A Belief Approach for Detecting Spammed Links in Social Networks

Salma Ben Dhaou, Mouloud Kharoune, Arnaud Martin, Boutheina Ben Yaghlane

A Belief Approach for Detecting Spammed Links in Social Networks

Salma Ben Dhaou[1], Mouloud Kharoune[2], Arnaud Martin[2] and Boutheina Ben Yaghlane[1]
[1]*LARODEC, ISG of Tunis, Tunisia*
[2]*Univ Rennes, CNRS, IRISA, France*
Arnaud.Martin, mouloud.kharoune@univ-rennes1.fr, salma.bendhaou@hotmail.fr

Keywords: Social Networks, Communities, Theory of belief functions, Probability

Abstract: Nowadays, we are interconnected with people whether professionally or personally using different social networks. However, we sometimes receive messages or advertisements that are not correlated to the nature of the relation established between the persons. Therefore, it became important to be able to sort out our relationships. Thus, based on the type of links that connect us, we can decide if this last is spammed and should be deleted. Thereby, we propose in this paper a belief approach in order to detect the spammed links. Our method consists on modelling the belief that a link is perceived as spammed by taking into account the prior information of the nodes, the links and the messages that pass through them. To evaluate our method, we first add some noise to the messages, then to both links and messages in order to distinguish the spammed links in the network. Second, we select randomly spammed links of the network and observe if our model is able to detect them. The results of the proposed approach are compared with those of the baseline and to the k-nn algorithm. The experiments indicate the efficiency of the proposed model.

1 INTRODUCTION

Currently, a lot of researches focus on the analysis of social networks. Some authors are interested in the problem of predicting links (Al Hasan and Zaki, 2011), trying to predict likelihood of a future association between the nodes in the current state of the graph. Other works focus on the spammers detection in social networks such as (Zheng et al., 2015). They investigate and analyse the behaviour of the spammers. However, these researches do not deal with the case where a link can be spammed.

Sometimes, the actors of a social network can receive messages which are uncorrelated with the type of link that connects them. From there, it became imperative to be able to sort out the types of relationships in the social network and therefore, allow each actor to decide if this link is spammed and should be removed.

In this context, we propose a new approach which, from the information on the nodes, links and messages, makes it possible to determine if a link is spammed and should be deleted. As we are interested in the observation of interactions in credible social network, all information about nodes, links and messages are related. Indeed, the information on the nodes permits to know the type of link which connects them. From there, the type of message transiting on it is defined.

Currently, we have several information transiting in social networks. However, most of the time, this information may be imperfect, imprecise, uncertain, vague or even incomplete. It becomes essential to use formalism to model these imperfections. Historically, the formalism of probability theory is the most commonly used. Nevertheless, it does not allow the modelling of ignorance. Indeed, in the absence of information, this formalism associates the same probability with each event. In addition, due to the additivity axiom, the probability of an event implies a value on the probability of its complementary.

The limitations of this formalism was a motivation for the development of new theories of uncertainty such as the theory of possibilities (Zadeh, 1999) and the theory of belief functions (Dempster, 1967) which impose no relation between an event and its complementary and it allow to easily model ignorance. The theory of belief functions can be considered more general than that of probabilities or possibilities since we find these as particular cases.

From there, our approach is based on the theory of belief functions in order to model the uncertainty and imprecision due to the different sources of information (links, nodes and messages) and combine

network information. We compare the proposed approach with a probabilistic method and the k-nn algorithm.

This paper is structured as follows. In section 2, we recall some basic concepts related to this work. We propose in section 3 our belief approach to detect spammed links. Finally, section 4 is devoted to the experiments and section 5 concludes the paper.

2 BACKGROUND

In this section, we recall some related works and some basis of the theory of belief functions.

2.1 Related works

Several works have focused on the problem of prediction of links or the detection of spammers in social networks. The authors of (Al Hasan and Zaki, 2011) present a survey of some representative link prediction methods by categorising them by the type of the models: the traditional models which extract a set of features to train a binary classification mode. The second type of methods are the probabilistic approaches which model the joint-probability among the entities in a network by Bayesian graphical models. Finally, the linear algebraic approach which computes the similarity between the nodes in a network by rank-reduced similarity matrices.

