view_likes
2	rate.friend.**act_like**	20	ts.page.msg
3	rate.nonfriend.**act_like**	21	**rate.act.ratio.self**
4	rate.**act_add_comment**	22	n.person
5	**rate.acts.excluding.page.expand**	23	rate.act.non.expand.page.public
6	rate.act.non.expand.page.feed	24	rate.act.page.public
7	rate.expand_comments	25	rate.self.page_wall
8	rate.friend.expand_comments	26	rate.act.nonfriend
9	rate.page_group	27	ts.page.feed
10	rate.acts	28	rate.act.self
11	rate.page_message	29	rate.page_feed
12	rate.act.page.feed	30	rate.self.page_friends
13	rate.act.non.expand.page.msg	31	rate.expand_page
14	rate.nonfriend.expand_comments	32	rate.self.act_like
15	n.person.act.more.than.one	33	ts.page.self
16	rate.act.page.msg	34	ts.page.public
17	rate.friend.view_likes	35	rate.act.non.expand.page.self
18	rate.act.friend	36	rate.act.page.self

Table 5.4: SVM with p-value variable selection results

	Prediction		
Truth	Other	Self	Total
Other	104 (63.8%)	3 (1.9%)	107 (65.7%)
Self	13 (7.9%)	43 (26.4%)	56 (34.3%)
Total	117 (71.7%)	46 (28.3%)	163 (100%)

5.5.1 Weight Adjustment



"For SVM that minimizes the objective function $\frac{1}{2}\|\mathbf{w}\|_2^2 + C_1 \sum_{\xi_i:y_i=-1}^{l_1} \xi_i + C_2 \sum_{\xi_i:y_i=1}^{l_2} \xi_i$, you can choose constants C_1 and C_2 inversely proportional to the class sizes l_1 and l_2." [3]. Since we have 56 self-used and 107 other-used account samples, the class size proportion is approximately 1:2, so the corresponding constants should be set to 2:1. However, the results are exactly the same as the one in Table 5.4, so weight adjustment does not make any contribution to the classification.

5.5.2 Oversampling

Oversampling is to balance the ratio of positive and negative data by duplicating samples of the smaller class. There are 56 self-use and 107 other-use account samples in the dataset, and the ratio of account samples used by its owner and others is 1:2. As a result, the penalty of misclassifying a positive (other) sample is twice as misclassifying a negative (self) one, so the false positive rate is high due to the unbalanced dataset. By duplicating negative instances, we can avoid aliasing and reduce the false positive rate. By randomly picking 5 self-use samples and duplicating the remaining 51, we get $51 \times 2 + 5 = 107$ samples, the same number as the other-use ones.

The average results for 5 times are summarized as follows and in Table 5.5. This is much better than the production of SVM with p-value variable selection in Table 5.4.

- Accuracy = 97%
- False Positive Rate = 2%
- False Negative Rate = 4%

5.5.3 3-class SVM

The basic 2-class SVM can be extended to 3-class SVM by using the *kernlab* package and the function *ksvm* (k-class SVM) in R. The account usage can be classified into self

Table 5.5: SVM with oversampling results

	Prediction		
Truth	Other	Self	Total
Other	103 (48.1%)	4 (1.8%)	107 (50%)
Self	2 (1.0%)	105 (49.1%)	107 (50%)
Total	105 (49.1%)	109 (50.9%)	163 (100%)

and other, and the "other" category can also be divided into acquaintance and stranger usages. Acquaintance usage of an account means that the account was used by the owner's acquaintance in real life, such as a family member or a friend. Since we actually collected traces in these three categories as described in Fig 4.1, it is not difficult to implement 3-class SVM on the dataset.

SVM can be written in a kernel-based algorithmic form: [15]

$$f(\mathbf{x}) = \mathbf{w}^\top \Phi(\mathbf{x}) \text{ for some weight vector } \mathbf{w} \in F \tag{5.3}$$

$$K(\mathbf{x},\mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle; \text{ therefore } f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x_i}, \mathbf{x}) \tag{5.4}$$

For the 2-norm soft margin classification, the optimization problem is:

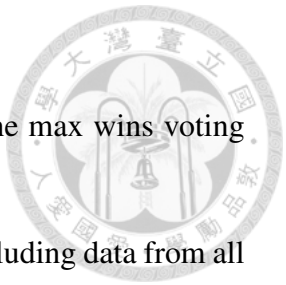
Minimize

$$t(\mathbf{w}, \xi) = \frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{C}{m} \sum_{i=1}^{m} \xi_i \tag{5.5}$$

