Trustworthy Website Detection Based on Social Hyperlink Network Analysis

Xiaofei Niu, Guangchi Liu, *Student Member,* and Qing Yang, *Senior Member*

Abstract—Trustworthy website detection plays an important role in providing users with meaningful web pages, from a search engine. Current solutions to this problem, however, mainly focus on detecting spam websites, instead of promoting more trustworthy ones. In this paper, we propose the enhanced OpinionWalk (EOW) algorithm to compute the trustworthiness of all websites and identify trustworthy websites with higher trust values. The proposed EOW algorithm treats the hyperlink structure of websites as a social network and applies social trust analysis to calculate the trustworthiness of individual websites. To mingle social trust analysis and trustworthy website detection, we model the trustworthiness of a website based on the quantity and quality of websites it points to. We further design a mechanism in EOW to record which websites' trustworthiness need to be updated while the algorithm "walks" through the network. As a result, the execution of EOW is reduced by 27.1%, compared to the OpinionWalk algorithm. Using the public dataset, WEBSPAM-UK2006, we validate the EOW algorithm and analyze the impacts of seed selection, size of seed set, maximum searching depth and starting nodes, on the algorithm. Experimental results indicate that EOW algorithm identifies 5.35% to 16.5% more trustworthy websites, compared to TrustRank.

Index Terms—Trust model, social trust network, trustworthy website detection, social hyperlink network.

1 Introduction

Search engines have become more and more important for our daily lives, due to their ability in providing relevant information or web pages to users. Although a search engine typically returns thousands of web pages to answer a query, users usually read only a few ones on top of the list of recommended pages [1]. The advantage of a company's website being ranked on top of the list can be converted to an increase in sales, revenue and profits. As a result, several techniques are created to clandestinely increase a web page's ranking position, to achieve an undeserved high click through rate (CTR). The deceptive actions produce untrustworthy websites that are generally referred to as spam websites [2]. It is shown that 22.08% of English websites/hosts are classified as spams [3]. Similarly, about 15% of Chinese web pages are spams. With spam websites ranked on the top of searching results, users waste their time in processing useless information, leading to a deteriorated users' quality of experience (QoE). Therefore, it is critical to design a mechanism to promote more trustworthy websites and eliminate spams in the searching results provided to users.

1.1 Limitations of Prior Art

Existing solutions to trustworthy website detection focus mainly on identifying spam websites, i.e., while spam websites are removed from the searching results, more trustworthy websites are promoted. Web spams can be broadly classified into four groups: content spam, link spam, cloaking and redirection, and click spam [4]. Content spam refers to deliberate changes in HTML fields of a web page so that the spam page becomes more relevant to certain queries. For example, keywords relevant to popular query terms can be inserted into a spam page. Link spam allows a web page to be highly ranked by means of manipulating the page's connections to other pages, resulting in a confusion of hyperlink structure analysis algorithms, e.g., PageRank [5] and HITS [6]. Cloaking is a technique that provides different versions of a page to users, based on the information contained in user queries. The redirection technology redirects users to malicious pages through executing JavaScript codes. Click spam is used to generate fraud clicks, with the intention to increase a spam page's ranking position.

Although human can easily recognize spam websites, it's unrealistic to mark all spam websites manually. Therefore, humongous anti-spam techniques are proposed, including solutions based on genetic algorithm [7] and genetic programming [8], [9], [10], [11], [12], [13], artificial immune system [14], swarm intelligence [15], particle swarm optimization [16] and ant colony optimization [17]. It is very difficult, however, to detect all types of web spams, due to the fact that new spam techniques are created almost instantly once a particular type of spam is identified and banned within the Internet. Instead of classifying and detecting spam websites, PageRank [5] and TrustRank [18] make an attempt to explore the possibility of promoting more trustworthy websites to users in the searching results. The solutions first rank all web pages or websites, based on their trust scores, in a descending order. Then, only websites ranked on top of the list are provided to users. As such, the number of spam websites that a user may encounter

- *X. Niu is with the School of Computer Science and Technology, Shandong Jianzhu University, Jinan, Shandong 250101, China. Email: niuxiaofei2002@163.com.*
- *G. Liu is with the Stratifyd Inc., Charlotte, NC 28207, USA. Email:luke.liu@stratifyd.com.*
- *Q. Yang is with the Department of Computer Science and Engineering, University of North Texas, Denton, Texas 76203, USA Email:qing.yang@unt.edu.*
- *Q. Yang is the corresponding author of this article.*

Manuscript received January 31, 2018; revised July 16, 2018.

with will be significantly reduced. Unfortunately, existing trust-ranking based algorithms do not accurately model the trustworthiness of web pages, and thus often mistakenly identify spams as trustworthy websites. To improve the performance of trustworthy website detection, we approach this problem by studying the trust relations among websites, leveraging social network analysis techniques.

1.2 Proposed Solution

Trust has been intensively studied in online social networks [19], [20], and the knowledge obtained from this field can be applied to analyze the hyperlink network, consisted of websites, to understand website trustworthiness. Considering a website as an individual user, and the hyperlinks connecting websites as the social relations among them, we can model the network of websites as a social hyperlink network.

According to the three-valued subjective logic model (3VSL) [21], the trust relation between two websites can be modeled as a trust opinion $< b, d, n >$. Based on the opinion operations defined in 3VSL, the trustworthiness of every website can be computed using the OpinionWalk algorithm [22]. The algorithm starts from a seed node and searches the network, in a breadth first search manner, to compute the trustworthiness of all other nodes, from the seed node's perspective. If multiple seed nodes are chosen, the algorithm will compute several different trust opinions of the same node. These opinions will then be combined to obtain the trustworthiness of the node. As such, the websites with higher trust values can be ranked on top of the list provided to users.

To apply the OpinionWalk algorithm in trustworthy website detection, however, we need to address two challenges. First, the 3VSL models trust as an opinion vector containing three values b (belief), d (distrust), and n (uncertainty). It unfortunately does not specify how the three values of an opinion are obtained. To initialize the trust opinion between two linked websites, we need to understand which factors affect the trustworthiness of a website. From previous studies, we find trustworthy websites rarely point to spam websites and the websites linking to spams are likely to be spams [18] [23]. Therefore, by checking how many trustworthy (or spam) websites a website links to, we can possibly determine the website's trustworthiness, i.e., the values of b, d and n in the corresponding trust opinion. The second challenge lies in the large execution time of the algorithm. As OpinionWalk searches a network level by level, the trustworthiness of all nodes will be updated in each searching/iteration, which yields frequent trust computation and updates that are often not necessary. To address this challenge, we design a mechanism to record which websites' trustworthiness need to be updated and only change them when the algorithm "walks" through the network. As a result, we are able to detect more trustworthy websites within a relatively shorter period of time, compared to the state-of-art solutions [5], [18], [22].

1.3 Contributions

In this paper, we discuss how to identify more trustworthy websites by proposing the enhanced OpinionWalk (EOW) algorithm. The key contributions of this paper are as follows.

For the first time, the hyperlinks between websites are viewed as the "social" connections between websites. Leveraging the trust model designed for social networks, the trustworthiness of websites can be quantified. Due to the accuracy of 3VSL in modelling trustworthiness, individual website's trustworthiness can be precisely calculated, using the proposed EOW algorithm. Based on the previous research results, the trustworthiness of a website are mainly determined by the numbers of trustworthy and spam websites it links to. As only labeled websites' trustworthiness are known, we treat all other websites as uncertain/undecided. Therefore, by counting the numbers of trustworthy, spam, and uncertain websites a website points to, the website's trustworthiness opinion can be formed. We enhance the OpinionWalk algorithm by identifying which opinions need to be updated while the algorithm searching within a social hyperlink network. Specifically, a Boolean vector is used to keep track of the websites that are connected from the current websites whose trustworthiness values are just updated. When the EOW algorithm searches the next level in the network, only these websites' trustworthiness are changed accordingly. The proposed EOW algorithm is validated by experiments using the WEBSPAM-UK2006 dataset that contains both trustworthy and spam websites crawled within the .uk domain. Experimental results indicate that the EOW algorithm identifies 16.5%, 12.65%, 8.77%, 5.35% more trustworthy websites, in the top 1000, 2000, 3000 and 4000 websites, respectively, compared to TrustRank (the start-of-art solution), when the number of trustworthy seeds is 200. In addition, EOW saves (on average) about 27.1% execution time, compared to the OpinoinWalk algorithm, in computing the trustworthiness of all websites.

The rest of this paper is organized as follows. In Section 2, we introduce the proposed EOW algorithm, followed by an example illustrating how it works. In Section 3, we describe how EOW algorithm performs, regarding to detecting trustworthy websites from a real-world dataset. Then, we summarize the related work of trustworthy website detection in Section 4. Finally, we conclude our work and point out future research directions in Section 5.

2 Detecting Trustworthy Websites using Enhanced OpinionWalk Algorithm

Considering websites as users, a hyperlink network can be modeled as a "social network" that reflects the "social connections" among different websites. The connection between two linked websites can be assigned a weight to indicate the trustworthiness between them, which results in a weighted trust social network. Within the network, we can leverage trust propagation and trust combination to compute the trustworthiness of individual websites. As such, trustworthy websites can be identified from all websites available.

