Defending Imitating Attacks in Web Credibility Evaluation Systems

Xin Liu
École Polytechnique Fédérale de Lausanne
Batiment BC, Station 14
1015 Lausanne, Switzerland
x.liu@epfl.ch

Radoslaw Nielek
Polish-Japanese Institute of Information Technology
Warsaw, Poland
radek@post.pl

Adam Wierzbicki
Polish-Japanese Institute of Information Technology
Warsaw, Poland
adamw@pjwstk.edu.pl

Karl Aberer
École Polytechnique Fédérale de Lausanne
Batiment BC, Station 14
1015 Lausanne, Switzerland
karl.aberer@epfl.ch

ABSTRACT

Unlike traditional media such as television and newspapers, web contents are relatively easy to be published without being rigorously fact-checked. This seriously influences people's daily life if non-credible web contents are utilized for decision making. Recently, web credibility evaluation systems have emerged where web credibility is derived by aggregating ratings from the community (e.g., MyWOT). In this paper, We focus on the robustness of such systems by identifying a new type of attack scenario where an attacker imitates the behavior of trustworthy experts by copying system's credibility ratings to quickly build high reputation and then attack certain web contents. In order to defend this attack, we propose a two-stage defence algorithm. At stage 1, our algorithm applies supervised learning algorithm to predict the credibility of a web content and compare it with a user's rating to estimate whether this user is malicious or not. In case the user's maliciousness can not be determined with high confidence, the algorithm goes to stage 2 where we investigate users' past rating patterns and detect the malicious one by applying hierarchical clustering algorithm. Evaluation using real datasets demonstrates the efficacy of our approach.

Categories and Subject Descriptors

K.6 [**Management of Computing and Information Systems**]: Security and Protection

Keywords

Web Credibility, Imitating Attack, Robustness, Machine Learning

1. INTRODUCTION

Nowadays, people increasingly rely on the Internet to seek information and communicate. In particular, with the advent of Web 2.0 era, people not only extract knowledge from various information sources, but also contribute and share their own generated contents. However, web contents are typically published without being seriously fact-checked, thus greatly influencing people's daily life if non-credible web contents are relied on to make decisions. For instance, in online Q&A systems, anyone can post their answers without beforehand validation, and it is questioner's own responsibility in judging the credibility of the answers. Therefore, the issue of assessing web credibility becomes of crucial importance.

In practice, several community based web content credibility evaluation systems have emerged recently, assessing credibility by aggregating users' personal opinions. For instance, MyWOT (http://www.mywot.com/) aggregates individual users' ratings on four aspects of web credibility: Trustworthiness, Vendor reliability, Privacy and Child Safety. In academia, similar systems were also proposed, improving the commercial counterparts from different aspects [15, 13]. The performance of such systems is greatly influenced by the reliability of users' ratings. In particular, when a page has few ratings[1], by colluding, attackers' ratings can easily deviate the real credibility of this page. In order to address this issue, reputation systems are employed to assign reputation score to each user based on the reliability of this user's past ratings.

In this paper, we focus on the robustness of a community based web credibility evaluation system (e.g., MyWOT). We first identify a new attack that can easily cheat traditional reputation systems: the attacker queries the credibility of a web page from the system, then he/she copies system's rating and submits it (probably with minor modifications) to the system again as his/her own contributions. From the perspective of the system, the attacker behaves quite similarly with the highly reputable users (i.e., providing genuine ratings). In this way, the attacker can easily gain high reputation or achieve high level roles, e.g., *expert*. For instance, MyWOT varies the reputation of a user by: "When you start using WOT, your ratings have little weight, but if you keep rating sites consistently, your ratings will be considered more reliable over time" [1]. The main advantage of such *imitating attack* is that the attacker does not need to dig out any special knowledge but just queries and re-sends the system's credibility information, thus greatly reducing costs (e.g., cheap for both individual attackers and collusion). For instance, one of the authors of this paper copied over 200 MyWOT's ratings in four days (became bronze member) and then attacked several *unrated* (i.e., system's ratings are unavailable) pages by issuing lowest ratings without being detected by the system[2]. We thus claim that imitating attack is potentially very harmful to web credibility evaluation systems like MyWOT, and in particular, if multiple at-

[1] This is very common since majority of web pages are not widely popular, thus are not rated by many users.

[2] We have deleted all relevant ratings of this experiment.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

tackers collaboratively attack certain web pages, system's ratings will be easily manipulated by malicious users.

In order to handle such imitating attack, we propose a two-stage defense algorithm. The main contributions of this work are summarized as follows: (1) At stage 1 of our algorithm, we study various characteristics of a web page, which we believe are correlated with the credibility of this page. By applying supervised machine learning algorithm, we predict the target page's credibility and compare the predicted rating with a user's rating to infer the possibility of the user's behavior change (i.e., attack). (2) If the user cannot be inferred to be malicious or genuine with *high* confidence at stage 1, the algorithm goes to stage 2 where we further predict whether the user is an imitating attacker or not by investigating his/her past rating patterns. Three patterns are identified: (i) rating category. In order to attack web contents in specific categories, an attacker may gain category-aware reputation by intensively rating pages in certain categories, but genuine users have no such constraints in general (i.e., they have broader interests in many categories). (ii) Rating frequency. Due to resources constraints (e.g., time), an attacker may frequently submit ratings within a short period of time to quickly build his/her reputation, while genuine users submit ratings whenever they want. (iii) Rating reliability. Due to the personalized view on the web credibility, genuine users' ratings may sometimes be inconsistent with system's ratings, but for imitating attackers, in order to gain high reputation as soon as possible, their ratings are close to system's ratings with high probability. We employ hierarchical agglomerative clustering algorithm to incorporate these rating patterns to detect imitating attackers. (3) Real datasets based experiments are conducted to comprehensively evaluate the performance of different components of the proposed algorithm. Traditional machine learning and collaborative filtering based approaches are involved in performance comparison.