Subject to

$$y_i(\langle \mathbf{x_i}, \mathbf{w} \rangle + b) \geq 1 - \xi_i, \text{ where } \xi_i \geq 0, \forall i = 1, \ldots, m \tag{5.6}$$

The function *ksvm* includes Sequential Minimization Optimization (SMO), which solves the smallest possible optimization problem involving two elements of α_i because they must obey one linear equality constraint. Then SMO jointly optimizes two α_is at a time, updating the values for SVM, to avoid numerical Quadratic Problem (QP) optimization. For multi-class classification, this method constructs $\binom{k}{2}$ classifiers where each



one is trained on data from two classes, and prediction is done by the max wins voting strategy.

The k-class classification solves a single optimization problem including data from all classes, and the k-class SVM equation form is as follows:

Suppose SVM is trained on the model $y_i = f(\mathbf{x_i}) + \delta_i$,

where δ_i is the independent and identically distributed random noise.

Minimize

$$t(\mathbf{w_n}, \xi) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i \tag{5.7}$$

Subject to

$$\langle \mathbf{x_i}, \mathbf{w_{y_i}} \rangle - \langle \mathbf{x_i}, \mathbf{w_{y_i}} \rangle \geq b_i^n - \xi_i, \text{ where } b_i^n = 1 - \delta_{y_i,n} \tag{5.8}$$

The decision function is $argmax_{m=1,\ldots,k}\langle \mathbf{x_i}, \mathbf{w_n} \rangle$.

Finally, the results of 3-class SVM are shown in terms of accuracy, false positive/negative rate, and a confusion matrix in Table 5.6. Both the acquaintance and stranger categories are regarded as "other" (positive) when calculating the three different rates. Note that if an acquaintance-used account sample is classified as being used by a stranger, it is still considered to be a true negative, and vice versa.

- Accuracy = 95%
- False Positive Rate = 0%
- False Negative Rate = 7%

Compared with oversampling, 3-class SVM also gives satisfying results, especially the zero false positive rate for the dataset. The account samples used by self, an acquaintance, or a stranger are almost of the same quantity, so there is no need to adjust the weights. Note that the false negative rate is higher than the one obtained by oversampling, and it is inferred that the account owner and his/her acquaintance have relatively similar behavior on Facebook because they usually have mutual friends.

Table 5.6: 3-class SVM results

	Prediction			
Truth	Self	Acquaintance	Stranger	Total
Self	56 (34.3%)	0 (0.0%)	0 (0.0%)	56 (34.3%)
Acquaintance	5 (3.1%)	42 (25.8%)	1 (0.6%)	48 (29.5%)
Stranger	3 (1.9%)	1 (0.6%)	55 (33.7%)	59 (36.2%)
Total	64 (39.3%)	43 (26.4%)	56 (34.3%)	163 (100%)

Chapter 6

Model Validation and Discussion

Model validation is to verify that the statistical model can be applied to unseen data, so the given dataset should be separated into two parts: training and testing. In Section 5.2, we used leave-one-out cross validation because it is an all-purpose statistical tool [37], especially when the dataset is not large. However, since our dataset consists of 163 samples, it is not too small to separate into training and testing parts, which is the fundamental validation method. The proportion of class sizes for training and testing is 2:1. We used the stratified sampling strategy when splitting the dataset, so the ratio of self-used and other-used account samples is 1:2, the same as the original one.

6.1 Separate Training and Testing Dataset Results

The results of separate training and testing datasets are presented in Table 6.1.

1. The weight adjustment method predicted almost the whole testing set to be accounts used by others.

2. The oversampling method dealt with an enlarged dataset with the same number of self-used and other-used account samples. It predicted all the positive data correctly, but there is 50% likelihood for a self-used account to be wrongly identified.

Table 6.1: Cross validation results: separate training and testing datatsets

	Weight adjustment	Oversampling	**3-class SVM**
Accuracy	62%	79%	**75%**
False Positive Rate	100%	43%	**11%**
False Negative Rate	3%	0%	**32%**

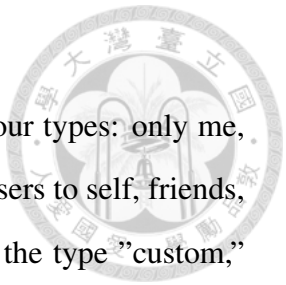
3. For 3-class SVM, the calculation of the three statistical indices is described in Section 5.5.3. This method produced a relatively good result.