OpinionWalk was designed to solve the massive trust assessment problem in social networks [22], however, a few challenges need to be addressed before it can be applied to trustworthy website detection. The first challenge is to design a mechanism to assign weights on connections/links

so that the trustworthiness between websites can be reflected by the weights. The second challenge is to enhance the OpinionWalk algorithm to make it more efficient as the current OpinionWalk algorithm is very slow. We propose the enhanced OpinionWalk (EOW) algorithm that starts from a trustworthy node and searches the hyperlink network to detect more trustworthy websites.

2.1 Social Hyperlink Network Model

The hyperlink graph of websites can be modeled as a directed graph $G = (V, E, W)$, where vertex $i \in V$ denotes a website, edge $e(i, j) \in E$ represents the hyperlink from websites i to j. We call website j as i's adjacent node, and weight $w(i, j) \in W$ of the edge $e(i, j)$ indicates how i trusts j. We further define the indegree of a website as the number of websites pointing to it. The outdegree of a website is defined as the number of websites that it points to. The weight on each edge in graph G is usually modeled as a real number [19], however, we find it cannot accurately reflect the trust between two nodes [21]. To conduct precise trust computation, OpinionWalk algorithm defines the trust as an opinion vector that contains three numbers reflecting how likely a user is trustful, not trustful and uncertain, respectively. We adopt this definition and propose a mechanism to assign the three values of an opinion in the trustworthy website detection problem.

2.2 Edge Weight Assignment

By analyzing the structure of existing hyperlink graphs, we find that a trustworthy site rarely links to a spam site [18]. Besides, a website pointing to many spam sites is very likely to be a spam site [23]. By looking at how many trustworthy websites that a website points to, and how trustworthy these websites are, we are able to determine the trustworthiness of the website. In other words, the quality and quantity of pointed websites should be considered in modeling the trustworthiness of a website.

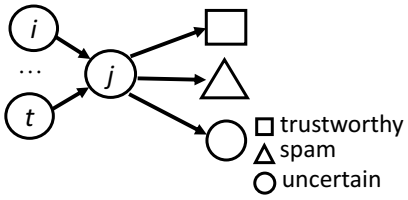


Fig. 1: Illustration of weight assignment in a hyperlink structure graph.

For trustworthy website detection, there are usually a group of websites that are labeled as either normal or spam by humans. This group of websites is commonly referred to as a labeled set. We further divide the labeled set into two groups: seed set and testing set. While the former is used to initialize trust relations between websites, the latter is used for evaluation purpose. As shown in Fig. 1, we now focus on the nodes that website j points to. These nodes could be either trustworthy, not trustworthy, or uncertain, depending on the nature of the corresponding websites. As such, we consider a labeled normal website as a trustworthy one, a spam website as a untrustworthy one. For those that are not analyzed by humans, or not being labeled, we call them undecided or uncertain websites.

We use g_j, s_j and u_j to denote the number of labeled good/normal websites, labeled spam websites, and undecided websites that j points to. Let b_j, d_j, n_j denote the probabilities that website j is trustworthy, not trustworthy, and uncertain, respectively. According to the three-valued subjective logic [21] that is used to model trust in OpinionWalk [22], these probabilities can be computed as follows.

$$\begin{cases} b_j = \dfrac{g_j}{g_j + s_j + u_j + 3} \\ d_j = \dfrac{s_j}{g_j + s_j + u_j + 3} \\ n_j = \dfrac{u_j}{g_j + s_j + u_j + 3} \\ e_j = \dfrac{3}{g_j + s_j + u_j + 3} \end{cases} . \tag{1}$$

In the above equations, e_j denotes the prior uncertainty existing in website j. As we can see, if website j points to no other website, the values of b_j, d_j and n_j are zeros and $e_j = 1$, indicating the trustworthiness of website j is unknown or fully uncertain. In this case, we assume website j points to 3 virtual websites, a normal one, a spam one, and an uncertain one. This is why e_j is called as the prior uncertainty of website j. The assumption is reasonable because Eq. 1 considers both prior (e_j) and posterior uncertainties (n_j) and still works even website j does not point to any other website. More details about the difference between prior and posterior uncertainties can be found in [21].

Based on the above analysis, we model the trustworthiness between two websites i and j as an opinion vector $\omega_{ij} = (b_{ij}, d_{ij}, n_{ij}, e_{ij}) = (b_j, d_j, n_j, e_j)$. ω_{ij} indicates how trustworthy website j is, from website i's perspective. As shown in Fig. 1, if there is another website t also pointing to j, we have $\omega_{tj} = \omega_{ij} = (b_j, d_j, n_j, e_j)$. Note that $\omega_{ij} = \omega_{tj}$, which is different from OpinionWalk that assumes different users have different opinions on the same user.

If there is no hyperlink from i to j, i.e., $e(i, j) \notin E$, then we define i has an uncertain opinion $\mathbb{O}$ on j as $\mathbb{O} \triangleq (0, 0, 0, 1)$ that indicates a website is totally uncertain about whether another website is trustworthy. For ω_{ii}, a certain opinion $\mathbb{I}$ is defined as $\mathbb{I} \triangleq (1, 0, 0, 0)$ that indicates a website absolutely trusts itself.

2.3 Opinion Matrix Initialization

Given a hyperlink graph $G = (V, E, W)$ containing n nodes and a subset $S \subset V$ with labeled nodes, we can obtain the opinion matrix M, as it is defined in [22]. In the matrix, $\omega_{ij} = (b_j, d_j, n_j, e_j)$ is calculated from Eq. 1, if $e(i, j) \in E$; otherwise, $\omega_{ij} = \mathbb{O}$ except for $\omega_{ii} = \mathbb{I}$.

$$M = \begin{bmatrix} \mathbb{I} & \omega_{12} & ... & \omega_{1n} \\ \omega_{21} & \mathbb{I} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \omega_{n1} & \cdots & \cdots & \mathbb{I} \end{bmatrix}.$$

The opinion matrix records all the trust relations among websites that are connected to each other. This matrix will then be used to compute trustworthiness of all websites, which will be introduced in later sections.

Algorithm 1 shows how to initialize the opinion matrix, based on a directed graph G and a labeled seed set S. Please note that S is composed of websites that are selected from the labeled set with seed selection method described in

section 3.1.1. Lines 1-9 initialize the opinion matrix M with $\mathbb{I}$ or $\mathbb{O}$. Lines 10-28 update $M[i][j]$ based on the number of normal, spam and unlabelled websites that j points to. Finally, line 29 returns the opinion matrix M.

Algorithm 1 GetOpinionMatrix(G, S)

Require: Directed graph G, labeled seed set S.
Ensure: Opinion matrix M.
1: **for all** node i **do**
2: **for all** node j **do**
3: **if** $i = j$ **then**
4: $M[i][j] = \mathbb{I}$
5: **else**
6: $M[i][j] = \mathbb{O}$
7: **end if**
8: **end for**
9: **end for**
10: **for all** node i **do**
11: **for all** nodes j s.t. $e(i, j) \in E$ **do**
12: $g_j \leftarrow 0$, $s_j \leftarrow 0$, $u_j \leftarrow 0$;
13: **for all** nodes p s.t. $e(j, p) \in E$ **do**
14: **if** $p \in S$ and p is normal **then**
15: $g_j \leftarrow g_j + 1$
16: **else**
17: **if** $p \in S$ and p is spam **then**
18: $s_j \leftarrow s_j + 1$
19: **else**
20: $u_j \leftarrow u_j + 1$
21: **end if**
22: **end if**
23: **end for**
24: $m \leftarrow g_j + s_j + u_j + 3$;
25: $b_j \leftarrow g_j/m$, $d_j \leftarrow s_j/m$, $n_j \leftarrow u_j/m$, $e_j \leftarrow 3/m$;
26: $M[i][j] \leftarrow (b_j, d_j, n_j, e_j)$;
27: **end for**
28: **end for**
29: **return** M

2.4 Trust Propagation and Combination

Given two edges $e(i, s)$ and $e(s, j) \in E$ with the weights $\omega_{is} = (b_{is}, d_{is}, n_{is}, e_{is})$ and $\omega_{sj} = (b_{sj}, d_{sj}, n_{sj}, e_{sj})$, we are able to compute $\omega_{ij} = \Delta(\omega_{is}, \omega_{sj})$. In other words, s's opinion on the trustworthiness of website j can be propagated to website i so that i derives its own opinion about j's trustworthiness. The above-mentioned process is call trust propagation in social networks. To distinguish from direct opinions, we use Ω_{ij} to denote i's indirect opinion on j. In this way, Ω_{ij} can be computed from the following equations [21].

$$\begin{cases} b_{ij} = b_{is} b_{sj} \\ d_{ij} = b_{is} d_{sj} \\ n_{ij} = 1 - b_{ij} - d_{ij} - e_{sj} \\ e_{ij} = e \end{cases}, \tag{2}$$

where $e = e_{sj}$ if $e_{is} \neq 1$, otherwise, $e = 1$. That implies the prior uncertainty in i's opinion on j is determined by that of ω_{sj} if ω_{sj} is not uncertain; otherwise, ω_{ij} will be uncertain. Note that ω_{is} and ω_{sj} can be replaced by indirect opinions Ω_{is} and Ω_{sj}.

