Detection of Multiple Identity Manipulation in Collaborative Projects

Zaher Yamak
INSA Rouen, LITIS Lab
685 Avenue de l'Université
76801 Saint Etienne du Rouvray cedex, France
zaher.yamak@insa-rouen.fr

Julien Saunier
INSA Rouen, LITIS Lab
685 Avenue de l'Université
76801 Saint Etienne du Rouvray cedex, France
julien.saunier@insa-rouen.fr

Laurent Vercouter
INSA Rouen, LITIS Lab
685 Avenue de l'Université
76801 Saint Etienne du Rouvray cedex, France
laurent.vercouter@insa-rouen.fr

ABSTRACT

Various techniques are used to manipulate users in OSN environments such as social spam, identity theft, spear phishing and Sybil attacks... In this article, we are interested in analyzing the behavior of multiple fake accounts that try to bypass the OSN regulation. In the context of social media manipulation detection, we focus on the special case of multiple Identity accounts (Sockpuppet) created on English Wikipedia (EnWiki). We set up a complete methodology spanning from the data extraction from EnWiki to the training and testing of our selected data using several machine learning algorithms.

In our methodology we propose a set of features that grows on previous literature to use in automatic data analysis in order to detect the Sockpuppets accounts created on EnWiki. We apply them on a database of 10.000 user accounts. The results compare several machine learning algorithms to show that our new features and training data enable to detect 99% of fake accounts, improving previous results from the literature.

Categories and Subject Descriptors

Security and privacy [**Intrusion/anomaly detection and malware mitigation**]: Social engineering attacks

General Terms

Security and privacy

1. INTRODUCTION

Since 2004, the social web has become a dominant force on the Internet. As of 2014, 52% of adults online now use two or more social media sites, a significant increase from 2013, when it stood at 42% of Internet users [5].Nowadays, traditional media are no longer the only source of news because major news stories routinely break on Twitter before traditional news media [9]. Online Social networks (OSN) have therefore become a preferential medium for diffusion of information (e.g. mass birthday celebration) or opinions (e.g. on a product or on a public person) and to promote ideas (i.e. activism, call for vandalism or riots). However, as the importance of social media grows, the number of manipulators increases.

A manipulator is a person who uses the rules of life in society, in this case those of the social media, to obtain advantages and personal benefits or exercise control over one or more users. Manipulation on social media can be made through verbal communication (e.g. video, audio, or text) and/or non-verbal behavior (eg: number of friend requests per hour, or the size/quality of added text...) using many technics like spams or Sybil attacks [4].

In this article, we propose to detect a special kind of attack, multiples accounts, that is the source of several different types of manipulation (Sybil, information manipulation, social spams...). In order to do this, we have selected the social media Wikipedia that allows the public extraction of a major part of its data.

Wikipedia is a collaborative project where anyone can edit most of its articles either with or without creating an account on it. We have chosen English Wikipedia (EnWiki) as our study focus because it has the highest number of collaborators and manipulators among the collaborative projects OSNs.

The objective of Wikipedia is to be a crowd-sourced encyclopedia, because it has a volunteer communities that maintain it. The editing of its contents is open to the public, thus allowing manipulation attempts. The sets of pages on Wikipedia are composed of pages with information or discussions about Wikipedia and they are called namespaces.

In the context of Wikipedia, the improper use of multiple accounts is called *Sockpuppetry*. The Sockpuppet is the fake account used for purposes of deception on the collaborative project sites. Any user that considers an other account as being a Sockpuppet can open an investigation page about this malicious account by providing a clear evidence to Wikipedia's experienced administrators. These administrators then try to detect the Sockpuppet manually, by studying the Sockpuppet behavior on Wikipedia or by detecting the similarity in writing style.In many cases, the administrator demands from 'checkuser', a more privileged set of users who have access on the IP-address of all accounts, to intervene by checking and comparing the IP-address with

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890586.

other accounts. Then, a choice is made according to similar user location and similar interventions styles to ban or not the account.