6.2 Method Selection

In fact, 3-class SVM performs the best because the false positive rate is more important than the false negative rate. False positive means that an ordinary user logs in his/her own account but receives an account misuse notification from the server, and this can be very annoying. False negative is the case of not catching a stealthy use account. It seems like a false negative is a more serious error, but false positives should be emphasized on more. If the server does nothing in classification, the false negative rate will be 100%. Therefore, any decreased false negative rate makes a contribution to the security. However, when it comes to availability, it is more important to reduce the false positive rate. Although there is a probability of 32% for 3-class SVM to allow a stealthy use account, only one of ten users on average receive a false alarm. This is much better than the weight adjustment and oversampling schemes.

6.3 Explanation of Results

The 3-class SVM is a more appropriate scheme than the 2-class SVM in our experiment for some reasons. To begin with, since there are usually many mutual friends between acquaintances and the account owner, so what they are concerned about can be very similar in Facebook. To avoid ambiguity, acquaintances using the account should be set to a



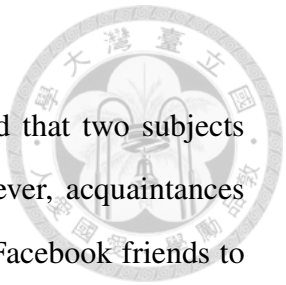
single label. Moreover, the privacy settings of Facebook consist of four types: only me, friends, public, and custom [1]. This implies Facebook initially sets users to self, friends, and strangers. Although the type ”friends of friends” is included in the type ”custom,” many friends of friends can be strangers to the account owner, and this can be inferred by the Six Degrees of Separation [31]. In this way, it is more appropriate to divide the users into three groups: self, acquaintances, and strangers, instead of two or four groups.

6.4 Security Analysis

Our detection scheme for stealthy account users can be applied to the online social network server, and it is independent of any cryptographic technology or specific detection model. The scheme is completely based on user behavior, and it can all be performed on the server's side. In this way, the attackers cannot evade our detection scheme without hacking into the server to modify the model. The model can be trained with known account user samples beforehand, and then the server can use the model to classify online accounts. When there is any account misuse suspected, the server can perform secondary security measures such as notifying the owner via email or mobile phone and advising him/her to change the password.

6.5 Limitations to this Research

Although the research has reached its aims, there are several limitations that may affect the account misuse detection results. First, subjects all knew they were doing an experiment, so they may behave differently from using Facebook in daily life. Second, the rules we set to the subjects may also affect their behavior. To avoid legal issues, we instructed the subjects not to make personal attacks when using a stranger's account. Third, we concentrate on the selection of non-binary features, so the whole binary feature set is kept unchanged, and it may contain unrelated features. Finally, for convenience, the terms



”acquaintance” and ”friends” are used alternately, and it is assumed that two subjects acquainted with each other are Facebook friends. In real life, however, acquaintances may not be friends on Facebook, and some people add strangers as Facebook friends to play online games. To some extent, the world in an online social network is different from the real society.

Chapter 7

Conclusion

In the thesis, we have proved Support Vector Machine (SVM) to be a feasible solution for account misuse detection on Facebook; that is, an account is used by a person other than its owner. We tested several methods and found that 3-class SVM performs the best, classifying account users into its owner, acquaintances, and strangers. The selection criteria comprise false positive rate, false negative rate, and accuracy based on cross validation. There is not a dominating behavior feature for account misuse detection, but we discovered that accounts being used stealthily tend to be ”quieter,” i.e. they click on ”like” and comment less frequently, but they view more personal messages and private clubs. Due to the popularity of online social networks, there are promising and related topics for future work, such as early account misuse detection for just one minute and the development of personalized owner behavior model, to name a few.

References

[1] Choose who you share with. http://www.facebook.com/help/459934584025324/. Online; accessed 2013.

[2] Class adaboostm1. http://weka.sourceforge.net/doc/weka/classifiers/meta/AdaBoostM1.html. Online; accessed 2013.

[3] A priori selection of svm class weights. http://stats.stackexchange.com/questions/24959/a-priori-selection-of-svm-class-weights. Online; accessed 2012.