If there are several different paths from website i to website j, we can combine these opinions. Let $\Omega_{ij}^1 = (b_{ij}^1, d_{ij}^1, n_{ij}^1, e_{ij}^1)$ and $\Omega_{ij}^2 = (b_{ij}^2, d_{ij}^2, n_{ij}^2, e_{ij}^2)$ be two different opinions derived from two parallel paths from i to j. Then, a new opinion $\Omega_{ij} = (b_{ij}, d_{ij}, n_{ij}, e_{ij})$ can be generated by the combining operation $\Theta(\Omega_{ij}^1, \Omega_{ij}^2)$ as follows [21]:

$$\begin{cases} b_{ij} = \dfrac{e_{ij}^2 b_{ij}^1 + e_{ij}^1 b_{ij}^2}{e_{ij}^1 + e_{ij}^2 - e_{ij}^1 e_{ij}^2} \\ d_{ij} = \dfrac{e_{ij}^2 d_{ij}^1 + e_{ij}^1 d_{ij}^2}{e_{ij}^1 + e_{ij}^2 - e_{ij}^1 e_{ij}^2} \\ n_{ij} = \dfrac{e_{ij}^2 n_{ij}^1 + e_{ij}^1 n_{ij}^2}{e_{ij}^1 + e_{ij}^2 - e_{ij}^1 e_{ij}^2} \\ e_{ij} = \dfrac{e_{ij}^1 e_{ij}^2}{e_{ij}^1 + e_{ij}^2 - e_{ij}^1 e_{ij}^2} \end{cases}. \tag{3}$$

For a computed opinion $\Omega_{ij} = (b_{ij}, d_{ij}, n_{ij}, e_{ij})$, it contains four values and cannot be directly used to sort websites. We need to convert it to a single trust value. The Eq. 4 is used to calculate the probability that j is a trustworthy website, where x and y are the coefficients of posterior uncertainty and prior uncertainty, indicating how much of the posterior uncertainty and prior uncertainty are credible, respectively.

$$E(\Omega_{ij}) = b_{ij} + x \times n_{ij} + y \times e_{ij} \tag{4}$$

2.5 Enhanced OpinionWalk Algorithm

Based on the previous discussion, opinions are used to quantify the trust relations between individual websites. Specifically, the opinions are derived from the labeled websites, based on formula (1). With these trust opinions, the opinion matrix can be initialized that reflects the social connections among websites and the strength of these connections. OpinionWalk algorithm starts from a seed node to search the network level by level, and the trustworthiness of all websites are iteratively obtained during the searching process. The trust computation is then realized by carrying out propagating and combining operations on opinions. However, the OpinionWalk algorithm updates the trustworthiness of all websites iteratively and every opinion needs to be recalculated in each iteration. Let's assume the algorithm starts from website i and aims at computing the trustworthiness of all other websites. The trustworthiness values are recorded in

$$Y_i^{(k)} = [\Omega_{i1}^{(k)}, \Omega_{i2}^{(k)}, \cdots, \Omega_{ij}^{(k)}, \cdots, \Omega_{in}^{(k)}]^T,$$

where $\Omega_{ij}^{(k)}$ denotes i's opinion about the trustworthiness of website j, after the algorithm searches k levels in the network. As a result, the execution time of OpinionWalk is not favorable for quick trustworthy website detection. To address this issue, in this section, we introduce the Enhanced OpinionWalk (EOW) algorithm that updates fewer opinions in each iteration, and thus results in a shorter running time.

When the algorithm starts from website i, the opinion vector $Y_i^{(1)} = [\omega_{i1}, \omega_{i2}, \cdots, \omega_{ij}, \cdots, \omega_{in}]^T$ is initialized, based on the direct links among websites. Next, we show how to obtain $Y_i^{(k+1)}$ from $Y_i^{(k)}$, which occurs when the algorithm moves from the k-th level to $(k+1)$-th level in the network. For a trust opinion in $Y_i^{(k)}$,

e.g., $\Omega_{is}^{(k)}$, if it is not an uncertain opinion $\mathbb{O}$ and $\omega_{sj} \neq \mathbb{O}$ where $\omega_{sj} \in M$ and $s \neq i \neq j$, we can compute a new opinion $\Delta(\Omega_{is}^{(k)}, \omega_{sj})$, based on the trust propagation operation. It denotes i's opinion on the trustworthiness of j, based on its k-hop "friend" website s's "recommendation". As shown in Fig. 2, if there exist m nodes that can make this type of recommendation, labeled as s_1, s_2, $\cdots$, s_m, we can combine the m newly obtained opinions to get a new opinion $\Theta(\Delta(\Omega_{is_1}^{(k)}, \omega_{s_1j}), \Delta(\Omega_{is_2}^{(k)}, \omega_{s_2j}), \cdots, \Delta(\Omega_{is_m}^{(k)}, \omega_{s_mj}))$. As the opinion expresses i's most current opinion on j's trustworthiness, we denote it as $\Omega_{ij}^{(k+1)}$. In the same way, we can update all elements in $Y_i^{(k)}$ to form $Y_i^{(k+1)}$.

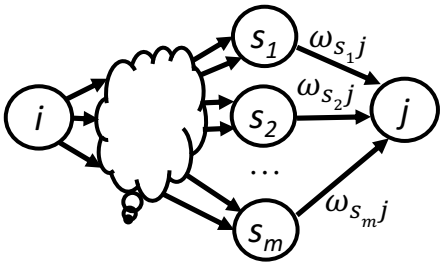


Fig. 2: Illustration of trustworthiness update from k-th level to $(k+1)$-th level in the network.

From the above description, we can see that $Y_i^{(k+1)}$ is solely determined by $Y_i^{(k)}$ and M, and not all opinions are changed in the updating process. In fact, if $\Omega_{ij}^{(k)}$ changes when the algorithm is processing the k-th level of the network, then only i's opinions on j's adjacent nodes need to be recalculated in the next iteration. We propose a mechanism to keep track of which elements in the opinion vector $Y_i^{(k)}$ need to be updated and only update those opinions in $Y_i^{(k+1)}$. In the enhanced OpinionWalk (EOW) algorithm, there exists a Boolean vector $F^{(k)} = [f_1, f_2, \cdots, f_j, \cdots, f_n]^T$ that indicates whether i's opinion on j needs to be updated in the $(k+1)-$th iteration. If f_j equals to 1, $\Omega_{ij}^{(k+1)}$ needs to be recalculated and unchanged otherwise. With the subtle modification on the OpinionWalk algorithm, the (average) execution time of EOW is only 72.9% of OpinionWalk's. We will use the example shown

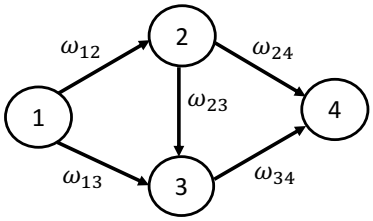


Fig. 3: An example of illustrating the Enhanced OpinionWalk algorithm.

in Fig. 3 to illustrate how the EOW algorithm works. With the given network, we derive the opinion matrix as follows.

$$M = \begin{bmatrix} \mathbb{I} & \omega_{12} & \omega_{13} & \mathbb{O} \\ \mathbb{O} & \mathbb{I} & \omega_{23} & \omega_{24} \\ \mathbb{O} & \mathbb{O} & \mathbb{I} & \omega_{34} \\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{I} \end{bmatrix}.$$

Let's assume EOW starts from node 1, then we have

$$Y_1^{(1)} = [\mathbb{I}, \omega_{12}, \omega_{13}, \mathbb{O}]^T,$$

and $F^{(1)} = [0, 0, 1, 1]^T$ that means $\Omega_{13}^{(1)}$ and $\Omega_{14}^{(1)}$ will be updated next but $\Omega_{11}^{(1)}$ and $\Omega_{12}^{(1)}$ remain the same, when EOW searches the second level of the network. For node 3, there is an opinion $\omega_{12} \in Y_1^{(1)}$, and an opinion $\omega_{23} \in M$, we can update node 1's opinion on node 3 via node 2 to $\Delta(\omega_{12}, \omega_{23})$. Similarly, because there is an opinion $\omega_{11} \in Y_1^{(1)}$ and $\omega_{13} \in M$, node 1 gets a new opinion on node 3 as follows.

$$\Delta(\omega_{11}, \omega_{13}) = \Delta(\mathbb{I}, \omega_{13}) = \omega_{13}$$

These two newly obtained opinion will be combined to form node 1's most current opinion on the node 3's trustworthiness.

$$\Omega_{13}^{(2)} = \Theta(\omega_{13}, \Delta(\omega_{12}, \omega_{23}))$$

For node 4, we have $\omega_{12} \in Y_1^{(1)}$ and $\omega_{24} \in M$, node 1's opinion on node 4 is updated to $\Delta(\omega_{12}, \omega_{24})$. With $\omega_{13} \in Y_1^{(1)}$ and $\omega_{34} \in M$, node 1 gets another opinion on node 4, i.e. $\Delta(\omega_{13}, \omega_{34})$. Combining these two opinions yields

$$\Omega_{14}^{(2)} = \Theta(\Delta(\omega_{12}, \omega_{24}), \Delta(\omega_{13}, \omega_{34})).$$

Therefore, after EOW algorithm finishes searching the second level, the opinion vector is updated to $Y_1^{(2)}$:

$$[\mathbb{I}, \omega_{12}, \Theta(\omega_{13}, \Delta(\omega_{12}, \omega_{23})), \Theta(\Delta(\omega_{12}, \omega_{24}), \Delta(\omega_{13}, \omega_{34}))]^T$$

After this iteration, because Ω_{13} and Ω_{14} change, we need to update the Boolean vector to $F^{(2)} = [0, 0, 0, 1]^T$ indicating node 1 will update its opinion on node 4 but keep its opinions on other nodes unchanged, in the next round. This is because node 4 is the adjacent neighbor of node 3.