The authors of (Liu et al., 2015) introduced an unsupervised link prediction method, the link sample feature representation method and the DBN-based link prediction method for signed social networks. As future works, the authors intend to find other approaches for link prediction in SSNs (Signed Social Networks). In addition, they try to ameliorate the performance of their method and to extract more features.

Regarding the problem of detecting spammers in social networks, (Zheng et al., 2015) adopt the spammers feature to detect spammers and test the result over Sina Weibo. In addition, they study a set of most important features related to message content and user behaviour in order to apply them on the SVM based classification algorithm for spammer detection. Although the proposed approach could achieve precise classification result, it takes over an hour in a process for model training. Furthermore, in the era of big data with huge data volume and convenient access, feature extraction mechanism in the proposed model might be low adaptive and take a lot of time.

The authors in (Martinez-Romo and Araujo, 2013) introduced a method based on the detection of spam tweets in isolation and without previous information of the user and the application of a statistical analysis of language to detect spam in trending topics. The authors present an approach to detect spam tweets in real time using language as the primary tool.

Although the work presented is interesting, the analysed dataset is limited and may still contain some bias. In addition, the number of spam tweets is a lower bound of the real number. As a future work, the authors intend to select the most appropriate features for use in a detection system in real time.

In (Washha et al., 2016), the authors present an approach for detecting spammers on Twitter. In their work, they try first to find to what extent it is possible to increase the robustness of user's and content features used in the literature. Then, the authors were interested to sort out if there is an accessible and unmodifiable property overtime such that it can be leveraged for advancing the available features as well as designing new features.

To sum up, some works in the literature focused on the prediction of the class label of tweet such as in (Martinez-Romo and Araujo, 2013). Other researches (Washha et al., 2016; Zheng et al., 2015) were interested on analysing the user's profile to predict whether the user is a spammer or not.

All these works are interesting either in detecting spammers or predicting links in social networks. However, the researches interested in the problem of predicting links, focus only on how to add links to the network when an entity disappears. As for researches on the problem of spammer detection, they focus solely on the analysis of node behaviour or the content of the tweets.

Therefore, we propose in this paper a method that allows the detection of spammed links in a social network. Contrary to the works of the literature mentioned above, we suppose that the messages exchanged on the network are correct and that the nature of the links can be modified according to the type of the messages which transit on it. In addition, in this work, we take into account the imperfections of the informations in the social network.

We present in the following few concepts from the theory of belief functions used in this work.

2.2 Belief Function Theory

The theory of belief functions allows explicitly the uncertainty and imprecision of knowledge using mathematical tools (Shafer, 1976; Dempster, 1967). In fact, it is a suitable theory for the representation and management of imperfect knowledge. It allows to handle uncertainty and imprecision found in data,

fuse information and make decisions.

Let Ω be a finite and exhaustive set whose elements are mutually exclusive, Ω is called a frame of discernment. A mass function is a mapping

$$m^{\Omega} : 2^{\Omega} \rightarrow [0,1]$$

such that

$$\sum_{X \in 2^{\Omega}} m^{\Omega}(X) = 1 \text{ and } m^{\Omega}(\emptyset) = 0 \tag{1}$$

The mass $m^{\Omega}(X)$ expresses the amount of belief that is allocated to the subset X. We call X a focal element if $m^{\Omega}(X) > 0$.

A simple support function is a mass function which has only one focal element other than the frame of discernment Ω. If $m^{\Omega}(A) = a$ and $m^{\Omega}(\Omega) = 1 - a$, with $a \in [0,1]$ then the source has uncertain and imprecise knowledge. The source believes partially in A, but nothing more and A can be imprecise.

We consider the normalised conjunctive rule called the Dempster rule (Shafer, 2016), given for two mass functions m_1^{Ω} and m_2^{Ω} for all $X \in 2^{\Omega}$, $X \neq \emptyset$ by:

$$m_{\oplus}^{\Omega}(X) = \frac{1}{1-k} \sum_{A \cap B = X} m_1^{\Omega}(A).m_2^{\Omega}(B) \tag{2}$$

where $k = \sum_{A \cap B = \emptyset} m_1^{\Omega}(A).m_2^{\Omega}(B)$ is the global conflict of the combination. This rule is adapted when the combined mass functions are reliable and independent.