The objective of this article is to detect automatically the Sockpuppet accounts on Wikipedia using non-verbal indicators.

The main contributions of this paper are (1) a methodology to extract and analyze large amounts of data in order to automatically detect OSN deception techniques and (2) its application to the case of Sockpuppets in Wikipedia. The main steps are the following : (1) We crawl Data from Wikipedia for accounts (blocked or not) and we log all their activities. (2) We filter Data to select 10 000 active and Sockpuppet accounts. (3) We create a set of non-verbal behavior features in order to detect the Sockpuppet accounts on Wikipedia. (4) We calculate the values of the proposed features. (5) We evaluate several supervised learning algorithms that use these features and we compare our results to other researchers that used verbal and non-verbal communication features.

The rest of the paper is organized as follows : In section 2, we present collaborative projects and the current technique to detect sockpuppets manually. In section 3, we present the state of the art in the field of detection of deception and manipulation in OSN. Then, in section 4, we describe our proposed methodology using real data extraction, the feature selection, and our experiment metrics. In section 5, we present the results of our proposed method and discuss these results in section 6. Finally, in section 7, we conclude and show our future directions.

2. RELATED WORK

Fake accounts are used to increase the visibility of niche content, forum posts, and fan pages by manipulating votes or view counts [10]. Others are created for malicious behavior, such as spamming, click-fraud, malware distribution, and identity fraud [11].

Many solutions were created in order to detect the malicious behavior on social media. Two main approaches are used: verbal communication analysis and non-verbal behavior analysis. In this section we review past studies close to our study focus.

2.1 Verbal Communication

Gao *et al.* [7] propose a solution to detect spammers on Facebook. They start by filtering the crawled data from Facebook to keep only the posts that contain URLs, their hypothesis being that every spammer will try to redirect the social media user to an outside fake site. Then, they link the similar posts by checking if they share the same destination or the same text content. Finally, they cluster 1,402,028 linked posts by checking if the link is guiding to a fake website or not, in order to detect the malicious users and posts. The authors obtain good results with this method, which detect 93.9% of malicious wall posts.

Solorio *et al.* [13] use natural language processing techniques to detect on Wikipedia the users who maintain multiple accounts based on their verbal behavior. Textual features are used such as alphabet count, number of tokens, emoticons count or the use of words without vowels (e.g. try,cry,...). These features are tested on all revisions made by the users on pages throughout Wikipedia. A Support Vector Machine (SVM) algorithm has shown 68.83% overall accuracy using an experimental dataset of 77 cases of legitimates users and Sockpuppets.

In fact, the automatic detection of verbal behavior is not always accurate because the manipulators can expect the detection method and thus change their writing method, which makes the auto detection more difficult. This voluntary behavior change is hardest to implement when the automatic detection uses the non-verbal behavior.

2.2 Non-Verbal Behavior

Yang *et al.* [16] integrate a real time detection in the Chinese OSN RenRen for Sybil accounts by studying the non-verbal behavior. They characterized the RenRen accounts according to the following features: Invitation Frequency, Outgoing Requests Accepted, Incoming Requests Accepted and a mutual connectivity of a user's friends measure. They apply a Support Vector Machine (SVM) classifier to their ground truth dataset of 1000 normal users and 1000 Sybils. They randomly partition the original sample into 5 sub-samples, 4 of which are used for training the classifier, and the last used to test the classifier. The results show that the classifier is very accurate, correctly identifying 99 % of both Sybil and non-Sybil accounts.

Closer to our study is Tsikerdekis and Zeadally [15] who propose a method for detecting identity deception through the use of non-verbal user activity in the social media environment. They use publicly available data from Wikipedia and machine learning algorithms. Some variables where representing non-verbal user behavior such as the number of revisions made by a user for a specific time window since their initial registration with the website, the total number of bytes added and total number of bytes removed,... For testing they sample 7,500 cases of Sockpuppets and they include Support Vector Machine (SVM), Random Forest (RF) and Adaptive Boosting (ADA), that shows a 71.3% overall accuracy.