When EOW searches the third level, because there exist $\omega_{12} \in Y_1^{(2)}$ and $\omega_{24} \in M$, we update Ω_{14} to $\Delta(\omega_{12}, \omega_{24})$. With $\Omega_{13}^{(2)} \in Y_1^{(2)}$ and $\omega_{34} \in M$, node 1 has a new opinion on 4, $\Delta(\Omega_{13}^{(2)}, \omega_{34})$. Combining these two opinions, node 1 derives a new opinion on node 4 as follow.

$$\Omega_{14}^{(3)} = \Theta(\Delta(\omega_{12}, \omega_{24}), \Delta(\Omega_{13}^{(2)}, \omega_{34})).$$

As such the opinion vector is update to

$$Y_1^{(3)} = \left[\mathbb{I}, \omega_{12}, \Omega_{13}^{(2)}, \Omega_{14}^{(3)}\right]^T = [\mathbb{I}, \omega_{12}, \Theta(\omega_{13}, \Delta(\omega_{12}, \omega_{23})), \Theta(\Delta(\omega_{12}, \omega_{24}), \Delta(\Theta(\omega_{13}, \Delta(\omega_{12}, \omega_{23})), \omega_{34}))]^T$$

After this iteration, because node 4 has no adjacent node, we have $F^{(3)} = [0, 0, 0, 0]^T$. That also means the EOW algorithm stops and node 1's opinions on the trustworthiness of all other nodes are obtained.

Algorithm 2 describes how to get the trustworthiness of all other websites, from website i's perspective. Line 1 calls Algorithm 1 to obtain the opinion matrix M. Lines 2-5 initialize F and $Y_i^{(1)}$, based on M. Lines 6-12 update the bit corresponding to i's adjacent nodes to 1. Line 13 initializes the searching level k to 1. Line 14 controls how many levels that EOW algorithm will search on the graph G. Lines 15-29 compute $Y_i^{(k+1)}$ based on $Y_i^{(k)}$ and M, and update the Boolean vector F accordingly. Line 15 copies all opinions from $Y_i^{(k)}$ to $Y_i^{(k+1)}$. Lines 16-21 recalculate node i's opinions on the websites whose trustworthiness need to be updated. Lines 18-20 combine all opinions derived from $\omega_{sj} \neq \mathbb{O}$. Lines 22-29 update $F[j]$ based on

the information of which element in $Y_i^{(k+1)}[j]$ is different from that of $Y_i^{(k)}[j]$. Finally, the vector $Y_i^{(k)}$ will contain node i's opinions on all other nodes, after EOW searches H levels within the graph G. By increasing the value of H, more accurate trust computation is expected, however, it will increase the execution time of the EOW algorithm. In practice, the value H is set to be a number ranging from 3 to 6 to achieve the good trade-off between accuracy and performance.

Algorithm 2 EOW(G,S,i,H)

Require: A directed graph G, labeled sample set S, starting node i, maximum searching depth H.
Ensure: i's opinion on j where $j \neq i$.

$M = GetOpinionMatrix(G, S)$;
for all node j **do**
  $Y_i^{(1)}[j] \leftarrow M[i][j]$
  $F[j] \leftarrow 0$
end for
for all node j **do**
  if $Y_i^{(1)}[j] \neq \mathbb{O}$ **then**
    for all node q s.t. $M[j][q] \neq \mathbb{O}$ **do**
      $F[q] \leftarrow 1$
    end for
  end if
end for
$k \leftarrow 1$
while $k < H$ **do**
  $Y_i^{(k+1)} \leftarrow Y_i^{(k)}$
  for all node $j \neq i$ s.t. $F[j] = 1$ **do**
    $Y_i^{(k+1)}[j] \leftarrow \mathbb{O}$
    for all node s s.t. $Y_i^{(k)}[s] \neq \mathbb{O}$ s.t. $M[s][j] \neq \mathbb{O}$ **do**
      $Y_i^{(k+1)}[j] \leftarrow \Theta(Y_i^{(k+1)}[j], \Delta(Y_i^{(k)}[s], M[s][j]))$
    end for
  end for
  for all node j **do**
    $F[j] \leftarrow 0$
  end for
  for all $j \neq i$ s.t. $Y_i^{(k+1)}[j] \neq Y_i^{(k)}[j]$ **do**
    for all node q s.t. $M[j][q] \neq \mathbb{O}$ **do**
      $F[q] \leftarrow 1$
    end for
  end for
  $k \leftarrow k + 1$
end while
return $Y_i^{(k)}$

3 Experimental Results and Analysis

In this section, we first introduce the experimental setup and evaluation metrics. Then, we provided a detailed experimental result analysis.

3.1 Dataset and Experiment Setup

To study trustworthy or spam website detection, there are two publicly available datasets WEBSPAM-UK2006 and WEBSPAM-UK2007 [1]. The WEBSPAM-UK2007 dataset includes 114,529 websites with 6479 websites labeled (5.65%), while WEBSPAM-UK2006 contains 65.5% labeled hosts. Generally more labeled data lead to more detailed evaluation, so we adopt WEBSPAM-UK2006 dataset in evaluating the proposed EOW algorithm. The WEBSPAM-UK2006 dataset was collected by a web crawler through "crawling" the .uk domain. It contains 77.9 million web pages, which equals to 11402 websites with 7473 websites/hosts labelled as either normal/trustworthy or spam.

3.1.1 Seed Selection

Because both TrustRank and EOW start from a set of seed nodes/websites to search for more trustworthy websites, in this section, we first introduce how seed nodes are usually chosen. The commonly-used seed selection methods for trustworthy website detection are high PageRank and inverse PageRank [18]. Applying the PageRank algorithm on a website hyperlink network, every website will be assigned a PageRank score. If the websites are sorted based on their scores, in a descending order, then those located at the top of the list are considered trustworthy. Among these website, the labelled trustworthy hosts are chosen as the seed nodes, which is why this algorithm is called high PageRank. On the other hand, inverse PageRank algorithm first inverts the links in the network, and then performs PageRank on the inverted graph. These labeled websites with large PageRank scores are then treated as the seed nodes.

3.1.2 Evaluation

The proposed EOW algorithm will rank all websites based on their trust scores (derived from trust opinions), and those with higher trust values (e.g., the top N websites) are considered trustworthy websites. As expected, an algorithm is effective in searching for trustworthy websites if it identify more normal and fewer spam samples in the top N chosen websites. To evaluate the EOW algorithm's performance, we sort 11402 hosts in a descending order, based on their trust scores. We then measure the ratio/percentage of labeled normal hosts (true positive) and labeled spam hosts (false positive). The value of N varies from 1000 to 4000. With the same performance measure, the EOW algorithm is compared with TrustRank algorithm, the state-of-art spam detection algorithm based on trust propagation.

3.1.3 Parameter setting

For the seed selection algorithms, i.e., high PageRank and inverse PageRank algorithms, the iteration time and decay factor are set to be 20 and 0.85, respectively. In [5], it is reported that the decay factor of 0.85 was considered a standard setting in PageRank. It is also believed that 20 interactions are enough for PageRank algorithms to produce converged results [18]. Because TrustRank is a variation of the PageRank algorithm, we adopt the same parameter settings in TrustRank. For the EOW algorithm, after the seed nodes are chosen, it iteratively propagates trust from a starting node to all other nodes in the network. Note that every normal sample in the seed set can be used as a starting node. We pick the top 100 or 200 normal samples as the starting nodes, and select the best result to evaluate EOW's performance. To understand how the maximum searching depth affects the EOW's performance, we set the value to be 6, 10, and 20, respectively. The reason of choosing 20 as the maximum depth is that the iteration time of PageRank algorithm is 20. To make a fair comparison with PageRank, we adopt 20 as the maximum searching depth of EOW

1. http://chato.cl/webspam/datasets/

TABLE 1: Parameter setting

Parameters	Values
# of normal websites	100, 200, 500, 1000, 1500, 2000, 3000
# of spam websites	50, 100, 500, 1000
Maximum searching depth	6, 10, 20

algorithm. On the other hand, based on the "six degrees of separation" theory, i.e., a person can be connected to any other person through a chain of acquaintances that has no more than five intermediaries, we adopt 6 as the maximum searching depth of EOW algorithm. The parameter settings of our experiments are summarized in Table 1.

3.2 Impact of Seed Selection

Good seeds tend to generate good results, therefore, we evaluate the impacts of high PageRank and inverse PageRank, as seed selection methods, on the performance of TrustRank and EOW algorithms. Specifically, we carry out different experiments with either the high PageRank or the inverse PageRank as the seed selection scheme. We execute these two algorithms with the same number of iterations and decay factor. We then choose the top 100 normal samples as the seeds for TrustRank and EOW algorithms. Moreover, for the EOW algorithm, we set its maximum searching depth as 6. Finally, we compare the performance of TrustRank and EOW, regarding to their true positive and false positive values.