To focus on the type of relationship between two different frames of discernment Ω and Θ, we may use the multi-valued mapping introduced by Hyun Lee (Lee, 2011):

$$m_{\Gamma}^{\Theta}(B_j) = \sum_{\Gamma(e_i) = B_j} m^{\Omega}(e_i) \tag{3}$$

with $e_i \subseteq \Omega$ and $B_j \subseteq \Theta$. Therefore the function Γ is defined as follow $\Gamma : \Omega \rightarrow 2^{\Theta}$.

The vacuous extension, being a particular case of multi-valued mapping has the objective to transfer the mass functions defined on two different frames of discernment towards an extended frame of discernment $\Omega \times \Theta$. For that purpose, we apply the operation of vacuous extension defined by:

$$m^{\Omega \uparrow \Omega \times \Theta}(B) = \begin{cases} m^{\Omega}(A) & \text{if } B = A \times \Theta \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In order to make decision, we use the pignistic probability introduced by:

$$BetP(X) = \sum_{Y \in 2^{\Omega}, Y \neq \emptyset} \frac{\mid X \cap Y \mid}{\mid Y \mid} \frac{m(Y)}{1 - m(\emptyset)} \tag{5}$$

The decision is made according to the maximum of pignistic probabilities.

3 PROPOSED APPROACH

In social network, we can find the case of a node that sends certain number of messages that are not compatible with a small portion of links connecting it to the other nodes of the network. Therefore, it would be more appropriate to delete the spammed links and keep the node in the network instead of deleting it.

In this work, we consider a spammed link any link whose initial class has been modified because of the incompatibility of the messages that pass through it in all the iterations. In one iteration, the mass function of the link is updated and it is the input of the next iteration.

With the intention of detecting spammed links, the proposed algorithm takes into account the information on the links and the messages. It proceeds by combining the extended mass functions on the product space. Then, it transfer the obtained mass function to the frame of discernment of the links. After that, the pignistic probability is used to take decision on the obtained type of link. This last is then compared with its initial class. Finally, the modified links types are compared in k iterations. If we obtain the same modified link, then it is considered as spammed. Otherwise, it is considered as outlier.

The proposed approach will be detailed in what follow.

In order to model our idea, we use a belief graph $G = \{V^b; E^b\}$ with: V^b a set of nodes and E^b a set of edges.

In this paper, we consider three frames of discernment for nodes, links and messages:

- $\Omega_N = \{\omega_{n_1}, \ldots, \omega_{n_N}\}$ for the set of nodes.
- $\Omega_L = \{\omega_{l_1}, \ldots, \omega_{l_L}\}$ for the set of links.
- $\Omega_M = \{\omega_{m_1}, \ldots, \omega_{m_M}\}$ for the set of messages.

In addition, we consider a network with N communities. Each community has a specific type that has been defined according to the type of links that make it up such as a "professional", "friendly" or "familial" community .

We present in the following our approach to detect spammed links.

In order to integrate the belief on the links and on the messages, we first make a vacuous extension on $\Omega_L \times \Omega_M$ for each mass of the message of M^b and for each mass of the edge of E^b. Therefore, we obtain on each message M_i^b a mass: $m_i^{\Omega_L \times \Omega_M}$ and on each edge $E_{ij} = (V_i^b, V_j^b)$ between the nodes V_i^b and V_j^b a mass: $m_{ij}^{\Omega_L \times \Omega_M}$.

Then, we combine the extended mass functions using the combination rule of Dempster:

$$m^{\Omega_L \times \Omega_M} = m_{E_{ij}}^{\Omega_L \uparrow \Omega_L \times \Omega_M} \oplus m_{M_i}^{\Omega_M \uparrow \Omega_L \times \Omega_M} \tag{6}$$

We use the multi-valued operation to transfer the combined mass functions on $\Omega_L \times \Omega_M$ to Ω_L. In fact, a multi-valued mapping Γ describes a mapping function:

$$\Gamma : \Omega_L \times \Omega_M \rightarrow \Omega_L \tag{7}$$

We can calculate these equations by using the formula:

$$\Gamma : m_{\Gamma}^{\Omega_L}(B_j) = \sum_{\Gamma(e_i)=B_j} m^{\Omega_L \times \Omega_M}(e_i) \tag{8}$$

with $e_i \in \Omega_L \times \Omega_M$ and $B_j \subseteq \Omega_L$.