2. RELATED WORKS

To the best of our knowledge, there is no work dedicated to defending attacks in community based web credibility evaluation systems. In this section, we divide the related works into two parts: web credibility evaluation systems and attack defence mechanisms in other collaborative rating systems.

Web credibility evaluation systems. Web credibility is a complex concept. A number of works provided different definitions and identified various factors that may influence an individual's perception of the credibility of web contents [6, 5]. Schwarz at al. [14] showed that visualizations by considering features such as overall web page popularity, domain type, the location origin of the page hits, any awards or certifications of the page, the PageRank metric, etc. can improve a user's web credibility assessment in web search results. Similarly, Yamamoto et al. [16] proposed a system to visualize score of web search results from five aspects, i.e., accuracy, authority, objectivity, coverage and currency. By predicting a user's credibility judgement based on his/her past credibility feedback, the system re-ranks the search results to provide a credibility-oriented web search.

Recently, some community based web credibility evaluation systems are proposed. Sharifi et al. [15] proposed SmartNotes, a crowdsourcing system (implemented as a browser extension) to detect web security threats such as Internet scams, misleading web information, etc. The main idea is that the users are encouraged to identify and report security threats. Machine learning and natural language processing are then applied to analyze and integrate user feedback. In [13], web credibility is assessed by a decentralized social recommender system. A single credibility metric is derived by combining three components: (1) item-based collaborative filtering, which is based on the features identified from the contents of pages, (2) user-based collaborative filtering, which is based on users' social relationships and (3) web search page ranking.

However, most of these collaborative rating systems focus on efficiently aggregating users' ratings to generate meaningful credibility metric but pay little attention to system's robustness[3].

Attack defence mechanisms in collaborative rating systems. Robustness has been studied in other collaborative rating systems. In recommender systems, inappropriate products can be recommended to users due to unfair ratings provided by malicious profiles that are injected into the systems. These attacks are referred to as "shilling attacks" or "profile injection attacks" [3]. For instance, a type of shilling attack called average attack is introduced in [9], where the basic idea is to help the injected profiles be more similar to normal users by providing ratings with mean equal (or similar) to the average rating of the item being rated. Several works have been proposed to handle shilling attacks. In [2] the authors proposed a classification approach to the problem of detecting profile injection attacks. Two types of features are employed to train a classifier: (1) generic features, which are descriptive statistics for each profile, e.g., rating deviation from mean agreement, and (2) model derived features, which aim to recognize the distinctive characteristics of a particular attack model. Zhang et al. [18] tried to detect attack events by investigating rating changes in averages and entropy in different time-series windows. Selection of theoretically optimal window size is studied to improve the detection accuracy.

Shilling attacks (and the defence mechanisms) are dedicated to collaborative filtering based recommender systems. In this paper, we identify and investigate a new type of attack, i.e., imitating attack, which are more harmful to web credibility evaluation systems, but has not been thoroughly studied, thus cannot be efficiently defended by existing defence mechanisms.

3. IMITATING BEHAVIOR RECOGNITION

3.1 Characteristics of web pages

In a credibility evaluation system, the ultimate goal of an attacker is to promote or demote the credibility of certain web pages. According to past research on web credibility [4, 12], credible pages often demonstrate some characteristics which might distinguish them from non-credible ones. Based on this finding, we argue by comparing characteristics (i.e., features) of the target page with that of known credible and non-credible pages, we are able to (to certain extent) infer the credibility of the target page. By adopting features from previous works [8, 13, 12], as well as exploring other sources, we divide all possible features into two categories: textual content related features and link structure related features.

Textual content features.

This class of features are extracted from the textual content of a web page. Specifically, these features are categorized into (1) syntactic and lexical features, (2) semantic features and (3) Natural Language Processing (NLP) features.

The *syntactic features* we identify from the textual content of the web pages include Part-of-speech, e.g., the number of nouns, verbs, etc. counted from the main content of the page, and punctuation marks, e.g., the number of questions marks, exclamation marks, etc. The *lexical features* are identified by investigating the text complexity and the number of spelling errors. For a web page p,

[3] MyWOT does have defense mechanisms (e.g., reputation system, trusted third parties, etc.) but they do not reveal much details, particularly on user's ratings aggregation.

with λ words and the frequency of each word (p_i), text complexity [8] of this page is computed by document entropy: $entropy(p) = \frac{1}{\lambda}\sum_{i=1}^{\lambda} p_i[log_{10}(\lambda) - log_{10}(p_i)]$

We identify two *semantic features*: (1) Category of a web page. We divide all web pages into different categories. For instance, one categorization could be: Arts & Entertainment, Business, Computers & Internet, Culture & Politics, Gaming, Health, Law & Crime, Religion, Recreation, Science & Technology, Sports and Weather. (2) Informativeness of a web page [8]. Informativeness captures the importance of the content of a page relative to other pages in the same page corpus (e.g., a set of web pages returned by a search engine provider). We measure the informativeness of a page p using traditional TF∗IDF approach:

$$informativeness(p) = \sum_{t_j \in p} tf_{t_j,p} \times idf_{t_j,P'}, \qquad (1)$$

$tf_{t_j,p} = \frac{n_j}{\sum_k n_k}$ where n_j is the number of occurrences of term t_j and $\sum_k n_k$ is the total number of occurrences of all terms in page p. $idf_{t_j,P'} = log\frac{|P'|}{|p':t_j \in p'|+1}$ values terms that occur infrequently across all pages (denoted by P').