In these studies, we have seen that verbal and non-verbal communications are interesting axis, but that the overall Sockpuppet detection was not sufficient for automatic discovery of such behavior. In the next section, we will describe our methodology to detect the Sockpuppet by selecting some new non-verbal behavior features using real data from En-Wiki.

3. METHODOLOGY

In this study, we work on the user's non-verbal behavior. The global methodology is illustrated in figure 1. There are four main steps:

Data extraction

In order to extract relevant information for the machine learning algorithms, the first step is to crawl available data from EnWiki. The total amount of data contained in EnWiki is about 10 TB uncompressed[1], so that it is not efficient to retrieve and process it. Hence, the starting point is to crawl all blocked users with all their contributions on Wikipedia from February 2004 to April 2015.

Many reasons may cause to block an account, such as spam, vandalism, or threats. In our case, we are interested in studying the Sockpuppet accounts, so we filtered only the accounts blocked for Sockpuppet reason and for infinite

[1]100 GB compressed using 7-Zip

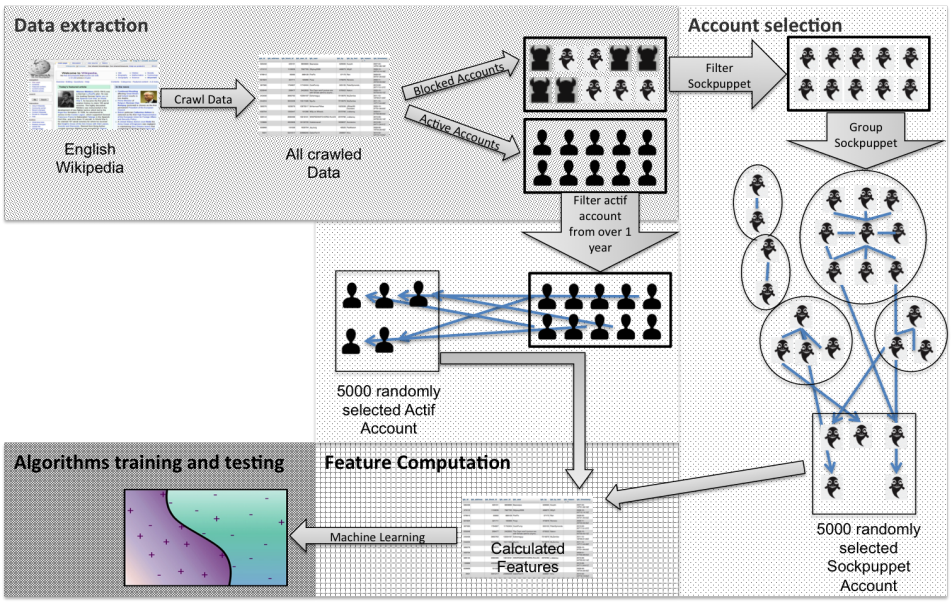

Figure 1: The four main steps in our methodology.

time. The total number of accounts blocked for this reason amounts to 118 414 accounts. When an account is blocked because of Sockpuppetry, the administrators state to which original account the blocked one was linked.

Once these accounts were found, we have extracted all related data in the thirty namespaces. Furthermore, we also retrieved all information about these pages, including the behavior of other users towards the users modifications. This allows for example to automatically detect whether the modifications done by a specific user are kept by other users or reverted.
The final database used in this study thus contained 71 GB of information.

Account selection

The second step is the account selection, both for positive and negative Sockpuppets accounts.
Concerning the Sockpuppets, we first group the 118414 accounts and refer them to their Sockpuppeter, *i.e.* the first account created by the user. We obtained 12088 groups, ranging from 2 to 557 members. Among those groups, we select randomly 5000 Sockpuppet accounts that are in a group with more than 3 Sockpuppets, in order to be sure that we are selecting accounts of users who tried many times to manipulate.