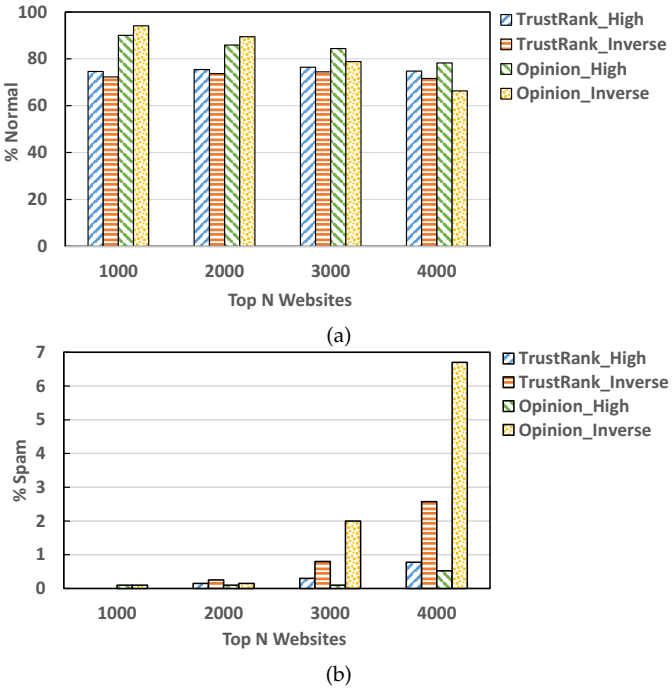


Fig. 4: (a) Percentages of labeled normal websites identified by TrustRank and EOW with different seed selection algorithms. (b) Percentages of labeled spam websites identified by TrustRank and EOW with different seed selection algorithms.

Fig. 4 shows the percentages of labeled normal and labeled spam hosts, within the top N websites, detected by the TrustRank and EOW algorithms. With different seed selection algorithms, we summarize the results as follows. TrustRank_High and Opinion_High denote the results generated by TrustRank and EOW with high PageRank as the seed selection algorithm. TrustRank_Inverse and Opinion_Inverse indicate the results obtained using inverse PageRank to select seeds. From Fig. 4 we can see that the true positive of TrustRank_High is slightly better than that of TrustRank_Inverse, and the false positive of TrustRank_High is also smaller than that of TrustRank_Inverse. This implies high PageRank selects better seeds, which is opposite to the conclusion that inverse PageRank is slightly better in choosing seeds in [18]. For the EOW algorithm, on the other hand, Opinion_Inverse generates larger true positive values than Opinion_High, when $N = 1000, 2000$. The true positive values of Opinion_Inverse, however, are smaller than those of Opinion_High when $N = 3000$ and 4000. When we look at the false positive values, Opinion_High performs much better than Opinion_Inverse. Overall, we conclude that high PageRank algorithm offers a better seed selection when N is greater than 2000, which is the common case for trustworthy website detection.

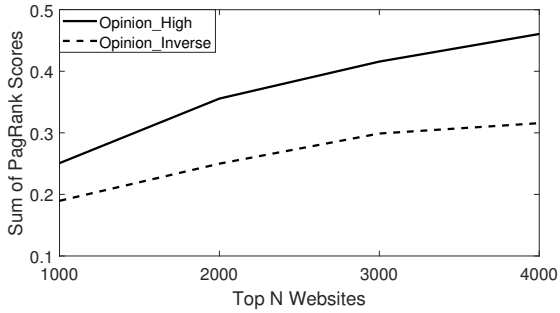


Fig. 5: Sum of PageRank scores of top N websites .

Because PageRank evaluates a website as trustworthy if and only it is linked from several other trustworthy websites, we speculate hosts with high PageRank scores tend to link to each other. To validate our hypothesis, we sort all websites based on their scores and sum up the scores of all top N websites generated by the EOW algorithm. As we can see from Fig. 5, different seed selection algorithm yields different results. In detail, the sum of PageRank scores of top N websites identified by Opinion_High is much higher than that generated by Opinion_Inverse. *It also implies that high PageRank algorithm, serving as the seed selection algorithm, offers EOW the best opportunity in identifying more trustworthy websites.* Because high PageRank seed selection method makes both EOW and TrustRank algorithms identify more normal websites (and fewer spams), we adopt it in the rest of our experiments.

3.3 Influence of Number of Seeds

To understand the influence of number of seeds on EOW's performance, we conduct experiments by varying the numbers of normal and spam samples in the seed sets. We use OpinionX_Y to denote the results generated by EOW, with X normal and Y spam samples in the corresponding seed set. If there is no spam sample in the seeds, the results are labeled as Opinion$X$, indicating X normal samples in the seeds. Opinion100_50 and Opinion100_100 use the top 100 normal samples as the starting nodes, while Opinion200, Opinion2000 and Opinion2000_500 use the top 200 normal samples as the starting nodes. The maximum searching depth of EOW is set to be 6.

Fig. 6 shows the true positive and false positive values of EOW with different number of seeds. We can see that the percentages of labeled normal and spam samples of Opinion100_50 are the same as Opinion100_100. Similarly, the portions of labeled normal and spam samples of Opinion2000 are the same as Opinion2000_500. Furthermore, we find that Opinion2000 and Opinion2000_500 almost always generate the same results. For example, the trust opinion on host "www.comp.rgu.ac.uk" is (0.071527,0,0.927648,8.25E-04) in Opinion2000 and (0.071527,6.33E-04,0.927015,8.25E-04) in Opinion2000_500, respectively. All the observations leads to one conclusion: *spam samples in seed sets have almost no impact on EOW algorithm.* This is mainly because the value of d in an opinion becomes smaller and smaller when trust propagates within the network.

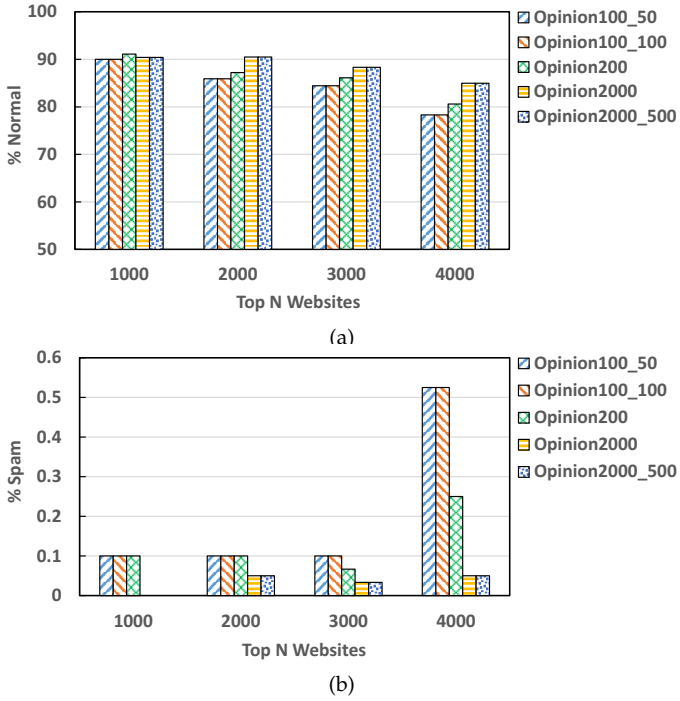


Fig. 6: Percentages of labeled (a) normal websites and (b) spam websites identified by EOW with different numbers of seeds.

Comparing the results of Opinion100_100, Opinion200 and Opinion2000, we can clearly observe that the values of true positive are getting higher and the false positive values are becoming smaller. In other words, *the number of normal samples in a seed set plays a critical role in EOW detecting trustworthy websites* . This is mainly because EOW leverages large amount of trustworthy websites to reach/identify more trustworthy ones, thanks to the principle of trust propagation [20]. For the top 1000 websites identified by EOW, the percentages of labeled normal samples in Opinion100_100, Opinion200 and Opinion2000 are almost the same. Additionally, there is only 1 spam sample, detected as trustworthy website by Opinion100_100 and Opinion200, within the top 1000 websites, which turns out to be a 0.1% false positive rate. We further manually check all the unlabeled websites in the top 1000 ones in Opinion100, and find that there are only 2 spam websites (with 10 more websites that are not accessible). These observation conclude that *EOW algorithm performs well, with 100 normal seeds, if we are only interested in detecting no more than 1000 trustworthy websites.*

3.4 Influence of Maximum Searching Depth

As the EOW algorithm searches deeper on the hyperlink graph, more accurate trust evaluation on websites are expected. The execution time of EOW algorithm, however, will increases significantly when the searching depth is getting larger. To identify the best yet effective search depth, we carry out experiments with the different searching depths in EOW. Specifically, we set it to be 6 and 20, given 200 normal samples in the seed set. We also set the maximum searching depth as 6, 10 and 20, when the seed set is composed of 2000 normal samples. Fig. 7 provides the experimental results where Opinion$X(Y)$ represents the results generated by EOW with a searching depth of Y and X normal seed samples.