Thereafter, we use the pignistic probability in order to make a decision on the obtained type of links. This operation allows to make a comparison with the initial classes of links.

Since our algorithm is iterative, we decide that a link is spammed and must be removed if its class changes at all iterations.

Figure 1 summarises the process steps explained before.

4 EXPERIMENTS

In this section, we present the results obtained after applying our algorithm.

In this work, we use the LFR benchmark (Lancichinetti et al., 2008) which is an algorithm that generates artificial networks that simulate real-world networks.

In these experiments, we use 4 LFR network composed of: 99 nodes with 468 links, 200 nodes with 818 links, 300 nodes with 1227 links and 400 nodes with 1864 links. All the networks have 3 communities. In addition, we consider 3 frames of discernment: $\Omega_N = \{C_1, C_2, C_3\}$, $\Omega_L = \{Friendly, Family, Professional\}$ and $\Omega_M = \{PNC, PC, INC, IC\}$ with *PNC* for Personal Not Commercial, *PC* for Personal Commercial, *INC* for Impersonal Not Commercial and *IC* for Impersonal Commercial.

In this experiment, we consider LFR networks with three communities. We assume that the first community is of type "friendly", the second of type "family" and the third is of type "professional". The type of community is defined from the types of links that make up the majority.

We start by generating the mass functions on nodes and links according to the structure of the network. For each node and each link, we generate 2 focal elements, one on the type of node/link and the second on Ω_N/Ω_L by placing the largest value on the node/link type.

Γ	Friendly	Family	Professional
PNC	$\times$	$\times$	
PC	$\times$	$\times$	
INC			$\times$
IC			$\times$

Table 1: Definition of function Γ given the correspondences between $\Omega_L \times \Omega_M$ and Ω_L

Then, we generate the mass functions on the messages depending on the link type. For each message which transits on the network, we generate 2 focal elements, one on the corresponding type of the message and the second on Ω_M.

Unlike the nodes and links of the network, we generate new mass functions on the messages at each iteration.

We use the passage function Γ defined in table 1 to transfer the mass functions from $\Omega_L \times \Omega_M$ to Ω_L.

In order to validate our approach, we performed two types of experimentations:

- The first type: adding noise on the messages only, then adding noise on the messages in addition of the links.
- The second: pre-selection of a number of spammed links and see if the proposed approach detects them.

In this work, we consider a noisy element (*i.e.* a link or a message) as an element whose mass function or probability has been modified and generated randomly. For the first part of the experiment, we consider different level of noise as follows:

- Case of noisy messages only: 20%, 40%, 50% and 70% of messages from each community were noisy.
- Case of noisy messages and noisy links
  - 20% noisy messages and 20% noisy links.
  - 40% noisy messages and 40% noisy links.
  - 50% noisy messages and 50% noisy links.
  - 70% noisy messages and 70% noisy links.

4.1 Baseline

In order to show the efficiency of our method, we have performed an algorithm that uses the same principle in a probabilistic version.

The probabilistic method consists of projecting the probabilities of links and messages on the Cartesian frame. Then they are combined using the average. This makes it possible to know the type of the link according to the messages which transit on it.