Recent research has demonstrated that sentiment bias may evidently influence the credibility of web contents [4, 17]. By conducting sentiment analysis, we extract a set of *sentiment related features* such as positive and negative opinions of a piece of web content, the number of subjective and objective sentences, etc. Several tools are available for conducting natural language processing, e.g., LingPipe (http://alias-i.com/lingpipe/) and Alchemy API (http://www. alchemyapi.com/).

Link structure features.

Web page link analysis has been widely used in the area of web search [11]. Although link structure of a web page does not accurately reflect the credibility of this page, however, by investigating links between different web pages, we believe a page's credibility can be inferred (to certain extent). For instance, being pointed by many other pages indicates that the page is very popular in certain contexts, and such popularity is often positively correlated with the page's credibility. So we believe link structure related features could be a promising indicator for web credibility assessment [14].

Based on the discussion above, we identify a set of features: (1) Google's PageRank, (2) TrustRank [7], which is designed to combat web spam by assigning a trust score to each page, (3) the number of internal inbound links (i.e., the links coming from the same root domain), (4) the number of external inbound links (i.e., the links coming from external sites), (5) the ratio of internal inbound links to outbound links and (6) the ratio of external inbound links and outbound links, (7) traffic rank. These metrics can be easily obtained by some third party services (e.g., http://www.seomastering.com).

3.2 Rating patterns

Although from the viewpoint of the system, an imitating attacker acts like trustworthy users (before launching attacks), we still believe an attacker's imitating behavior can be inferred by investigating his/her profile (i.e., past rating behavior). In this section, we try to identify a variety of rating patterns that may distinguish imitating attackers from genuine users. MovieLens-1M dataset[4] is used to demonstrate rating patterns.

3.2.1 Rating category

Since most systems organize their web contents into different categories (e.g., eBay.com, Epinions.com, etc.), the impact of a user's rating on the credibility of the target page largely depends on his/her reputation in the corresponding category. So in order to build such category-aware reputation, an attacker will primarily copy and submit ratings in one or several categories that are the same with or closely related to the target page's category. On the other hand, genuine users have no such constraints and they often have broad interests in diverse categories. This is demonstrated using MovieLens data, where each movie is associated with one or more categories. For instance, the movie "Toy Story" belongs to the categories of "Animation", "Children's" and "Comedy".

We show in Figure 1(a) the cumulative distribution function (CDF) of the number of categories of users (there are 18 categories in total). We observe that around 97% of the users have more than 11 categories, and 70% of the users have even more than 14 categories. However, in order to avoid being detected, an intelligent attacker may randomly rate a small fraction of pages from many other categories. To address this issue, we use the entropy of the categories of users (see Figure 1(b)). Category entropy captures the degree of dispersal or concentration of the distribution of a user's interested categories. For a user u_i, we assume he/she has $C_i = \{c_1, c_2, ..., c_m\}$ categories, the entropy of categories of user u_i is calculated as:

$$Entropy(i) = -\sum_{j=1}^{m}(n_j/N)log_2(n_j/N), \qquad (2)$$

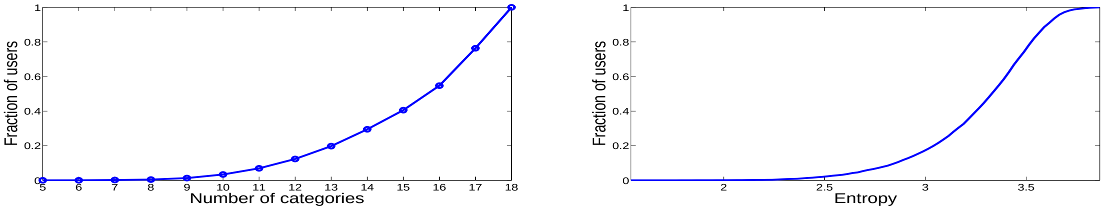
where n_j is the number of u_i's ratings in category c_j, and $N = \sum_{j=1}^{m} n_j$ is the total number of ratings across all categories. The value of entropy falls into the range $[0, log_2 m]$. The value 0 is taken when all ratings belong to one category and the value $log_2 m$ is taken when the ratings are evenly distributed to all categories. Figure 1(b) shows that most users relatively evenly spread their attentions to multiple categories, which is different from the imitating attacker who only concentrates on individual categories. To sum up, we believe by investigating categories that a user has rated in, we may infer the difference between attackers and genuine users.

3.2.2 Rating frequency

We assume attackers' malicious behavior is subject to their resource budgets (e.g., time). In order to attack the target web page, an imitating attacker typically tries to build reputation as soon as possible. So one rating pattern that an attacker may demonstrate is that he/she submits ratings within a short period of time. In order to measure a user u_i's rating frequency, we divide his/her life cycle into equal-sized M time windows (e.g., window size = 12 hours). We denote the value of u_i's jth window $w_{i,j}$ by $v_{i,j}$, which is the number of ratings in this time window. So if an attacker frequently rates pages to quickly build reputation, his/her ratings will densely fall in a few time windows, while genuine users rate pages whenever they want, so their ratings are relatively sparsely distributed to many time windows.

We again calculate the entropy of time windows for each user (refer to Equation 2), and show the cumulative distribution function of such entropy (see Figure 2). We set the time window size to 12 hours for all users. Since users' system ages vary a lot (i.e., different users have different numbers of time windows), in order to accurately measure real entropy of time windows, we normalize each user's entropy by dividing the corresponding maximum entropy: $Entropy(i)/log_2 W_i$, where W_i is the number of time windows of user u_i. Note that we only consider windows whose

[4] Downloaded from http://www.grouplens.org/node/12



(a) CDF of number of categories. (b) CDF of entropy of categories.

Figure 1: Rating category patterns.

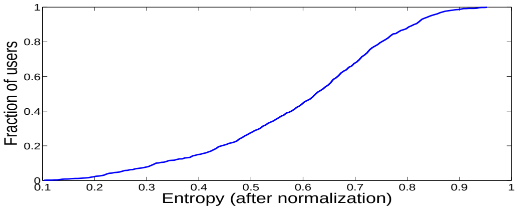


Figure 2: CDF of entropy of rating frequency.