[4] Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, pages 1–9, 2005.

[5] Beomjoon Kim Boris Babenko, Abhishek Shivkumar. For what kind of classification problems is svm a bad approach? http://www.quora.com/Support-Vector-Machines/For-what-kind-of-classification-problems-is-SVM-a-bad-approach, 2013. Online; accessed 2013.

[6] Josh Constine. Facebook has users identify friends in photos to verify accounts, prevent unauthorized access. http://www.insidefacebook.com/2010/07/26/facebook-photos-verify/, 2010. Online; accessed 2013.

[7] Hanif Mohaddes Deylami and Yashwant Prasad Singh. Cybercrime detection techniques based on support vector machines. *Artificial Intelligence Research*, 2(1):1–12, 2012.

[8] Dominique Gay et. al. Can we say that svm is the best classifier to date? `http://www.researchgate.net/post/Can_we_say_that_SVM_is_the_best_classifier_to_date14`, 2012. Online; accessed 2013.

[9] Lenny Zeltser (GIAC Security Expert). Beyond logins: Continuous and seamless user authentication. `http://blog.zeltser.com/post/41275913909/continuous-user-authentication`, 2013. Online; accessed 2013.

[10] Facebook. Facebook security. `http://www.facebook.com/security`. Online; accessed 2012.

[11] A. Felt and D. Evans. Privacy protection for social networking apis. *2008 Web 2.0 Security and Privacy (W2SP'08)*, 2008.

[12] L. Gyarmati and T.A. Trinh. Measuring user behavior in online social networks. *Network, IEEE*, 24(5):26–31, 2010.

[13] Jianming He, Wesley Chu, and Zhenyu Liu. Inferring privacy information from social networks. *Intelligence and Security Informatics*, pages 154–165, 2006.

[14] Plurilock Security Solutions Inc. What is continuous authentication? `http://www.plurilock.com/blog/what-continuous-authentication`, 2011. Online; accessed 2013.

[15] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab-an s4 package for kernel methods in r. 2004.

[16] Paul Mah. Stored passwords add to mobile security risks. `http://www.itbusinessedge.com/cm/blogs/mah/`

stored-passwords-add-to-mobile-security-risks/?cs=47183, 2011. Online; accessed 2013.

[17] S. Mahmood and Y. Desmedt. Your facebook deactivated friend or a cloaked spy. In *10th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 367–373. IEEE, 2012.

[18] Lee Newberg. Some useful statistics definitions. http://www.cs.rpi.edu/~leen/misc-publications/SomeStatDefs.html, 2005. Online; accessed 2013.

[19] Koichiro Niinuma and Anil K Jain. Continuous user authentication using temporal information. *Defense, Security, and Sensing*, 7667:76670L, 2010.

[20] Koichiro Niinuma, Unsang Park, and Anil K Jain. Soft biometric traits for continuous user authentication. *IEEE Transactions on Information Forensics and Security*, 5(4):771–780, 2010.

[21] SJ Shepherd. Continuous authentication by analysis of keyboard typing characteristics. 1995.

[22] StatSoft. Support vector machines (svm). http://www.statsoft.com/textbook/support-vector-machines/. Online; accessed 2012.

[23] S Sudarvizhi and S Sumathi. A review on continuous authentication using multimodal biometrics. *International Journal of Emerging Technology and Advanced Engineering*, 3(1), 2013.

[24] Cong Tang, Keith Ross, Nitesh Saxena, and Ruichuan Chen. What's in a name: A study of names, gender inference, and gender behavior in facebook. *Database Systems for Adanced Applications*, pages 344–356, 2011.

[25] Credant Technologies. Phone data makes 4.2 million* brits vulnerable to id theft. http://www.credant.com/news-a-events/press-releases/

69-phone-data-makes-42-million-brits-vulnerable-to-id-theft.html. Online; accessed 2013.

[26] Telerik. Fiddler web debugger. http://www.fiddler2.com/fiddler2/. Online; accessed 2012.

[27] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for svms. *Advances in neural information processing systems*, pages 668–674, 2001.

[28] Wikipedia. Cross-validation (statistics). http://en.wikipedia.org/wiki/Cross-validation_(statistics). Online; accessed 2013.