An example of such groups is that of the owner of the account *"YCWebCrawler"*, who has created 4 supplementary accounts (*Fixthemistakes*, *Wordslayer*, *Wikidirt* and *PageOneEditor*) to manipulate on Wikipedia. All these accounts are blocked as Sockpuppet accounts in EnWiki because they were created solely for the deletion of material on the *Dennis L. Montgomery* page. All these accounts use similar language about protecting family members in summaries.

Once the Sockpuppet accounts are selected, we also have to select active accounts to train the algorithm on acceptable behaviors. These are called "Active" accounts because they are not blocked and they can use Wikipedia without any restriction. We select randomly 5000 active accounts, which fulfill two criteria: they have been active for more than one year and they have made at least one contribution on Wikipedia. These criteria were selected in order to have a high probability that these accounts are not Sockpuppet yet undiscovered, and to have sufficient non-verbal behavior.

Feature Computation

We have selected and calculated a set of features, detailed in section 3.1. None of these features are directly available in the extracted data.
Hence, the third step of our method is to pre-process the raw data for the constitution of the vectors of the training and testing data. This is done through php scripts that work on raw pages to calculate all the features values for each account.

The obtained data is thus composed of a set of 11 features for each account.

Algorithms training and testing

The last step of our method is the supervised learning training and testing. We decided to compare our data-extracted features with several machine learning algorithms to prove the accuracy and robustness of our method.

In order to evaluate the efficiency of our model of our proposed method we used the following machine learning algorithms: Support Vector Machine (SVM) [3], Random Forest (RF) [2], Naíve Bayesian (NB) [12], K-Nearest Neighbor (KNN) [1], Bayesian Network (BN) [8], Adaptive Boosting (ADA) [6].

These algorithms were tested using ten-fold cross-validation: we partition the sample of 10.000 data into complementary subsets, performing the analysis on the training subset, and validating the analysis on the validation set. The random partition, training and validation are done ten times.

To quantify the quality of prediction, we have selected a set of metrics described in section 3.2.

3.1 Features Selection

In order to study our data and create our new method, we propose several features to differentiate between a Sockpuppet and active account. These can be divided in three sets: contribution behavior, other users behavior toward these contributions, and account behavior.

Since our objective is to detect the Sockpuppet, whose goal is to manipulate contributions in the encyclopedia, we are interested in the contribution behavior of the users. In particular, we are interested in the number and the areas -in terms of namespaces- of the contributions of each user in Wikipedia because the namespaces are the places where the user is contributing, so they are a good indicator for the behavior of each user. We also select the average of added and removed bytes in each contribution, and the average of the contribution of each user in the same article.

Secondly, we study the frequency of direct cancellation of the work of an account in an article by other users (also called reverts). This is an indirect use of other users expertise that can be retrieved automatically from data.

Finally, we compare the interval between the registration time of each user and their first contribution time.

For each account we calculate in the second step of our method the values of the selected features that capture the behavior of users on Wikipedia. The features and their underlying hypothesis are described below:

The number of user's contributions by namespaces:
We compose the user's contributions according to namespaces into six categories: article, article discussion, user page, user discussion page, project namespace and other (all other namespaces combined under one feature). The rationale is that these are the most important namespaces that help to detect the writing behavior and the interest(s) of Wikipedia's contributors.

The frequency of revert after each contribution in the articles:
We take as hypothesis that most of time the manipulation of a Sockpuppet will be reverted by another user, because each page is managed by many contributors and when they find a malicious contribution they will revert it directly.
In this feature we calculate the frequency of direct revert after each contribution in an article.

Let Na be the number of contributions in article pages and Nr be the number of reverts, the feature Fr is then:

$$Fr \quad = \frac{Nr*100}{Na} \tag{1}$$

For example, a user that contributed 9 changes in two pages, but had 3 out of 9 contributions reverted will have a score of 33%.