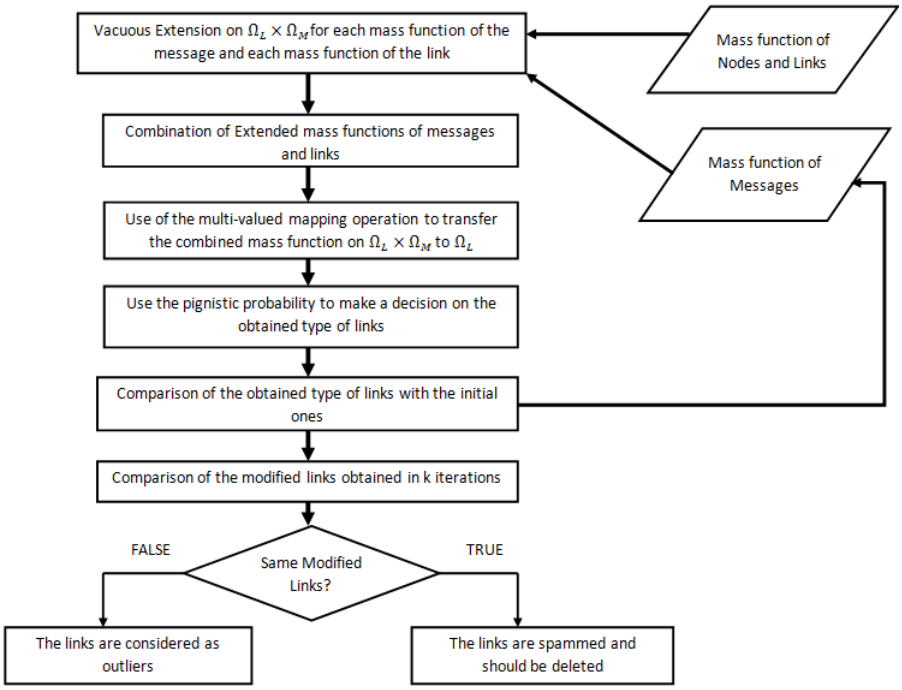

Figure 1: Process of the belief approach.

4.1.1 Extension of probabilities in the Cartesian product

Let the frames of links and messages in a general case: $\Omega_L = \{\omega_{l_1}, \omega_{l_2}, ..., \omega_{l_L}\}$ and $\Omega_M = \{\omega_{m_1}, \omega_{m_2}, ..., \omega_{m_M}\}$.

The Cartesian frame is given by:
$\Omega_L \times \Omega_M = \{(\omega_{l_1}, \omega_{m_1}), (\omega_{l_1}, \omega_{m_2}), ..., (\omega_{l_L}, \omega_{m_M})\}$
Let 2 vectors of probabilities $P_L = (P_{\omega_{l_1}}, P_{\omega_{l_2}}, ..., P_{\omega_{l_L}})$ and $P_M = (P_{\omega_{m_1}}, P_{\omega_{m_2}}, ..., P_{\omega_{m_M}})$.

Given that the frames of the links and messages are independent, we need to project both probability vectors on the Cartesian frame $\Omega_L \times \Omega_M$ in order to combine them.

The fact that the theory of probabilities cannot model ignorance forces us to use an equi-probability when moving from one frame of discernment of links or messages to the Cartesian frame.

Hence for a given probability: $P_L = (\omega_{l_i}, i = 1, ..., L$, we consider the equi-probability on Ω_M to model the ignorance. The result is affected to each pair of Cartesian frame containing ω_{l_i}. For example:

$$P_L^{\Omega_L \times \Omega_M}(\omega_{l_1}, \omega_{m_1}) = \frac{P_{\omega_{l_1}}}{|\Omega_M|}, \ldots, P^{\Omega_L \times \Omega_M}(\omega_{l_1}, \omega_{m_M}) = \frac{P_{\omega_{l_1}}}{|\Omega_M|},$$
$$P_L^{\Omega_L \times \Omega_M}(\omega_{l_2}, \omega_{m_1}) = \frac{P_{\omega_{l_2}}}{|\Omega_M|}, \ldots, P^{\Omega_L \times \Omega_M}(\omega_{l_2}, \omega_{m_M}) = \frac{P_{\omega_{l_2}}}{|\Omega_M|},$$
$$\ldots$$

By the same process, in order to consider the probability $P_M = (\omega_{m_j}, j = 1, ..., M)$ in the Cartesian space $\Omega_L \times \Omega_M$, we consider the equi-probability on Ω_L to model the ignorance. For example:

$$P_M^{\Omega_L \times \Omega_M}(\omega_{l_1}, \omega_{m_1}) = \frac{P_{\omega_{m_1}}}{|\Omega_L|}, \ldots, P^{\Omega_1 \times \Omega_M}(\omega_{l_L}, \omega_{m_1}) = \frac{P_{\omega_{m_1}}}{|\Omega_L|},$$
$$P_M^{\Omega_L \times \Omega_M}(\omega_{l_1}, \omega_{m_2}) = \frac{P_{\omega_{m_2}}}{|\Omega_L|}, \ldots, P^{\Omega_2 \times \Omega_M}(\omega_{l_L}, \omega_{m_2}) = \frac{P_{\omega_{m_2}}}{|\Omega_L|},$$
$$\ldots$$