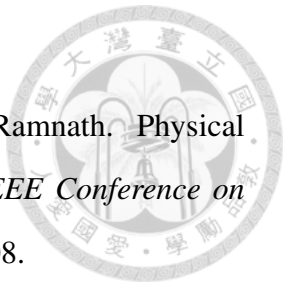
[29] Wikipedia. Mann–whitney u. http://en.wikipedia.org/wiki/Mann\%E2%80%93Whitney_U. Online; accessed 2012.

[30] Wikipedia. P-value. http://en.wikipedia.org/wiki/P-value. Online; accessed 2012.

[31] Wikipedia. Six degrees of separation. http://en.wikipedia.org/wiki/Six_degrees_of_separation. Online; accessed 2013.

[32] C. Wilson, B. Boe, A. Sala, K.P.N. Puttaswamy, and B.Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European conference on Computer systems*, pages 205–218. ACM, 2009.

[33] R. Wishart, D. Corapi, A. Madhavapeddy, and M. Sloman. Privacy butler: A personal privacy rights manager for online presence. In *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 672–677. IEEE, 2010.



[34] Roland HC Yap, Terence Sim, Geraldine XY Kwang, and R Ramnath. Physical access protection using continuous authentication. In *2008 IEEE Conference on Technologies for Homeland Security*, pages 510–512. IEEE, 2008.

[35] Sausan Yazji, Xi Chen, Robert P Dick, and Peter Scheuermann. Implicit user re-authentication for mobile devices. In *Ubiquitous Intelligence and Computing*, pages 325–339. Springer, 2009.

[36] Alyson L Young and Anabel Quan-Haase. Information revelation and internet privacy concerns on social network sites: a case study of facebook. In *Proceedings of the fourth international conference on Communities and technologies*, pages 265–274. ACM, 2009.

[37] Chong Ho Yu. Resampling methods: concepts, applications, and justification. *Practical Assessment, Research & Evaluation*, 8(19):1–23, 2003.

[38] 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 *Proceedings of the 18th international conference on World wide web*, pages 531–540. ACM, 2009.

Appendix A

The Most Important 36 Features

This is the variable set consisting of 36 non-binary features with p-values less than 12%, in ascending order of p-values. This variable set is used over and over again in the paper, starting from Section 5.4.3.

1. **rate.act_like**: How many likes per minute the user gives; that is, the number of "likes" given by the user divided by the total time span (minutes).

2. **rate.friend.act_like**: Similar to rate.act_like, but only likes to friends' status or pages are calculated.

3. **rate.nonfriend.act_like**: Also similar to rate.act_like, but only likes to non-friends' status or pages are calculated.

4. **rate.act_add_comment**: How many comments per minute the user makes; that is, the number of comments given by the user divided by the total time span (minutes).

5. **rate.acts.excluding.page.expand**: How many actions other than expanding pages per minute the user does; that is, the number of actions other than expanding pages given by the user divided by the total time span (minutes).

6. **rate.act.non.expand.page.feed**: How many actions done by the user per minute and not related to page expand. If a user expands a page and then comments on the expanded part, this will not be calculated in this.

7. **rate.expand_comments**: How many times the user expands to read the whole comments per minute.

8. **rate.friend.expand_comments**: Similar to rate.expand_comments, but only when the user expands friends' comments is the rate calculated.

9. **rate.page_group**: How many times the user goes into a group or club on Facebook per minute, requesting a new page from the server.

10. **rate.acts**: The total actions the user does per minute.

11. **rate.page_message**: The total message pages read by the user per minute.

12. **rate.act.page.feed**: The total pages of news feed read by the user per minute. Each time a user comes to the news feed from another page, 1 is added to act.page.feed.

13. **rate.act.non.expand.page.msg**: How many actions not related to expanding page messages per minute the user does. If a user expands a message page and then comments on the expanded part, this will not be calculated in this.

14. **rate.nonfriend.expand_comments**: How many times the user expand comments in a non-friend page per minute.

15. **n.person.act.more.than.one**: The number of target people the user interacts with during the session, and this feature only considers the case of more than one person.

16. **rate.act.page.msg**: How many times per minute the user requests a message page from the server.