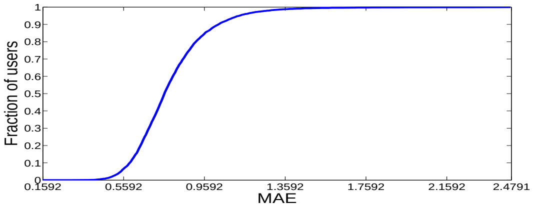


Figure 3: CDF of MAE.

values are larger than 0, i.e., at least one rating falls into this window. We observe that around 80% of the users have normalized entropy less than 0.75, which means most users randomly spread their ratings across multiple time windows. This distinguishes the behavior of attackers whose ratings are aggregated in a few time windows, thus producing high normalized entropies.

3.2.3 Rating reliability

Recall that the unique behavior of an imitating attacker is to copy and resubmit system's ratings. So the majority of ratings submitted by an imitating attacker are the same (or quite similar to) the corresponding system's ratings. On the other hand, since a genuine user may have personalized view on the credibility of a particular page, it is difficult to ensure that his/her ratings are always consistent with the system's ratings. We further argue that even the trustworthy experts cannot perfectly guarantee the reliability of their ratings. For instance, their ratings are reliable in certain categories where they are experts, but may be less reliable in other categories where they are less specialistic.

We measure the reliability of a user u_i's ratings using *mean absolute error* (MAE) by comparing the ground truth (system's credibility ratings[5]) with u_i's ratings:

$$MAE(i) = \frac{\sum_{j=1}^{|R_i|} |l^{i,j} - \bar{l^j}|}{|R_i|}, \qquad (3)$$

where R_i is a set of u_i's submitted ratings, and $l^{i,j}$ is value of u_i's jth rating. $\bar{l^j}$ is system's rating when u_i's jth rating is submitted. We show in Figure 3 the cumulative distribution function of MAE of users. Note that in MovieLens dataset, the rating for each movie ranges from 1 to 5. From the figure we observe that MAE of about 98% of users is larger than 0.5. We argue that since imitating attackers simply copy or slightly modify the system's ratings (otherwise, attackers' reputation may be damaged, thus defeating the purpose of attacking), the MAE of attackers should be kept as small as possible. We therefore can utilize this metric to infer an imitating attacker's behavior.

4. DEFENCE ALGORITHM

Based on the imitating behavior patterns recognized in Section 3.1 and 3.2, we propose a two-stage algorithm to detect imitating attack, which will be elaborated in the next subsections.

4.1 Stage 1: Machine learning based credibility predication

In Section 3.1, we identify a set of features F that are expected to have the potential to distinguish different levels ($\in L$) of web credibility. Since different features may possess different distinguishing power, we utilize entropy based information gain to select the most discriminating features. We assume a set of web pages P' with known credibility are available as the training data. We denote the fraction of pages with rating $l_i \in L$ by fr_i. The entropy of all pages P' is calculated as $Entropy(P') = -\sum_{i=1}^{|L|} fr_i log_2 fr_i$.

Entropy is used to characterize the (im)purity of a collection of samples. For each feature $f \in F$, we assume it has a set of values (for discrete variable) or intervals (for continuous variable), which are denoted by $\Upsilon(f)$. For each $v \in \Upsilon(f)$, we denote the set of web pages that are associated with v by P'_v. The information gain of feature f is therefore calculated as:

$$IGain(P', f) = Entropy(P') - \sum_{v \in \Upsilon(f)} \frac{|P'_v|}{|P'|} Entropy(P'_v). \qquad (4)$$

Information gain of a feature measures the expected reduction in entropy by considering this feature. Clearly, the higher the information gain, the lower the corresponding entropy becomes and thus the better the classification of web pages is achieved (by using this feature). Then we may choose top-K features that have the highest information gain. We will show in Section 6.2.1 which features are finally selected.

After selecting useful features, we apply machine learning algorithms to classify the target web page to the group with certain credibility rating. We choose linear discriminant analysis (LDA) to combine the selected features for classification. We first divide P' into $|L|$ groups according to the pages' credibility ratings ($\in L$), and each page is represented by its feature vector. The objective of LDA is to find a transformation Φ that can maximize the inter group variance S_b and minimize the intra group variance S_w. Formally, the criterion function to be maximized is defined:

$$J(\Phi) = \frac{\Phi^T S_b \Phi}{\Phi^T S_w \Phi} \qquad (5)$$

[5] It is not always true to use system's credibility information as ground truth. However, in practice, it is non-trivial to obtain real credibility, so we use system's rating which is derived by aggregating users' ratings to approximate ground truth.

The projection direction Φ is found as the eigenvector associated with the largest eigenvalue of $S_w^{-1}S_b$. We then transform all groups of web pages and the target page (represented by its feature vector $\{f_{p,1}, f_{p,2}, ...\}$) using Φ. Classification is done by measuring the distances between the target and the $|L|$ groups: $Distance(j) = \Phi^T p - \Phi^T c_j$, where j represents the jth group, and c_j is centroid of that group.

By applying LDA, the system is able to make the initial estimate of the credibility of page p, denoted by l_j. We denote the credibility rating submitted by user u by l_i. The purpose of the first stage of our approach is to estimate to what extent the user's rating is consistent with the predicted credibility. Specifically, we define $\Delta = |l_i - l_j|$. If Δ is smaller than a predefined threshold, the system considers the user's rating is consistent with the predicted rating, otherwise, the system predicts the user to be malicious.