4.1.2 Calculation of the average of the probabilities

Once the probabilities of the links and messages are projected on the Cartesian frame, we proceed then to the combination of both vectors of probabilities using the average. For example: In this work, we chose to use the average because it has a compromise behaviour. Indeed, if the data contain estimation errors, the calculation of the average makes it possible to reduce this rate of error. For example:

$$\frac{P_L^{\Omega_L \times \Omega_M}(\omega_{l_1}, \omega_{m_1}) + P_M^{\Omega_L \times \Omega_M}(\omega_{l_1}, \omega_{m_1})}{2},$$
$$\frac{P_L^{\Omega_L \times \Omega_M}(\omega_{l_2}, \omega_{m_2}) + P_M^{\Omega_L \times \Omega_M}(\omega_{l_2}, \omega_{m_2})}{2},$$
$$\ldots$$

4.1.3 Projection of obtained averages on the frame of links

In order to return to the frame of the links, we proceed to sum the average probabilities of the hypothe-

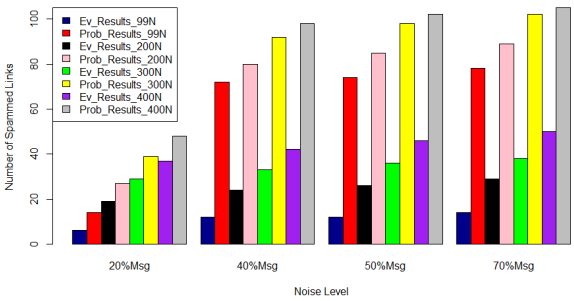

Figure 2: Spammed Links after 10 iterations: case of noisy messages only.

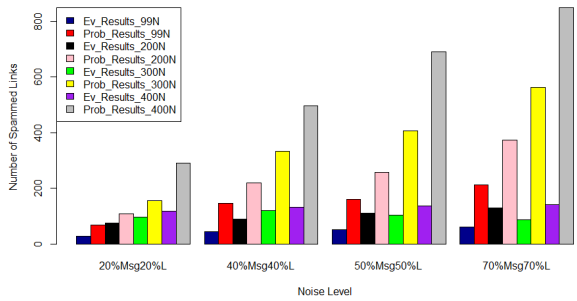

Figure 3: Spammed links after 10 iterations: case of noisy messages and links.

ses that are related to each type of link $(\omega_{l_i}, \omega_{m_j})$, $i = 1, ..., L; j = 1, ..., M$.

4.1.4 Decision making

From each probability vector relative to each link, we determine the current type of the given link $max(\omega_{l_i}), i = 1, ..., L$. Hence, we compare the obtained type with the initial one and decide if the link is spammed or not.

4.2 Case of noisy messages only

In this section, we present the results obtained after adding 20%, 40%, 50% and 70% of noisy messages in each community. The histogram given on Figure 2 shows the number of spammed links that appeared after 10 iterations.

We notice that the more the percentage of the noisy messages increases the more the number of spammed links increases likewise. We also note that in the case of the baseline a larger number of links would be removed compared to the belief approach. This could cause disconnection of the network.

4.3 Case of noisy messages and noisy links

In this section, we present the results after adding noisy links and noisy messages.

The histogram given in Figure 3 shows the number of spammed links that appeared after 10 iterations while varying noise. We note that the baseline begets the removal of a large number of network links. As a result, the network is no longer connected. For example, in the case of 70% noisy messages and 70% noisy links, it detects 213 links which represents about 45.5% of the total links of the network composed of 99 nodes.

4.4 Detection of Spammed Links

In this section, we present the obtained accuracy results after 10 iterations. The goal of this experiment is to test if our model manages to detect the known spammed links. The generated mass functions on the messages are not compatible with the spammed links classes. We consider a LFR network composed of 99 nodes and 10 spammed links.

We will compare the obtained results given by the proposed approach, the baseline and the k-nn algorithm.

The k-nearest neighbour (Altman, 1992) is a supervised learning method. Its principle is as follows: An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours.