17. **rate.friend.view_likes**: How many times per minute the user views who likes a particular status on a friend's page.
18. **rate.act.friend**: How many times per minute the user performs an action on a friend, including giving likes and commenting.
19. **rate.view_likes**: How many times per minute the user views who likes a particular status on any page.
20. **ts.page.msg**: The time span of the user's staying on the message page.
21. **rate.act.ratio.self**: The percentage that actions to the user himself/herself accounts for all actions.
22. **n.person**: The number of target person(s) the user interacts with during the session, including one to one interaction.
23. **rate.act.non.expand.page.public**: How many times per minute the user performs an action on a public page without expanding it.
24. **rate.act.page.public**: How many times per minute the user performs any action on a public page.
25. **rate.self.page_wall**: How many times per minute the user goes to his/her own page wall.
26. **rate.act.nonfriend**: Similar to rate.act.friend, but only actions on non-friends are calculated.
27. **ts.page.feed**: The time span of the user's staying on the news feed page.
28. **rate.act.self**: Similar to rate.act.friend, but only actions toward the user himself/herself is counted.
29. **rate.page_feed**: How many times per minute the user requests for the news feed.

30. **rate.self.page_friends**: How many times per minute the user reads his/her own friends list.

31. **rate.expand_page**: How many times per minute the user expands a page.

32. **rate.self.act_like**: How many times per minute the user gives a ”like” on his/her personal page.

33. **ts.page.self**: The time span of the user’s staying on his/her personal page.

34. **ts.page.public**: The time span of the user’s staying on a public page.

35. **rate.act.non.expand.page.self**: How many times per minute the user performs an action on himself/herself but not related to expanding pages.

36. **rate.act.page.self**: How many times per minute the user performs actions requesting a new page on his/her personal page.

Appendix B

All 43 Binary Features

There are 43 binary features in total, and their names start with ”b.” They are used to record whether the user performs a certain action in the measured time span. When the user does, the value is 1; otherwise, the value is 0. According to the target of the actions, these features can be divided into four categories: general, self, friend, and non-friend.

I. General

All listed actions, no matter who the target is, are counted in this kind of features.

1. **b.act_add_comments**: Whether the user adds a comment
2. **b.act_delete_comment**: Whether the user deleted a comment
3. **b.act_follow_link**: Whether the user follows an external link
4. **b.act_like**: Whether the user gives a ”like”
5. **b.expand_comments**: Whether the user expands a comment
6. **b.expand_page**: Whether the user expands a page
7. **b.page_fbpage**: Whether the user reads a personal page on Facebook
8. **b.page_feed**: Whether the user reads the news feed
9. **b.page_friends**: Whether the user reads his/her friends' personal pages

10. **b.page_group**: Whether the user reads a page in a group or club
11. **b.page_message**: Whether the user reads a message page
12. **b.page_notes**: Whether the user reads a note page, which is like a blog article
13. **b.page_photos**: Whether the user reads the photo list page
14. **b.page_wall**: Whether the user goes to his/her personal page (wall)
15. **b.view_likes**: Whether the user views who gave "likes" to a certain page or status
16. **b.view_message**: Whether the user views a message through a hovering entity
17. **b.view_photo**: Whether the user views a photo by zooming in

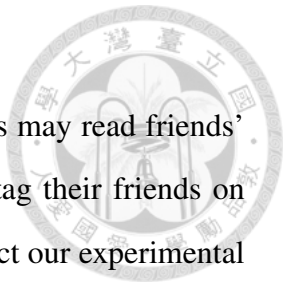
II. Self

Only actions whose target is the user himself/herself are counted.

1. **b.self.act_like**: Whether the user gives a "like" to the content generated by himself/herself
2. **b.self.expand_comments**: Whether the user expands a comment regarding to his/her posts
3. **b.self.page_friends**: Whether the user reads his/her friends list
4. **b.self.page_notes**: Whether the user reads his/her own notes
5. **b.self.page_photos**: Whether the user reads the photo list page
6. **b.self.page_wall**: Whether the user goes to his/her personal page (wall)
7. **b.self.view_card**: Whether the user views a hover card related to himself/herself
8. **b.self.view_likes**: Whether the user views who gave "likes" to his/her own status
9. **b.self.view_photo**: Whether the user views his/her own photo by zooming in

III. Friend

Only actions whose target is the user's friends are counted in this type of features.



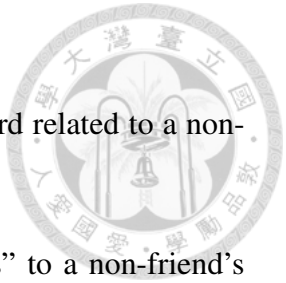
We did not set a feature of reading friends' notes because users may read friends' blog articles more often than notes. Moreover, some people tag their friends on the notes to force the friends read their notes, and this may affect our experimental results.