Let us note that this feature is not directly available in the data, and the preprocessing had to compare all changes in the history of article pages to detect these reverts.

The average of bytes added and removed from each revision:
Our purpose for these two features is to check the user's behavior of writing in the articles.
The first feature calculates the average of the numbers of bytes of the information added in the articles for all the contributions (revision) of each account, and the second feature calculates the average of numbers of bytes of the information removed in the articles for all the contributions of each account.

$$Og+ \quad = \frac{\sum_{i=1}^{N} Op}{N} \tag{2}$$

Where Op is the number of positive bytes and N is the number of revisions

$$Og- \quad = \frac{\sum_{i=1}^{N} On}{N} \tag{3}$$

Where On is the number of negative bytes and N is the number of revisions.

The hypothesis is that the manipulation goal of the Sockpuppeter can be extracted from its addition/removal behavior. The manipulation goal of a Sockpuppeter may be either to add and publish a particular piece of (mis) information, or to remove (part of) a previous contribution.

The average of contribution in the same article:
We compute the average of contribution in the same article, because we consider that a manipulator is more prone to try to manipulate many times in the same article(s).
In this feature we calculate the number of contributions in the same article for each account, then we calculate the average of these numbers.

$$Ci \quad = \sum Ai \tag{4}$$

Where Ai is the number of contribution in the article i

$$Cg \quad = \frac{\sum_{i=1}^{N} Ci}{N} \tag{5}$$

Where N is the number of articles.

The interval between the user's registration and his first contribution:
In this feature, we calculate the difference (in seconds) between the time of registration and the time of the first contribution in the EnWiki for each account.
The underlying assumption is that a manipulator creates at the beginning of its manipulation attempt many accounts, and then leaves them sleeping to use them separately as backup when an active account is blocked.

Let us note that the last feature and the average of added and removed bytes features were used in Tsikerdekis[15], while the features related to the contributions in namespaces were modified in order to be more precise by composing the contribution in more namespaces. The Features about the frequency of revert after each contribution in the articles and the average of contribution in the same article, are new proposals.

3.2 Algorithms and metrics

In order to evaluate the efficiency of our model, we first show the model precision n and then use the confusion matrix shown in Table 1. This matrix is used to visualize the performance of the different algorithms using the following metrics:

$$TruePositiveRate(TPR) \quad = \frac{TP}{TP+FN} \tag{6}$$

This metric indicates the rate of truly detected sockpuppet.

$$FalsePositiveRate(FPR) \quad = \frac{FP}{FP+TN} \tag{7}$$

	Sockpuppet	Actif Accounts
Predicted Sock	True Positive (TP)	False Positive (FP)
Predicted AA	True Negative (TN)	False Negative (FN)

Table 1: Table representing the confusion matrix used to evaluate the efficiency of our proposed method.

This metric indicates the rate of falsely detected sockpuppet.

$$Precision \quad = \frac{TP}{TP+FP} \tag{8}$$

This metric indicates the fraction of returned cases that are valid sockpuppet cases.

$$F-measure \quad = \frac{2*Precision*TPR}{Precision+TPR} \tag{9}$$

This metric indicates the fraction of the combination of TPR and precision.

$$MCC = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{10}$$

Matthews correlation coefficient (MCC) metric indicates a balanced measure which can be used even if the classes are of very different sizes.

In the previous work, Solorio *et al.* [13] evaluate their proposed model using only Support Vector Machine (SVM), while Tsikerdekis *et al.* [15] used Support Vector Machine [3], Random Forest (RF) [2], and Adaptive Boosting (ADA) [6] in order to evaluate their model.

We summarize our obtained results using the mentioned performance metrics and the confusion matrix in Table 1 in the next section.