It should be noted that in Figures 4, 5 and 6, the accuracy values given by the k-nn oscillate between 0.6 and 0.69. This is because the k-nn requires learning data in contrary to the proposed approach and the baseline. In the following, we present the results of 3 cases:

Generation of 10 **messages of type** $PNCUPC$ The spammed links are of type "professional". Hence, we generate 10 incompatible messages of type "PNC U PC". The curves in figure 4 show that for both evidential and probabilistic approaches, only few spammed links were detected at the first iteration. However, the evidential accuracy is higher than the probabilistic one. For the case of the k-nn algorithm, we notice that it has better accuracy results at the first iterations. Nevertheless, at the tenth iteration, we notice that the evidential accuracy become equal to 79%. So, we can conclude that our model is able to detect correctly more spammed links than the baseline and the k-nn algorithm.

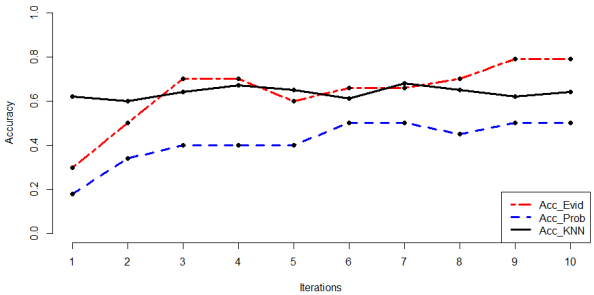

Figure 4: Accuracy Results: Case of *PNCUPC*.

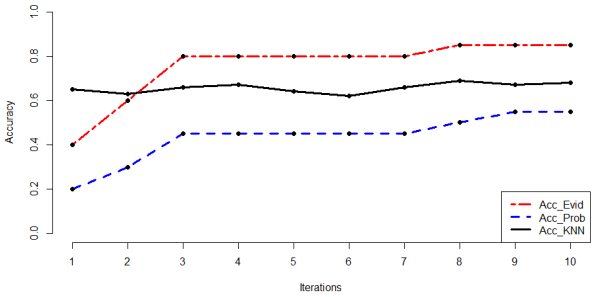

Figure 5: Accuracy Results: Case of *PNC*, *PC*, and *PNCUPC*.

Generation of 10 **messages of type** *PNC*, *PC* **and** *PNCUPC* We generate: 3 messages of type *PNC*, 3 of type *PC* 4 of type *PNCUPC*.

In Figure 5 we can note a clear improvement of detection of spammed links at the tenth iteration. Indeed, the evidential accuracy results given by the proposed approach is equal to 85%.

Generation of 10 **messages of type** *PNCUPC* **and random** We generate: 6 random messages and 4 messages of type *PNCUPC*. We specify that in the case of random message, the focal element can be everywhere except on the empty set in the case of the proposed model.

Figure 6 shows that even when we have a portion of random messages generated on spammed links, our model always gives the best results of accuracy at the tenth iteration.

4.4.1 Evaluation of the algorithm in terms of precision and recall

In this section, we will present the obtained precision and recall results of the proposed approach, the baseline and the *k*-nn algorithm in the case of an LFR network composed of 200 and 400 nodes.

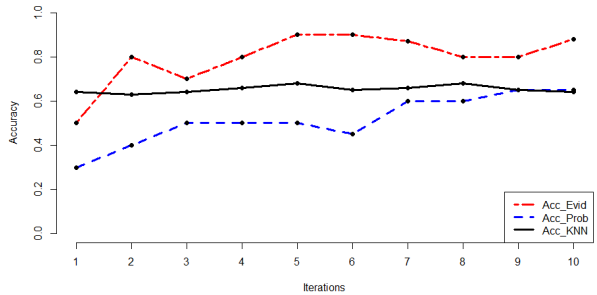

Figure 6: Accuracy Results: Case of random and *PNCUPC* messages.

Method	Precision		Recall	
	It-1	It-10	It-1	It-10
Belief-App	0.36	0.85	0.366	0.87
Baseline	0.23	0.65	0.25	0.67
k-nn	0.55	0.6	0.39	0.4

Table 2: Precision/Recall Results at 1st and 10th iterations.

We remind that precision is the fraction of relevant instances among the retrieved instances while recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances.

Case of LFR network 200 **Nodes** We will start by spamming 20 links of type "professional", 20 links of type "friendly" and 20 links of type "family".

For each type of links, 20 incompatible message were generated:

- For the case of the "professional" link, we generate messages of type "PNC", "PC" and "PNC U PC".
- For the case of the "friendly" and "family" links, we generate messages of type "IC", "INC" and "IC U INC".