1. **b.friend.act_like**: Whether the user gives a "like" to a friend
2. **b.friend.expand_comments**: Whether the user expands a comment regarding to a friend's post
3. **b.friend.page_friends**: Whether the user reads his/her friend's friend list
4. **b.friend.page_wall**: Whether the user goes to his/her friend's personal page (wall)
5. **b.friend.view_card**: Whether the user views a hover card related to a friend
6. **b.friend.view_likes**: Whether the user views who gave "likes" to his/her friend's status
7. **b.friend.view_message**: Whether the user views a message from a friend
8. **b.friend.view_photo**: Whether the user views a friend's photo by zooming in

IV. Nonfriend

In this category, the target of actions should not be the user himself/herself or his/her friends. Users may or may not know the "non-friend."

1. **b.nonfriend.act_like**: Whether the user gives a "like" to a non-friend
2. **b.nonfriend.expand_comments**: Whether the user expands a comment regarding to a non-friend's post
3. **b.nonfriend.page_friends**: Whether the user reads a non-friend's friend list
4. **b.nonfriend.page_notes**: Whether the user reads a non-friend's notes
5. **b.nonfriend.page_wall**: Whether the user goes to a non-friend's personal page (wall)



6. **b.nonfriend.view_card**: Whether the user views a hover card related to a non-friend
7. **b.nonfriend.view_likes**: Whether the user who gave ”likes” to a non-friend’s status
8. **b.nonfriend.view_message**: Whether the user views a message from a non-friend
9. **b.nonfriend.view_photo**: Whether the user views a non-friend’s photo by zooming in

Appendix C

Initial Top 30 Features

This is the initial result of our experiment on Facebook account misuse detection. Please refer to Fig. 5.2 for the top 30 features discovered at the beginning. The following is the meaning of the selected features:

1. **ratio.self.page_wall**: How many times per minute the user posts something on his/her own page wall.
2. **ratio.act_like**: How many likes per minute the user gives.
3. **ratio.friend.view_photo**: How many times per minute the user views a friend's photo by zooming in.
4. **ts.page.msg**: The time span of the user's staying on the message page.
5. **ratio.expand_page**: How many times per minute the user expands a page.
6. **ratio.page_feed**: How many times per minute the user requests for the news feed.
7. **ratio.nonfriend.view_message**: How many times per minute the user views a message from a non-friend.
8. **n.act.person.sd**: The standard deviation of the number of target person(s) who the user interacts with during the session, including one to one interaction.

9. **b.expand_page**: Whether the user expands a page.
10. **n.act.page.feed**: The number of times when the user requests a new page from the server.
11. **ratio.act.add_comment**: How many comments per minute the user makes; that is, the number of comments given by the user divided by the total time span (minutes).
12. **ratio.nonfriend.view_photo**: How many times per minute the user views a non-friend's photo by zooming in.
13. **n.act.page.self**: The number of times when the user performs actions requesting a new page on his/her personal page.
14. **ratio.view_likes**: How many times per minute the user views who gave "likes" to a certain page or status.
15. **ratio.act.friend**: How many times per minute the user performs an action on a friend, including giving "likes" and commenting.
16. **ts.page.friend**: The time span of the user reads a friends list page.
17. **ratio.self.view_photo**: How many times per minute the user views his/her own photo by zooming in.
18. **n.act.page.msg**: The number of times when the user goes to the message page.
19. **ratio.page_group**: How many times the user goes into a group or club on Facebook per minute, requesting a new page from the server.
20. **b.view_likes**: Whether the user view who gave "likes" to a certain page or status.
21. **b.page_feed**: Whether the user reads the news feed.

22. **n.person**: The number of target person(s) the user interacts with during the session, including one to one interaction.

23. **b.page_message**: Whether the user reads a message page.

24. **b.act_add_comment**: Whether the user adds a comment.

25. **b.view_message**: Whether the user views a message through a hovering entity.

26. **ratio.friend.page_wall**: How many times per minute the user goes to his/her friend's page wall.

27. **ratio.nonfriend.page_wall**: How many times per minute the user goes to a non-friend's page wall.

28. **n.act.page.friend**: The number of times when the user reads a friends list page.

29. **ratio.act.self**: How many times per minute the user performs an action to himself/herself.

30. **ration.act.follow_link**: How many times per minute the user follows an external link.