If machine learning algorithms can accurately predict the web credibility by using the selected features, the system is able to detect any attackers (thus stage 2, which will be described in the next section is not necessary anymore). However, it is extremely difficult, if not impossible, to identify all powerful features in such a complex system. So in order to handle the uncertainty introduced by machine learning, we assign a confidence score ρ to the predicted rating. We denote distances between the target page p and the $|L|$ groups by $D_p = \{d_1, d_2, ..., d_{|L|}\}$. If we assume the shortest distance is $d_k \in D_p$, the confidence score is defined as:

$$\rho = \frac{\sum_{m \in \{1,2,...,|L|\} \bigcap m \neq k} \frac{|d_k - d_m|}{d_k + d_m}}{|L| - 1}. \tag{6}$$

We can derive from this equation that if $d_k = 0$, then $\rho = 1$, i.e., the target page's credibility is predicted to be l_k confidently; if all distances are identical, $\rho = 0$, the target page is classified to every group with the same probability, i.e., random classification. The predicted rating is valid only if its confidence score $\rho(\in [0, 1])$ is larger than a threshold. We will demonstrate in Section 6 how this threshold influences the performance of our algorithm.

4.2 Stage 2: Rating pattern based imitating attacker detection

If web credibility cannot be confidently predicted at stage (1), our algorithm goes to stage (2) to detect imitating attackers using the identified rating patterns (see Section 3.2): (1) rating category, (2) rating frequency, and (3) rating reliability.

We first define that a user is considered to be genuine if (1) he/she has been active in the system for a long time (e.g., no intense rating submission within a short period of time), (2) he/she rated web pages for multiple categories but without particular focus on individual ones, (3) his/her ratings are generally consistent with but not too close to the system's ratings.

Given a set of 'genuine' users (based on the definition) and the suspicious user u, by leveraging the three identified rating pattern metrics, our algorithm applies unsupervised learning algorithm to cluster these users into two groups, where one for real genuine users and the other for imitating attackers. The general assumption is that although the population of imitating attackers might be considerable in the system, the amount of intelligent and well-equipped (in terms of various resources) attackers is small. So we believe if the suspicious user u is clustered into the group with smaller size, he/she is considered to be imitating attacker, otherwise, he/she is considered to be genuine.

We apply agglomerative hierarchical clustering algorithm (AHCA) [10] to cluster the users into genuine user group and attacker group (see Algorithm 1). We denote the set of selected 'genuine' users plus the suspicious user u by U'. Each user is described by the three rating pattern metrics, denoted by M_c, M_f and M_r respectively. To start with, each user ($\in U'$) forms a cluster, so we have $|U'|$ singleton clusters, denoted by $\mathcal{C}$ (line 1). We calculate (Euclidean) distances between all pair of clusters (line 3). The pair of clusters with the minimum distance is selected (line 7): $D(C', C'') = \arg\min_{1 \leq x,y \leq |U'|, x \neq y}[D(C_x, C_y)]$.

In case multiple pairs of clusters have the same minimum distance, a pair of them is randomly selected (line 5). Then the selected clusters are merged to form a new cluster (line 9). We update the distances between the new cluster and rest of the clusters by employing Lance and Williams method. Specifically, distance between the new cluster and a cluster $\hat{C}$ is formulated as:

$$\begin{split} D((C' \bigcup C''), \hat{C}) = \alpha' D(C', \hat{C}) + \alpha'' D(C'', \hat{C}) + \\ \beta D(C', C'') + \gamma |D(C', \hat{C}) - D(C'', \hat{C})|. \end{split} \tag{7}$$

When single linkage method is applied, the parameters are set to: α' = α'' = 1/2, β = 0, and γ = -1/2. So Equation 7 can be rewritten as:

$$D((C' \bigcup C''), \hat{C}) = \arg\max[D(C', \hat{C}), D(C'', \hat{C})]. \tag{8}$$

This cluster merging process continues until only two clusters are left (line 2 – 11). The two clusters are expected to distinguish imitating attackers from genuine users. That is, based on the discussion above, the smaller cluster is considered to be attacker cluster (line 12 – 16). Then whether the suspicious user u is an attacker or not is determined by which cluster u belongs to (line 17).

Algorithm 1 Users clustering algorithm

```
 1: We generate $|U'|$ clusters, denoted by $\mathcal{C}$. Each $C_k \in \mathcal{C}$ contains
    only one user $u_k \in U'$.
 2: while $|\mathcal{C}| > 2$ do
 3:    Calculating distances between all pairs of clusters in $\mathcal{C}$.
 4:    if Multiple pairs of clusters with the same minimum distance
       exist then
 5:       Randomly selecting two clusters ($C'$,$C''$) from these clusters
 6:    else
 7:       Searching the pair of clusters ($C'$,$C''$) with minimum distance.
 8:    end if
 9:    Merging the pair of clusters: $C' \bigcup C''$ ($|\mathcal{C}| = |\mathcal{C}| - 1$).
10: end while
11: Two final clusters are obtained: $C_a$, $C_b$.
12: if $|C_a| < |C_b|$ then
13:    $C_a$ is labeled as attacker cluster, $C_b$ is labeled as genuine
       user cluster.
14: else
15:    $C_a$ is labeled as genuine user cluster, $C_b$ is labeled as attacker cluster.
16: end if
17: The suspicious user $u$ is predicted as an imitating attacker if
    he/she is in attacker cluster, otherwise, he/she is predicted as
    genuine user.
```

5. ROBUSTNESS OF THE PROPOSED ALGORITHM

It is worth mentioning that for the first stage of our algorithm, some intelligent attackers (the web content authors) may try to avoid being detected by carefully composing their web contents (e.g., clear and elegant writing style, comfortable page design, manipulated TF$*$IDF, etc.), however we argue that by incorporating features from diverse sources, our approach is robust against such attackers from two aspects: (1) an attacker must comprehensively investigate features that are relevant to web credibility, and then spend much efforts in manipulating these features. The tradeoff between such efforts and the gains by launching an attack will greatly restrict an attacker's behavior. (2) some features, like PageRank, etc., which are measured by third party services are extremely difficult, if not impossible, to be manipulated.