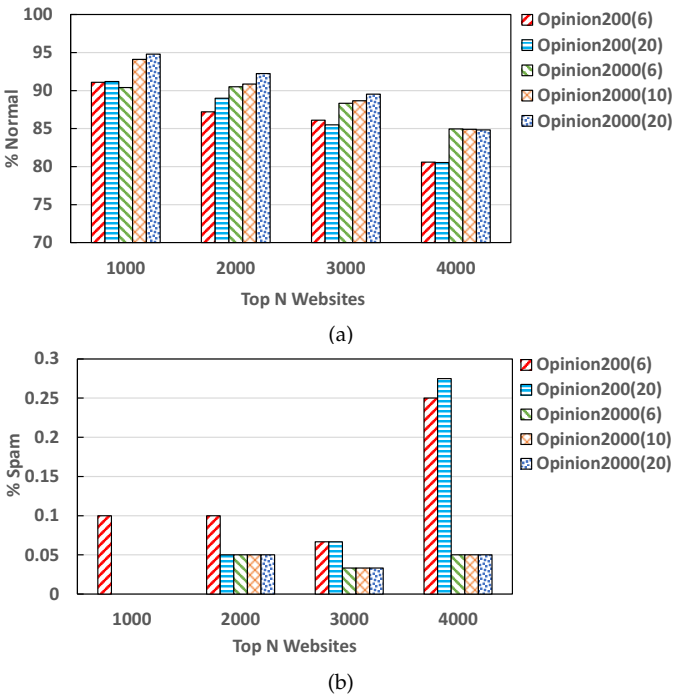


Fig. 7: Performance of the EOW algorithm in detecting trustworthy websites with different setting of the maximum searching depth.

In Fig. 7, we find that the performance of Opinion200(6) and Opinion200(20) are very similar, i.e., deeper searching does not always lead to better results, given 200 normal seeds. This may be because EOW algorithm is not able to reach more trustworthy websites, due to the limited amount of trustworthy seeds that it can start with. This reason can also explain the following observations: the results of Opinion2000(20) are slightly better than Opinion2000(10) which is in turn slightly better than 2000(6). In a word, EOW with a deeper searching and a larger seed set tends to detect more trustworthy websites. To achieve a better performance, however, EOW takes a longer time to search the whole network. Table 2 gives the running times of EOW with different search depths. In the table, we can see that Opinion2000(10) takes about two times of the time taken by Opinion2000(6), and Opinion2000(20) needs 4.5 times as many as that of Opinion2000(6). The computer used in our

TABLE 2: Running time of EOW algorithms with different maximum searching depths

Maximum Searching Depth	Running Time (Second)
6	92.565
10	188.229
20	420.072

experiments has an Intel R5 CPU, 4G memory, and 320G hard disk. The operating system is Windows10, and the programming language is Java.

When the searching depth is 6, EOW's performance is almost only determined by the size of seeds; therefore, we conclude that *a search depth of 6 is adequate for EOW to work properly in detecting normal websites.* This conclusion is also supported by the "six degrees of separation" theory. It seems that the hyperlink network constructed among websites also represent the "small world" phenomena. Therefore, we set the maximum searching depth as 6 in the rest of our experiments.

3.5 Impact of Starting Nodes

Every normal sample in the chosen seed set can be used as a starting node of the EOW algorithm. Therefore, the performance of the EOW algorithm may vary if it starts from different seed nodes. To investigate whether and how different starting nodes will affect EOW's performance, we conduct a series of experiments with 1000 normal websites in the seed set. Out of the 1000 samples, we randomly select 200 websites to serve as the starting nodes. We plot the performance of EOW in Fig. 8 by showing the best, worst and combined results. For every node in the 200 websites, we let EOW algorithm start from it and rank all websites accordingly. From the ranking results, we identify the cases where EOW gives the best and worst performance and labeled them as "Best1000" and "Worst1000", respectively. For each website, different trust opinions will be generated if EOW starts from different seeds. We then combine all these trust opinions, according to the combining operation defined previously, to derive the "overall" trust opinion of the website. All websites with combined trust opinions will be ranked again, based on their (combined) trust values. The newly website ranking results are named as "Combine1000".

In Fig. 8, we can see the best results are better than the combined ones which are much better than the worst results. We also notice that *the combined results do not merely provide an average between the best and worst results, i.e., they lean very much towards the best results.* This implies that a combining operation of results derived from different seeds tend to offer a near optimal result, which should be considered a practical solution. To understand the impact of different starting nodes on EOW, we zoom into the results and present two examples here. When EOW starts from the website www.jobs.ac.uk, EOW places 1728 labelled normal and 1 labelled spam websites in its top 2000 trustworthy websites. On the other hand, when it starts from www.lovehoney.co.uk, EOW put 851 labelled normal and 122 labelled spam samples in its top 2000 trustworthy websites. Apparently, more spam websites could be treated as trustworthy if a bad starting node is chosen. To mitigate this issue, results generated by EOW using different starting nodes need to be combined. Thanks to the opinion combining operation, untrustworthy websites are likely to be eliminated from the final results, while trustworthy ones may not suffer from the process.

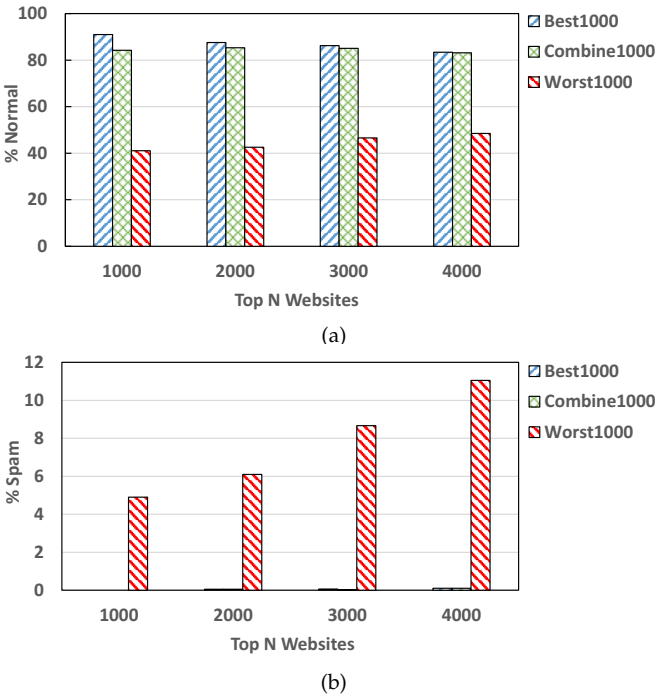


Fig. 8: Performance of EOW algorithm in detecting trustworthy websites with different starting nodes.

We further analyze the characteristics of the starting nodes and try to discover general rules in selecting good starting nodes for EOW. First, we find that *the outdegree and indegree of good starting nodes are much larger than the bad ones.* The average outdegree and indegree of the top 10 good starting nodes are 713.9 and 778.1, respectively. The outdegree and indegree of the best starting node are 1363 and 738, respectively. On the other hand, the average outdegree and indegree of the worst 10 starting nodes are 163.4 and 454.9, respectively. Five of these bad starting nodes are with a outdegree of 0. Second, we discover that *a good starting node tends to connect to more (labeled) normal websites, compared to the bad ones.* The average ratio of labeled normal websites linked from the top 10 good starting nodes is 87.09%, with the maximum ratio of 93.99%. On the other hand, the average ratio of the worst 10 starting nodes is only 21.5%. This observation is also in line with our conclusions about the importance of seed selection.

3.6 PageRank, TrustRank vs Enhanced OpinionWalk

This section compares the EOW algorithm with PageRank and TrustRank algorithms, regarding to their performance in detecting trustworthy websites. For the EOW and TrustRank algorithms, we only report the results with 200 and 2000 normal seeds. Fig. 9 shows percentages of labeled normal and labeled spam websites in the top N websites detected by the three algorithms.

We can see that the true positive values of EOW are always the highest and the corresponding false positive values of EOW are the lowest, given the same number of seeds. The performance of PageRank is the worst, i.e., lowest

true positive and highest false positive values. The reason is that PageRank algorithm does not take into account any knowledge about the trustworthiness of a website, nor does it penalize a spam host. TrustRank considers not only the trust values of websites but also the link relationships among websites, it provides slightly better results. Because EOW algorithm relies on a more accurate trust representation of websites and rigorously defined trust propagation and combing operations, it yields the best results.

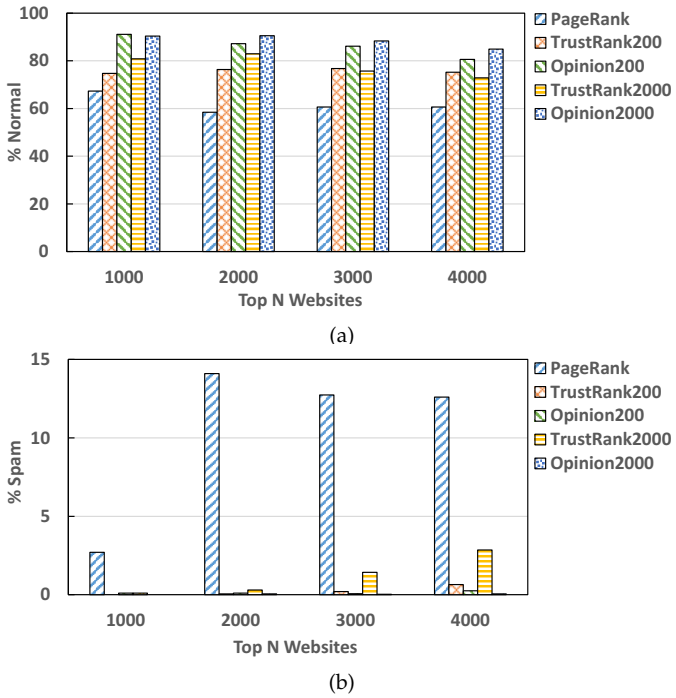


Fig. 9: Percentages of labeled (a) normal and (b) spam websites detected by different algorithms.