Table 2 shows a comparison of the obtained results in terms of precision and recall measures in the case of the proposed approach, the baseline and the *k*-nn. We represent the obtained values at the first and tenth iteration. We note that the results given by the *k*-nn at the first and tenth iterations are close. This is due to the fact that this algorithm requires learning data unlike the evidential and probabilistic methods. Therefore, the methods do not compare the same thing. We notice also that our algorithm gives better results than the baseline and the *k*-nn algorithm.

Case of LFR network 400 **nodes** In this experiment, 600 links were spammed. We will present the obtained results at the first and tenth iteration.

Method	Precision		Recall	
	It-1	It-10	It-1	It-10
Belief-App	0.38	0.8	0.4	0.82
Baseline	0.27	0.6	0.3	0.63
k-nn	0.5	0.51	0.41	0.43

Table 3: Precision/recall results at the 1st and 10th iteration.

Table 3 shows that the proposed approach gives better results in terms of precision and recall compared to the baseline and the k-nn algorithm. We remind that the closeness of the results given by the k-nn at the first and tenth is due to the fact that this algorithm requires learning data unlike the evidential and probabilistic methods.

5 CONCLUSION

The majority of researches in the literature about the evolution in time of a social network focused more on the prediction of entities than the removal of the latter. In this work, we propose a belief approach that detects spammed links in a social network. This work will allow everyone connected to sort out the types of its relationships in the social network and decide which links is spammed and should be deleted.

Throughout this work, we first recalled some related works of the literature as well as some basic notions of the theory of belief functions. Then, we presented our method which consists of detecting spammed links using the information of the nodes, links and messages. In order to test our approach, we performed two types of illustrations: first, we added noise on the messages only, and then we added noise on both messages and links. Second, we selected randomly spammed links and observed if our model manages to detect them.

Experiments have shown that the number of spammed links increases with the noise level. In addition, the results showed that the belief approach is better than the probabilistic one since the latter delete many links of the network. Furthermore, the accuracy, precision and recall results prove that our model is able to detect the majority of spammed links and gives better results than the considered baseline and the k-nn algorithm.

As future work, we will elaborate a strategy to deal with the outliers. Indeed, we will fix a threshold that represents the minimum number of occurrences for a link to be considered spammed. We remind that an outlier is a link that its initial class can be modified but not in all iterations.

Second, we intend to test our proposed algorithm on large and real social networks. To do so, we will associate a simple mass function to each node, link and message of the network based on the community structure. In terms of scaling up, there are several strategies that can reduce complexity such as representing only the focal elements or grouping them together if their values are negligible (Martin, 2009). In addition, the combination rule proposed by (Zhou et al., 2018) can be used to combine mass functions from a large number of sources.

REFERENCES

Al Hasan, M. and Zaki, M. J. (2011). A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, pages 325–339.

Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110.

Lee, H. (2011). Context reasoning under uncertainty based on evidential fusion networks in home-based care.

Liu, F., Liu, B., Sun, C., Liu, M., and Wang, X. (2015). Deep belief network-based approaches for link prediction in signed social networks. *Entropy*, 17(4):2140–2169.

Martin, A. (2009). Implementing general belief function framework with a practical codification for low complexity. *Advances and applications of DSmT for Information Fusion-Collected works*, 3:217–273.

Martinez-Romo, J. and Araujo, L. (2013). Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, 40(8):2992–3000.

Shafer, G. (1976). *A mathematical theory of evidence*, volume 1. Princeton university press Princeton.

Shafer, G. (2016). Dempster's rule of combination. *International Journal of Approximate Reasoning*, 79:26–40.

Washha, M., Qaroush, A., and Sedes, F. (2016). Leveraging time for spammers detection on twitter. In *Proceedings of the 8th International Conference on Management of Digital EcoSystems*, pages 109–116. ACM.

Zadeh, L. A. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 100(1):9–34.

Zheng, X., Zeng, Z., Chen, Z., Yu, Y., and Rong, C. (2015). Detecting spammers on social networks. *Neurocomputing*, 159:27–34.

Zhou, K., Martin, A., and Pan, Q. (2018). A belief combination rule for a large number of sources. *Journal of Advances in Information Fusion*, 13(2).