For the second stage of our algorithm, in order to launch effect attacks, most attackers won't spend a long time to build high reputation. So rating frequency is a promising indicator to fight against attackers. Furthermore, attackers are unaware of other users' rating patterns and how the system selects genuine users[6] for clustering task so it is difficult for attackers to imitate rating patterns of specific genuine users to cheat our algorithm.

To sum up, although at each stage, it may not be difficult for attackers to manipulate individual features or rating patterns, by applying machine learning algorithms and incorporating various features and rating patterns, our algorithm is quite robust against imitating attackers. This will be demonstrated by real data based experiments in the next section.

6. EVALUATION

6.1 Evaluation methodology

In order to evaluate the performance of the first stage of our approach, we use Microsoft web credibility corpus [14] which records 1000 web pages from 5 five topics: Health, Politics, Finance, Environmental Science, and Celebrity News. The credibility rating of each page ranges from 1 to 5 where 1 represents "completely non-credible" and 5 represents "completely credible". In order to evaluate the performance of the second stage of our approach, we still use MovieLens data, which consists of about 1 million ratings of approximately 3900 movies made by 6040 users. Ratings are also made on a 5-star scale, and each user has at least 20 ratings.

A set of imitating attackers were introduced into the system, and the fraction of attackers is denoted by F. We further divide attackers into three groups based on their intelligence and resource budget: (i) Naive attackers. They only copy system's ratings in 10 categories (we have totally 18 categories) without any modification. These ratings are normally distributed (μ = 20 , σ = 2) to x time windows (each window is 24 hours), where x is determined by quartering the average number of time windows of all genuine users. (ii) Medium attackers. They copy but slightly modify system's ratings (using the deviation of $\pm$0.5) for at most 15 categories. Their ratings are normally distributed (μ = 20 , σ = 2) to y time windows, where y is determined by halving the average number of time windows of all genuine users. (iii) Smart attackers. They copy but further modify system's ratings (using the deviation of $\pm$1) for all categories, and further spread their ratings evenly, i.e., the ratings are normally distributed (μ = 20 , σ = 2) to the time windows by averaging that of all genuine users. We denote the fractions of the naive, medium and smart attackers in all attackers by F_n, F_m and F_s respectively (F_n + F_m + F_s = 1).

Table 1: Classification accuracy

	LDA	decision tree	SVM	# of effective classifications
Accuracy (no confidence)	***0.527***	0.475	0.510	500
Accuracy (50% confidence)	***0.615***	0.566	0.584	410
Accuracy (60% confidence)	***0.657***	0.628	0.640	348
Accuracy (70% confidence)	***0.702***	0.648	0.672	272
Accuracy (80% confidence)	***0.781***	0.720	0.732	185

Table 2: Effectiveness of individual rating patterns

Rating pattern	precision	recall	F-measure
Rating category	0.740	0.751	0.745
Rating frequency	0.758	0.775	0.766
Rating reliability	0.695	0.705	0.700
All rating patterns	***0.841***	***0.865***	***0.850***

In order to evaluate the performance of the entire approach (stage 1 + stage 2), ideally, we need a dataset containing textual information of web pages as well as their ratings submitted by multiple users. Unfortunately, such data is not publicly available. Therefore, we try to set up a realistic simulation environment by combining relevant real datasets (i.e., Microsoft web credibility corpus and MovieLens dataset). Specifically, we randomly select 1000 MovieLens users who are considered to be genuine users. The attacker configuration is the same with that for evaluating stage 2 of our approach as mentioned in previous paragraph. We randomly select 500 web pages from Microsoft credibility corpus, and map these pages onto 500 selected movies (from MovieLens) that are rated by the 1000 users. Each attacker copies system's ratings of a subset of the 500 selected web pages (following Normal distribution [μ = 20 , σ = 2]) and then tries to attack the rest 500 pages, i.e., providing ratings that mostly deviate from the system's ratings.

We will compare the performance of our algorithm with machine learning based approach, and item based collaborative filtering algorithm. The metrics we use to evaluate the performance of different approaches include precision (i.e., the fraction of predictions that correctly detect imitating attackers), recall (the fraction of all imitating attackers that are correctly detected) and F-measure, which is measured by combining precision and recall: F-measure = $\frac{2PR}{P+R}$, where P represents precision and R represents recall.

6.2 Results

6.2.1 Stage 1: Machine learning based initial credibility prediction

In order to accurately classify the target web page, we identify 20 features from textual contents and link structure of web pages (Section 3.1). We then select top 10 features which have the highest information gain: (1) PageRank, (2) text complexity, (3) TrustRank, (4) informativeness, (5) the ratio of external inbound links and outbound links, (6) traffic rank, (7) the number of negative sentences, (8) the number of question marks, (9) the number of exclamation marks, and (10) the number of spelling errors.

Table 1 shows the classification accuracy of LDA in comparison with decision tree (C4.5) and Support Vector Machine (SVM). Weka (http://www.cs.waikato. ac.nz/ml/weka/) is used to run various machine learning algorithms. Recall that our approach associates a confidence score ρ ($\in [0, 1]$) to a classification (see Section 4.1). We can see when no confidence is considered, the highest classification accuracy is only 0.527. Row 2-5 in Table 1 shows the classification accuracy when the threshold is 0.5, 0.6, 0.7 and 0.8 respectively[7]. Clearly, the higher the confidence threshold, the higher the classification accuracy becomes. However, higher confi-

[6] The system may select genuine users with heterogeneous rating patterns.