When the number of normal seeds is 200, the true positive values of EOW algorithm are 16.5%, 12.65%, 8.77%, 5.35% larger than TrustRank, in the top 1000, 2000, 3000 and 4000 websites, respectively. When the number of normal seeds is 2000, the values of EOW algorithm are 9.6%, 7.55%, 12.63%, 12.1% larger than TrustRank, in the top 1000, 2000, 3000 and 4000 websites, respectively. On the other hand, the corresponding false positive values of EOW algorithm are 0.1%, 0.25%, 1.4%, 2.8% smaller than TrustRank algorithm (in the top 1000, 2000, 3000 and 4000 websites). With 2000 normal seeds, the false positive values of EOW algorithm are 0.2%, 1.37% and 2.6% smaller than TrustRank (in the top 2000, 3000 and 4000 websites). From the above results, we conclude that *EOW algorithm is able to detect more trustworthy websites with fewer seeds, offering a higher true positive and lower false positive values.*

3.7 Experiment Results Summary

From the experimental results analysis, we can make the following conclusions. (i) EOW algorithm favors for high PageRank serving as the seed selection method, not the inverse PageRank. (ii) The selected spam seeds have almost no impact on EOW's performance, however, the number of normal seeds plays a critical role in detecting trustworthy websites. (iii) EOW algorithm achieves adequately good results when its maximum searching depth is 6. (iv) Starting EOW algorithm from nodes with larger outdegree and indegree tends to yield a better detection result. (v) EOW detects more trustworthy websites (and fewer spam websites) with fewer seeds, comparing to TrustRank.

4 Related Work

The problem of trustworthy website detection can be approached from two different directions. One type of solution is to identify spam or untrustworthy websites, based on either the contents or link structures of websites. An alternative solution to the problem is to rank websites based on their trust values so that trustworthy websites are placed on top of the list. Majority of existing research works belongs to the first category, while there are not adequate studies in the second one.

4.1 Trustworthy Website Ranking

The most popular algorithm of ranking website based on their trustworthiness/importance is the PageRank algorithm [5]. The intuition behind PageRank is that a web page is important if many other important web pages point to it. If the PageRank algorithm is executed on the hyperlink graph of websites, then every host in the link structure will be assigned an importance score. The websites are then sorted according to their PageRank scores, in a descending order, so those located at the top of the list are considered trustworthy. Truncated PageRank [2] is a variation of PageRank in that it disregards the paths with length that is below a certain threshold. Similarly, the inverse PageRank algorithm is proposed in [18]. The algorithm first reverses the directions of all edges and then perform the PageRank algorithm to rank websites.

Another important solution to trustworthy website ranking is the TrustRank algorithm [18] that separates normal websites from spams. In the algorithm, a small set of seed nodes are hand picked by experts. The trust scores of these seeds are set to be $\frac{1}{D}$, where D is the total number of seeds. The TrustRank algorithm then propagates trust through the out-links of seeds to discover nodes that are likely to be trustworthy. Topical TrustRank [24] is proposed to use the topical information of websites to partition seeds, and then calculate the trust scores for each topic. Finally, the trust scores of all topics of a website are combined to determine its ranking. BrowseRank [25] makes an attempt to solve the problem in a different way: it models user browsing behavior data (e.g., dwelling time, click data, etc.) to construct a browsing graph and then calculates the stationary dwelling distribution to capture the importance of a website. This method is a promising solution, however, the browsing behavior data are usually confidential and difficult to obtain.

In addition to trustworthy website ranking, there exist works on how to select seeds to improve ranking performance. Zhang Xianchao et al. [26] propose an automatic seed set expansion algorithm (ASE) that enriches a small seed set to a larger one. The basic idea of ASE is that if a page is recommended by many trustworthy pages, the page itself must be trustworthy. The links among websites can be considered a means of conveying recommendation, therefore, several links recommending the same page is called joint recommendation. With joint recommendation, the ASE algorithm is able to obtain more trustworthy seeds from the given ones.

4.2 Spam Website Detection

There is a flurry of research works on spam website detection. Existing techniques can be roughly categorized into three groups [4]: content analysis, link structure analysis, and user behavior analysis. In the first group, labeled web contents are analyzed to construct classifiers to detect spam websites [27], [28], [29], [30], [31], [32], [33], [34], [35], [36]. For example, Fetterly et al. propose to detect spam websites through statistical analysis of the terms included in a website [27]. Similarly, Ntoulas et al. [28] propose to detect spams by building up a classification model that combines multiple heuristics, based on page content analysis. Urvoy et al. [29] suggest to first cluster web pages based on their HTML similarity and then detect spams with a classification model. Biro et al. [30] use modified Latent Dirichlet Allocation (LDA) method to determine spam and non-spam topics, and then use them to classify spam websites.

In the second group, spam websites are detected by analyzing the link structure between websites [3], [23], [37], [38], [39], [40], [41], [42], [43]. Some of this group of methods are similar to those used in trustworthy website detection, however, they focus on detecting only spams. For example, Anti-TrustRank [23] starts from a set of seed nodes and propagates anti-trust scores within the network to identify spams. Wu et al. [38] propose to use a linear combination of trust and distrust values to detect spam pages. Others are based on classification techniques. Such as, C. Castillo et al. [3], for the first, time, integrate link structure and content attributes to detect web spams. In [40], a novel classication technique is presented, based on the Minimum Description Length (MDL) principle. R. C. Patil and D. R. Patil [41] implement a spam detection system, based on a supporting vector machine (SVM) classifier that combines new link features with content. O. M. Eugene [43] propose a Link Authorization Model (LAM) to detect link spam propagation onto neighboring pages. G. G. Geng et al. [44] propose a two-stage ranking strategy that makes good use of hyperlink information among websites and websites intra-structure information to combat link spam.

The last group consists of algorithms that exploit click stream data, user behaviour data, query popularity information and HTTP sessions information, to detect spams. Yiqun Liu et al. [45] propose to use three user behavior features to separate web spams from ordinary ones. S. Webb et al. [46] design a lightweight client-side web spam detection method that considers HTTP session information, instead of analyzing content-based and link-based features. F. Radlinski [47] propose to use personalized ranking functions to prevent click fraud manipulation. Xin Li et al. [48] use a bipartite-graph iterative algorithm to get higher precision and recall of click spam detections.

5 Conclusions and Future Work

The proposed EOW algorithm is well-suited for trustworthy website detection, which confirms our hypothesis that a hyperlink structure can be analyzed, using the techniques designed for social networks. Particularly, we find that an appropriate seed selection scheme (e.g., high PageRank) can significantly improve the trustworthy website detection results. Interestingly, the "small world" phenomena is also observed in a hyperlink space. In other words, the EOW algorithm only needs to search a hyperlink network for six levels, to accurately determine a website's trustworthiness, i.e., websites are six-degree separated in the hyperlink space. Compared to existing solutions, e.g., PageRank and TrustRank, the EOW algorithm is able to detect 5.35% - 16.5% more trustworthy websites. Meanwhile, EOW saves about 27.1% of execution time, compared to OpinionWalk that is designed for social network analysis.

As seed selection plays a critical role in determining the trustworthiness of all other websites, we believe a comprehensive study of seed selection is necessary to further improve the detection results. Possible solutions may explore the centrality, influence, and in- and out-degree of a website while selecting seeds. Another important direction is to investigate whether the EOW algorithm starting from multiple seeds will outperform that starting from a single seed. It might also be possible to combine the EOW algorithm together with classification-based methods that consider web contents, to decide a website's trustworthiness. Last but not the least, it is possible to speed up the EOW algorithm by dividing a hyperlink network into individual communities so that the algorithm only searches each community, instead of the whole network. This can be done by grouping websites based on their features, e.g., website category, keywords, and contents.

Acknowledgment

This research is supported by National Science Foundation (NSF) through grant CNS-1644348, Bureau of Science and Technology of Jinan City (grant 201401211), Higher Educational Science and Technology Program of Shandong Province (grant J15LN23), the Doctoral Foundation of Shandong Jianzhu University, and the China Scholarship Council.

References

[1] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz, "Analysis of a very large web search engine query log," *ACM SIGIR Forum*, vol. 33, no. 1, pp. 6–12, 1999.

[2] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates, "Using rank propagation and probabilistic counting for link-based spam detection," in *Proc. of WebKDD*, vol. 6, 2006.

[3] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: Web spam detection using the web topology," in *ACM SIGIR*. ACM, 2007, pp. 423–430.