4. EXPERIMENT AND RESULTS

We have used the dataset presented in section 4 that comprises 10.000 users, half blocked users and half active users. We used the Weka[2] to evaluate the features. We used a ten-fold cross-validation: To validate our proposed method we split the data randomly to 2/3 for learning and 1/3 for testing and repeated this procedure ten times. The obtained results are shown in the table 2. We then discuss these results in the following section.

4.1 Algorithms comparison

The table 2 compares the precision percentage of the testing phase of our proposed model over different machine learning algorithms. It shows that we obtained the best accuracy using Random Forest(RF) 99.8% and Bayesian Network(BN) 99.6%.

The results of the remaining algorithms are significantly lower, with SVM at 78.1%, KNN and ADA share the same results 63%, and finally the poorest results was the Naïve bayesian(NB) algorithm at 50.1%.
Also this table compares the different values of our proposed model metrics between many machine learning algorithms. It shows that TP Rate, Precision and F-Measure using Random Forest algorithm are 0.998/1, and that the MCC is 0.997. The F-measure using SVM give 0.771, and 0.628 using KNN and ADA.

[2]`http://weka.wikispaces.com`

	TPR	FPR	Precision	F-Measure	MCC
SVM	0.782	0.218	0.848	0.771	0.627
RF	0.998	0.002	0.998	0.998	0.997
NB	0.501	0.498	0.521	0.351	0.011
KNN	0.630	0.370	0.633	0.628	0.264
BN	0.997	0.003	0.997	0.997	0.993
ADA	0.630	0.370	0.633	0.628	0.264

Table 2: Table comparing the performances' results between different machine learning algorithms.

These results show that only the Random forest and Bayesian network algorithms are suitable using our features for automatic suppression of Sockpuppets accounts, although the SVM algorithm also shows decent results.

4.2 Literature Comparison

We present a comparison between the results of our proposed method and three previous methods in the table 3. In this table, we can observe that our proposed method give the best accuracy percentage between all the different methods by arriving to 99,8%. Non-verbal expectancy Violations Detection [15] method arrived to a result of 71,3%. Adaptive SVM Text Attribute Disagreement Algorithm [14] have a 73% accuracy, and finally that Natural Language Processing Similarity Searching [13] method achieved 68,8% as overall accuracy.

A significant difference with the two latter studies is the size of the data sample. While we used 10.000 cases, the two studies from Solorio used only 623 and 77 cases. Indeed, the use of verbal behavior analysis is more computation-heavy than non-verbal behavior, and may not be as easily processed.

Compared to the first study that used 15.000 cases, only the feature selection is significantly different and explains the results variation.

5. DISCUSSION OF RESULTS

The results shown in the previous section are the best between our method and the three previous methods[14, 13, 15]. We found that the Random Forest algorithm has the best accuracy between all others algorithms, and that Bayesian networks is nearly as efficient. The fact that two different algorithms show accuracy results over 99.5% is an indication of the robustness of our features selection. Unsurprisingly, we also found that the lowest precision is in the Bayesian Network algorithm with 52%.

The best method for the detection of Sockpuppet before our contribution was the method made by Tsikerdekis[15]. If we compare our method to this method we found that both methods use only the non-verbal indicators and the overall accuracy is 71% for Tsikerdekis method[15] but 78% for our method using the SVM algorithm with radial basis function (RBF) as kernel. However, the linear SVM does not have the best accuracy, which is normal because the data cannot be detected easily using a linear function.

We evaluate our list of 11 features, mentioned in section 3.1, using correlation attribute evaluation method and we found that the best three features are (i) the frequency of revert after each contribution in the articles; (ii) the average of added bytes; and (iii) the third is the average of contribution in the same article. This validates and explains

	Adaptive SVM Text Attribute Disagreement Algorithm[14]	Natural Language Processing Similarity Searching[13]	Non-verbal expectancy Violations Detection[15]	Our proposed Method
Overall Accuracy	73%	68.8%	71.3%	99.8%
Data Set	623 cases	77 cases	15 000 cases	10 000 cases

Table 3: Table comparing the results between our proposed method and the others previous methods

our results because the first and the third features are the main differences between our method and Tsikerdekis's[15] method.