[7] Note that for decision tree, confidence score is measured by aggregating entropy reduction at each node; for SVM, confidence score is measured by comparing distances between the target and

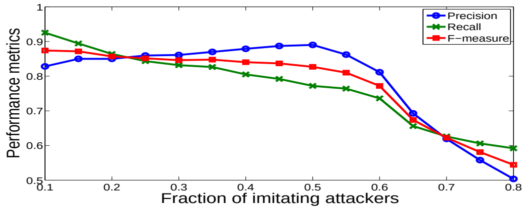


Figure 4: Performance with varying fraction of attackers.

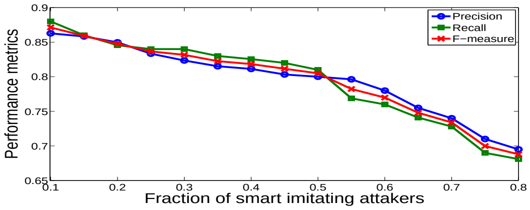


Figure 5: Performance with varying fraction of smart imitating attackers (in all imitating attackers).

dence threshold means more classifications (with low confidence) are ignored, thus resulting in less usable classifications. For instance, when confidence threshold is as high as 0.8, only 185 out of 500 classifications are usable. So the tradeoff between the classification accuracy and the number of usable classifications must be investigated such that the first stage of our approach is both effective and efficient. We will demonstrate in Section 6.2.3 how this tradeoff influences the overall performance of our approach. We also observe that LDA generates higher accuracy than decision tree and SVM in all cases, proving that LDA is more suitable for our approach (at least for Microsoft web credibility corpus).

6.2.2 Stage 2: Rating pattern based imitating attacker detection

We first evaluate the performance of our approach when individual rating patterns are used. Table 2 summarizes the results under the condition that $F = 0.25$ ($F_n = 0.5$, $F_m = 0.3$, $F_s = 0.2$) and time window size is 24 hours. We observe that rating frequency is the best pattern while rating reliability generates the lowest precision, recall and F-measure. We also observe when all three rating patterns are applied, our approach generates better results. This again proves the effectiveness of the identified rating patterns.

We then demonstrate the performance of our approach (considering all rating patterns) when the fraction (F) of imitating attackers varies from 0.1 to 0.8 with 0.05 as increment (we set $F_n = 0.5$, $F_m = 0.3$, $F_s = 0.2$). From Figure 4 we observe that when F increases, precision, recall and F-measure decrease in general. This is because when more attackers are introduced into the system, it becomes more difficult to select real genuine users for clustering the target user, thus influencing the accuracy of imitating attacker detection. An interesting phenomenon of precision trend is that before F grows to 0.5, instead of decreasing as recall and F-measure, precision increases slightly. This is because although the probability that an attacker is correctly clustered becomes lower when F increases, the probability that a genuine user is falsely clustered also decreases (due to clearer difference between clusters introduced by more attackers). The strengths of these two effects vary with F, making the precision first slightly increase and then decrease. Note that recall is not influenced by the accuracy of genuine user clustering, so it has no such trait.

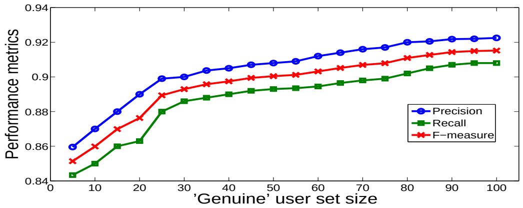


Figure 6: Performance with varying 'genuine' user set size.

Table 3: Performance comparison: varying confidence score

	precision	recall	F-measure
Our algorithm (40% confidence)	0.662	0.616	0.638
Our algorithm (50% confidence)	0.673	0.626	0.649
Our algorithm (60% confidence)	0.691	0.659	0.679
Our algorithm (70% confidence)	0.720	0.687	0.705
Our algorithm (80% confidence)	0.755	0.713	0.735
Our algorithm (90% confidence)	0.793	0.744	0.768
Machine learning based	0.598	0.589	0.596
Collaborative filtering based	0.616	0.658	0.637

Compared to other types of imitating attackers, smart attackers behave more similarly to reputable users, thus are more difficult to be detected. We then demonstrate how the fraction of smart imitating attackers influences the performance of our approach (see Figure 5). We first set F = 0.25. Among all attackers, we vary the fraction F_s of smart attackers from 0.1 to 0.8 with 0.05 as increment. Naive attackers and medium attackers equally share the rest portion (i.e., $1 - F_s$). As expected, the higher the fraction of smart imitating attackers, the lower the precision, recall and F-measure become. Nevertheless. we observe that even when F_s is as high as 0.7, F-measure still reaches a high value of over 0.6, demonstrating the robustness of stage 2 of our algorithm.

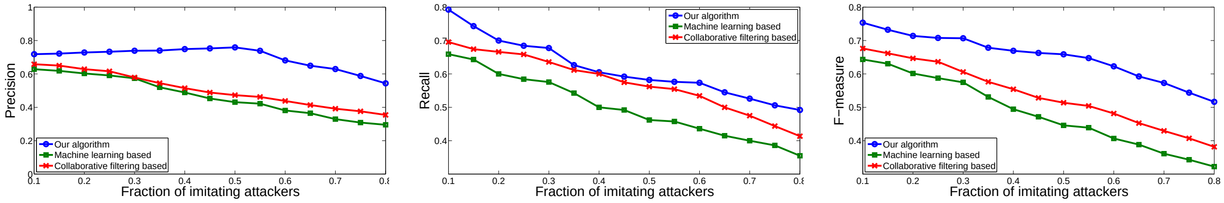
An important task of our approach is to select a set of 'genuine' users (refer to Section 4.2), which are used for clustering and detecting the suspicious attackers. We next investigate how the size of such 'genuine' user set influences the performance of our algorithm. From Figure 6 ($F = 0.25$; $F_n = 0.5$, $F_m = 0.3$, $F_s = 0.2$) we observe the general trend is that the larger the 'genuine' user set, the higher the precision, recall and F-measure. This is because larger user set size (i.e., more training data) means more genuine users with different characteristics of rating patterns are included, so learning difference between the imitating attacker and majority of genuine users becomes more reliable. We also observe that although precision, recall and F-measure increase with larger user set size, from the size of around 25, the performance become stable. So we believe that by selecting suitable set size, our algorithm is able to achieve high performance while keeping computational complexity reasonable.