[4] N. Spirin and J. Han, "Survey on web spam detection: principles and algorithms," *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 2, pp. 50–64, 2012.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[6] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[7] S. Jayanthi and S. Sasikala, "Wespact:detection of web spamdexing with decision trees in ga perspective," in *International Conference on Pattern Recognition*. IEEE, 2012, pp. 381–386.

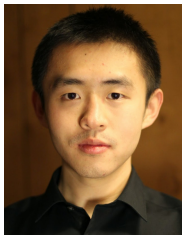
[8] J.-Y. Lin, H.-R. Ke, B.-C. Chien, and W.-P. Yang, "Designing a classifier by a layered multi-population genetic programming approach," *Pattern Recognition*, vol. 40, no. 8, pp. 2211–2225, 2007.

[9] X. NIU, J. MA, S. MA, and D. ZHANG, "Web spam detection by the genetic programming-based ensemble learning," *Journal of Chinese Information Processing*, vol. 26, no. 5, pp. 94–100, 2012.

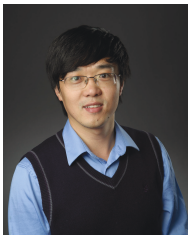
[10] X. Niu, J. Ma, Q. He, S. Wang, and D. Zhang, "Learning to detect web spam by genetic programming," *Web-Age Information Management*, pp. 18–27, 2010.

[11] X. Niu, S. Li, N. Yuan, X. Niu, and C. Zhu, "Link spam detection based on genetic programming," in *ICNC*, vol. 7. IEEE, 2010, pp. 3359–3363.
[12] A. H. Keyhanipour and B. Moshiri, "Designing a web spam classifier based on feature fusion in the layered multi-population genetic programming framework," in *FUSION*. IEEE, 2013, pp. 53–60.
[13] L. Shengen, N. Xiaofei, L. Peiqi, and W. Lin, "Generating new features using genetic programming to detect link spam," in *ICICTA*, vol. 1. IEEE, 2011, pp. 135–138.
[14] M. Iqbal, M. M. Abid, and M. Ahmad, "Catching webspam traffic with artificial immune system (ais) classification algorithm," in *ICSESS*. IEEE, 2016, pp. 402–405.
[15] A.-C. Enache and V. V. Patriciu, "Spam host classification using swarm intelligence," in *COMM*. IEEE, 2014, pp. 1–4.
[16] A.-C. Enache and V. Sgarciu, "Spam host classification using pso-svm," in *Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–5.
[17] A. Taweesiriwate, B. Manaskasemsak, and A. Rungsawang, "Web spam detection using link-based ant colony optimization," in *AINA*. IEEE, 2012, pp. 868–873.
[18] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 576–587.
[19] X. Li, Q. Yang, X. Lin, S. Wu, and M. Wittie, "itrust: Interpersonal trust measurements from social interactions," *IEEE Network*, vol. 30, no. 4, pp. 54–58, 2016.
[20] G. Liu, Q. Yang, H. Wang, S. Wu, and M. P. Wittie, "Uncovering the mystery of trust in an online social network," in *IEEE CNS*. IEEE, 2015, pp. 488–496.
[21] G. Liu, Q. Yang, H. Wang, X. Lin, and M. P. Wittie, "Assessment of multi-hop interpersonal trust in social networks by three-valued subjective logic," in *INFOCOM*. IEEE, 2014, pp. 1698–1706.
[22] G. Liu, Q. Chen, Q. Yang, B. Zhu, H. Wang, and W. Wang, "Opinionwalk: An efficient solution to massive trust assessment in online social networks," in *INFOCOM 2017*. IEEE, 2017, pp. 1–9.
[23] V. Krishnan and R. Raj, "Web spam detection with anti-trustrank," in *AIRWeb*, 2006, pp. 37–40.
[24] B. Wu, V. Goel, and B. D. Davison, "Topical trustrank: Using topicality to combat web spam," in *WWW*. ACM, 2006, pp. 63–72.
[25] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li, "Browserank: letting web users vote for page importance," in *ACM SIGIR*. ACM, 2008, pp. 451–458.
[26] X. Zhang, W. Liang, S. Zhu, and B. Han, "Automatic seed set expansion for trust propagation based anti-spam algorithms," *Information Sciences*, vol. 232, pp. 167–187, 2013.
[27] D. Fetterly, M. Manasse, and M. Najork, "Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages," in *ACM SIGMOD/PODS*. ACM, 2004, pp. 1–6.
[28] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," in *WWW*. ACM, 2006, pp. 83–92.
[29] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne, "Tracking web spam with html style similarities," *ACM Transactions on the Web (TWEB)*, vol. 2, no. 1, p. 3, 2008.
[30] I. Bíró, J. Szabó, and A. A. Benczúr, "Latent dirichlet allocation in web spam filtering," in *AIRWeb*. ACM, 2008, pp. 29–32.
[31] N. Dai, B. D. Davison, and X. Qi, "Looking into the past to better classify web spam," in *AIRWeb*. ACM, 2009, pp. 1–8.
[32] J. Martinez-Romo and L. Araujo, "Web spam identification through language model analysis," in *AIRWeb*. ACM, 2009, pp. 21–28.
[33] C. Dong and B. Zhou, "Effectively detecting content spam on the web using topical diversity measures," in *International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 2012, pp. 266–273.
[34] S. Bharti, B. Vachha, R. Pradhan, K. Babu, and S. Jena, "Sarcastic sentiment detection in tweets streamed in real time: a big data approach," *Digital Communications and Networks*, vol. 2, no. 3, pp. 108–121, 2016.
[35] J. Wan, M. Liu, J. Yi, and X. Zhang, "Detecting spam webpages through topic and semantics analysis," in *GSCIT*. IEEE, 2015, pp. 1–7.
[36] J. Hua and Z. Huaxiang, "Analysis on the content features and their correlation of web pages for spam detection," *China Communications*, vol. 12, no. 3, pp. 84–94, 2015.
[37] K. L. Goh, R. K. Patchmuthu, and A. K. Singh, "Distrust seed set propagation algorithm to detect web spam," *Journal of Intelligent Information Systems*, pp. 1–23, 2017.
[38] B. Wu, V. Goel, and B. D. Davison, "Propagating trust and distrust to demote web spam." *MTW*, vol. 190, 2006.
[39] S. Kumar, X. Gao, and I. Welch, "Novel features for web spam detection," in *ICTAI*. IEEE, 2016, pp. 593–597.
[40] R. M. Silva, T. A. Almeida, and A. Yamakami, "Towards web spam filtering using a classifier based on the minimum description length principle," in *ICMLA*. IEEE, 2016, pp. 470–475.
[41] R. C. Patil and D. Patil, "Web spam detection using svm classifier," in *ISCO*. IEEE, 2015, pp. 1–4.
[42] H. Jelodar, Y. Wang, C. Yuan, and X. Jiang, "A systematic framework to discover pattern for web spam classification," *arXiv preprint arXiv:1711.06955*, pp. 32–39, 2017.
[43] O.-M. Eugene, Z. Fengli, O. K. Adu-Boahen, and B. E. Yellakuor, "Spam detection through link authorization from neighboring nodes," in *ICeND*. IEEE, 2015, pp. 1–6.
[44] G.-G. Geng, C.-H. Wang, Q.-D. Li, and Y.-P. Zhu, "Fighting link spam with a two-stage ranking strategy," *Advances in Information Retrieval*, pp. 699–702, 2007.
[45] Y. Liu, R. Cen, M. Zhang, S. Ma, and L. Ru, "Identifying web spam with user behavior analysis," in *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*. ACM, 2008, pp. 9–16.
[46] S. Webb, J. Caverlee, and C. Pu, "Predicting web spam with http session information," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 339–348.
[47] F. Radlinski, "Addressing malicious noise in clickthrough data," in *Learning to rank for information retrieval workshop at SIGIR*, vol. 2007, 2007.
[48] X. Li, M. Zhang, Y. Liu, S. Ma, Y. Jin, and L. Ru, "Search engine click spam detection," in *CCIS*, vol. 2. IEEE, 2012, pp. 985–989.

Xiaofei Niu is an associate professor in the School of Computer Science and Technology at Shandong Jianzhu University, Jinan, Shandong, China. She received B.S. and M.S. degrees in Computer Science from Qufu Normal University and Shandong University, China, in 2001 and 2004, respectively. She received her Ph.D degree in Computer Science from Shandong University, China, in 2012. She worked in the School of Computer Science and Technology at Shandong Jianzhu University from 2004, and earned an associate professor in 2013. Her research interests include trust assessment, social network, information retrieval, and data mining.



Guangchi Liu is currently a research scientist in the research & development department of Stratifyd, Inc., Charlotte, NC, USA. He received his Ph.D. in Computer Science from Montana State University, USA. His research interests include Internet of things, trust assessment, social network, and wireless sensor network.



Qing Yang is an assistant professor in the Department of Computer Science and Engineering at University of North Texas, Denton, TX, USA. He received B.S. and M.S. degrees in Computer Science from Nankai University and Harbin Institute of Technology, China, in 2003 and 2005, respectively. He received his Ph.D degree in Computer Science from Auburn University in 2011. He worked as an assistant professor in the Gianforte School of Computing at Montana State University from 2011. His research interests include Internet of Things, trust model, network security and privacy. He serves as an Associate Editor of Security and Communication Networks journal.