We can consider that our method is effective in the detection of Sockpuppet because the manipulator is less aware of the non-verbal indicators. Also, he cannot manipulate them as easily, by example the revert frequency after each revision is a very good indicator because the manipulator cannot control this feature to try to hide from detection. The non-trivial data preprocessing may also create a difficulty for the manipulators to be aware that there is a method that detects the time difference between the registration and the first contribution.

Finally, the users that have an objective to manipulate in a precise article will have a difficulty to skip from the detection and to change their non-verbal writing behavior if they want to achieve their goal but it is easier to change their writing method to skip from verbal detection.

6. CONCLUSION AND FUTURE WORK

In this article, we present a method to detect the identity deception over the collaborative project. In order to test and validate our model we propose features for non-verbal behavior using account on EnWiki.

The success of a machine learning algorithm depends of the selection of features that are the inputs to the algorithm. In this way, we have achieved an accuracy over 99,7% with two methods, Random Forest and Bayesian Network, and 78% with Support Vector Machine. The machine learning algorithms was trained and tested using 10 fold cross validation method for 10 000 cases as dataset.

These good results are due to a well selected feature set that help to identify a Sockpuppet from legitimate user. This feature set is automatically calculated from publicly available data.

The Random Forest and Bayesian Network algorithms can be overfitted due to specific artifacts found in the training and testing dataset. We plan to study more closely how these algorithms perform with the use of a third (development) dataset.

Our method gave very good results. In the future, we plan to explore other social media to verify our feature set in other settings, such as forums or twitter. This may lead to difficulties, because behaviors vary among social media types. Furthermore, in social media that use more free-text expressions, we will work on detecting the verbal communication in addition for the non-verbal behavior.

Finally, our work demonstrates that automated detection techniques can be successfully used in EnWiki, so that it could also be used in other collaborative projects like Wikitionary and Wikiversity.

7. REFERENCES

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

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[4] J. R. Douceur. The sybil attack. In *Peer-to-peer Systems*, pages 251–260. Springer, 2002.

[5] M. DUGGAN, N. B. ELLISON, C. LAMPE, A. LENHART, and M. MADDEN. Social media update 2014., January 2015. http://www.pewinternet.org/2015/01/09/social-media-update-2014/.

[6] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[7] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 35–47. ACM, 2010.

[8] D. Heckerman. A tutorial on learning with bayesian networks. In *Innovations in Bayesian Networks*, pages 33–82. Springer, 2008.

[9] M. Ingram. If you think twitter doesn't break news, you're living in a dream world, February 2012. https://gigaom.com/2012/02/29/if-you-think-twitter-doesnt-break-news-youre-living-in-a-dream-world/.

[10] S. Nyberg. Fake accounts in facebook - how to counter it., February 2010. http://ezinearticles.com/?id=3703889.

[11] R. RICHMOND. Stolen facebook accounts for sale., May 2010. http://www.nytimes.com/2010/05/03/technology/internet/03facebook.html.

[12] S. Russell, P. Norvig, and A. Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

[13] T. Solorio, R. Hasan, and M. Mizan. A case study of sockpuppet detection in wikipedia. In *Workshop on Language Analysis in Social Media (LASM) at NAACL HLT*, pages 59–68, 2013.

[14] T. Solorio, R. Hasan, and M. Mizan. Sockpuppet detection in wikipedia: A corpus of real-world deceptive writing for linking identities. *arXiv preprint arXiv:1310.6772*, 2013.

[15] M. Tsikerdekis and S. Zeadally. Multiple account identity deception detection in social media using nonverbal behavior. *Information Forensics and Security, IEEE Transactions on*, 9(8):1311–1321, 2014.

[16] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering social network sybils in the wild. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):2, 2014.