6.2.3 Comparison study

Finally, we compare the performance of our algorithm (stage 1 + stage 2) with two traditional approaches: (1) machine learning (LDA) based approach, which predicts web credibility based on the features selected in Section 6.2.1, (2) item-based collaborative filtering algorithm, which predicts the credibility of the target web page by aggregating ratings on similar pages. Note that for machine learning and collaborative rating based approaches, if the difference between the predicted rating and the user's rating is larger than or equal to 2, the user is considered to be attackers.

We first compare the performance of the three approaches with varying fraction F of imitating attackers (F_n = 0.5, F_m = 0.3, F_s = 0.2, and confidence threshold of 0.75). We observe from Figure 7 that item-based collaborative filtering outperforms machine learning. This shows that current features have limited distinguishing

the groups. Also note that due to different confidence computation methods, the numbers of effective classifications for different algorithms are *slightly* different, and in the table we only show the numbers for LDA.



(a) Precision.

(b) Recall.

(c) F-measure.

Figure 7: Performance comparison with varying F.

power, thus cannot produce highly accurate classification (for both machine learning approach and the first stage of our algorithm). Nevertheless, by (1) setting confidence threshold and (2) investigating rating providers' past rating patterns, our algorithm has higher precision, recall and F-measure, thus is more robust against imitating attack compared to machine learning and collaborative filtering.

We then compare the performance with varying confidence score (for our algorithm). F is set as 0.25. Table 3 summarizes various performance metrics for the three approaches. Note that Table 1 and Table 3 demonstrate different results, where Table 1 summarizes the classification accuracy, and Table 3 shows the performance of attack detection. It is clear that bigger confidence score evidently improves the performance of our algorithm. This again proves that simply applying machine learning cannot achieve high performance, and stage 2 is a compulsory component for effective and efficient web credibility assessment.

7. CONCLUSION

In this paper, we identify a new type of attack, i.e., imitating attack in community based web credibility evaluation systems. In order to defend this attacker, we propose a two-stage defence mechanism. At stage 1, we apply supervised learning algorithm to estimate the credibility of the target web content to make the first round of detection of attackers. If such a detection is not confident enough, the algorithm goes to stage 2 where we investigate a variety of rating patterns of users to detect attackers by applying hierarchical clustering algorithm. Real datasets based experiments show that even when the population of imitating attackers is very high (e.g., 60%), our approach still achieves higher precision, recall and F-measure than traditional machine learning and collaborative filtering based approaches. Future study will include a more comprehensive investigation of web content features and rating patterns. We are also interested in extending current defence algorithm to defend more types of attacks to ensure a robust community based web credibility evaluation system.

8. ACKNOWLEDGEMENT

This work was partially supported by the grant Reconcile: Robust Online Credibility Evaluation of Web Content from Switzerland through the Swiss Contribution to the enlarged European Union.

9. REFERENCES

[1] How reliable are the ratings? (mywot). http://www.mywot.com/en/faq/website/rating-websites#reliableratings.

[2] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik. Detecting profile injection attacks in collaborative recommender systems. In *Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, 2006.

[3] R. Burke, B. Mobasher, R. Zabicki, and R. Bhaumik. Identifying attack models for secure recommendation. In *Beyond Personalization: A Workshop on the Next Generation of Recommender Systems*, 2005.

[4] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th WWW*, 2011.

[5] B. J. Fogg, J. Marshall, O. Laraki, A. Osipovich, C. Varma, N. Fang, J. Paul, A. Rangnekar, J. Shon, P. Swani, and M. Treinen. What makes web sites credible?: a report on a large quantitative study. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2001.

[6] B. J. Fogg and H. Tseng. The elements of computer credibility. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1999.

[7] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth VLDB - Volume 30*, 2004.

[8] C.-F. Hsu, E. Khabiri, and J. Caverlee. Ranking comments on the social web. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, 2009.

[9] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, 2004.

[10] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *Computer Journal*, 26(4):354–359, 1983.

[11] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *Proceedings of the 24th SIGIR*, 2001.

[12] A. Olteanu, S. Peshterliev, X. Liu, and K. Aberer. Web credibility: Features exploration and credibility prediction. In *Proceedings of the 35th ECIR*, 2013.

[13] A. Papaioannou, J.-E. Ranvier, A. O., and K. Aberer. A decentralized recommender system for effective web credibility assessment. In *Proceedings of the CIKM*, 2012.

[14] J. Schwarz and M. Morris. Augmenting web pages and search results to support credibility assessment. In *Proceedings of the SIGCHI*, 2011.

[15] M. Sharifi, E. Fink, and J. G. Carbonell. Smartnotes: Application of crowdsourcing to the detection of web threats. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2011.

[16] Y. Yamamoto and K. Tanaka. Enhancing credibility judgment of web search results. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, 2011.

[17] J. Zhang, Y. Kawai, S. Nakajima, Y. Matsumoto, and K. Tanaka. Sentiment bias detection in support of news credibility judgment. In *Proceedings of the 44th Hawaii International Conference on System Sciences*, 2011.

[18] S. Zhang, A. Chakrabarti, J. Ford, and F. Makedon. Attack detection in time series for recommender systems. In *Proceedings of the 12th ACM SIGKDD*, 2006.