Hierarchical Meta-Path Based Collective Classification of Spammers Abusing Online Social Networks

Student Name: Abhinav Khattar
Roll Number: 2015120

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering

on April 18, 2018

BTP Track: Research Track

BTP Advisor
Dr. Ponnurangam Kumaraguru
Dr. Tanmoy Chakraborty

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **"Hierarchical Meta-Path Based Collective Classification of Spammers Abusing Online Social Networks"** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Engineering* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr. Ponnurangam Kumaraguru** and **Dr. Tanmoy Chakraborty**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

..... **Place & Date: .....**
Abhinav Khattar

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

..... **Place & Date: .....**
Dr. Ponnurangam Kumaraguru

..... **Place & Date: .....**
Dr. Tanmoy Chakraborty

Abstract

We have entered an era where cybercriminals and spammers tend to use Online Social Networks to spread Spam-Campaigns, misinformation and dupe people. These spammers tend to share URLs, phone numbers and other malicious contents. The aim of the project is to detect such spammers who are spreading phone numbers on Online Social Media. In this project, Meta-Path based properties are used along with an Active Learning based feedback strategy to find such spammers sharing phone numbers on Twitter. The analysis is done on 670,251 Twitter users across 3,370 campaigns. Users that have been suspended by Twitter have been used as the ground truth to train the prediction model. It is shown that the usage of the feedback model suggested helps achieve a better results despite the unavailability of a large initial ground truth dataset. This method also removes the the need of a manual annotator for annotating the training dataset as only the data collected by Twitter is used. F1 Score and Area Under ROC curve are used to evaluate the performance of the classifier. The suggested methodology achieves 6.9% higher F1 and 67.3% higher AUC than the best baseline. Efforts have been further made to incorporate the usage of conversation networks into the system to increase the campaigns it encompasses. HMPS is further tested on multiple publicly available datasets to test its validity and generalisability.

Keywords: Social Network Analysis, Spam Users Detection, Meta-Path based classification

Acknowledgments

I would like to acknowledge the role of Dr. Ponnurangam Kumaraguru in helping shape this Project with his constant guidance. I would also like to thank Dr. Tanmoy Chakraborty without whose guidance this project would not have been possible. Finally, I would like to thank Srishti Gupta, PhD student at IIIT-D, for mentoring me throughout the course of the project.

Work Distribution

1-15 August : Chapter 1 : Understanding the basics of the problem and its complexity. Understanding certain basic concepts of Complex Networks.
15 August - 15 September : Chapter 2 : Learning the Dataset and representing in the form of HIN. Literature Review of HIN.
15 September - 15 October : Chapter 3 : Working on HMPS and Active Learning algorithms
15 - 30 October : Chapter 3 : Fine tuning things and preparing final result
1-15 January : Chapter 5: Analysing the campaigns our algo ran on and creating incites
15-30 January: Chapter 5: Creating conversation nets and analysing if they can be used for extending the results
February: Chapter 6: Making a bigger script that performs all tasks required for final classification, that runs automatically and keeps predicting unless convergence
March: Chapter 6: Making the general script run till a fixed time
April: Chapter 7: Analysing multiple datasets HMPS can run on to show that the concept can be generalised

Contents

Chapter 1

Introduction

1.1 Problem Statement

With a constant increase in the number of people using Online Social Media, spammers are leveraging its popularity to spread spam content. The aim of the project is to detect spam users who are sharing their phone numbers on Twitter. Phone numbers are an effective means of spreading spam content on OSNs as by spreading their phone numbers across a social network, spammers save the effort of reaching out to their victims themselves. In the project, a user account is called a spam account if it uses phone number to aggressively promote products, spread pornographic content or mislead people by making false claims and entice them into lotteries, discounts etc.

1.2 Motivation

The problem of finding spammers on Online Social Networks has long been known. A lot of previous research has used data like Landing page information, Web of Trust score and length of URLs shared to find the about the spamicity of an account. Contrary to this, phone numbers provide no such information making these methods to find spammers ineffective. Adding to the complexity of the problem, people instantly fall for such phone based spammers due to the inherent trust that is associated with a phone number. The problem of such phone based spam has recently gained traction along with coverage by a lot of news sources[1].
A famous example of such phone based spam is the TechSupport scam where people are asked to pay exorbitant amounts of money by scammers who pretend to be tech support employees for companies. These scammers not just try to earn money from the victim, but also try to get the access to the victim's computer.

[1] https://www.marketwatch.com/story/heres-how-much-phone-scams-cost-americans-last-year-2017-04-19

Chapter 2

Dataset and Data Representation

2.1 Dataset

The dataset[1] compromises of around 22 million Tweets collected from April-October 2016; collected using a list of 400 keywords like 'SMS', 'call', 'ring', 'WA' etc.
We further divided the dataset into multiple campaigns where a campaign is defined as a group of similar posts shared by a set of users propagating multiple phone numbers. This would mean that multiple phone numbers can be part of the same campaign if the Tweets containing them have similar texts. Similarity was found by aggregating Tweets based on Phone Numbers, extracting unigrams and finally finding the Jaccard's similarity.

2.2 Data Representation

2.2.1 Heterogeneous Information Network

In order to model the relationship between users and the content they share, we can use a Heterogeneous Information Network (HIN). An HIN is different from a regular network in the sense that nodes in an HIN don't tend to be of the same type. For example, a typical graph where Users are connected to each other based on their friendships is a Homogeneous Network as all the nodes are of the same type i.e. user. In a network, if we connect the users to URLs and Phone Numbers they share, we essentially make a user, phone number and a URL a node in the graph. In this case the graph can be called a HIN as the nodes are not of the same type. HINs provide a powerful yet simple mechanism to study similarity between nodes of a graph.
Heterogeneous network is represented as a graph, $G = \{V, E, T\}$ in which each node $v \in V$ and each link $e \in E$ are associated with their mapping functions:

$$\phi(v) : V \rightarrow T_V$$

[1]Dataset and Tweets were collected by Srishti Gupta, PhD @ IIIT-D.

$$\phi(e) : E \rightarrow T_E$$

respectively. $T_V \in T$ and $T_E \in T$ denote the sets of users and edge types.

2.2.2 Meta-paths

Meta-paths are defined as a set of nodes C_1, C_2, ... C_n connected to each other by a sequence of edges e_1, e_2, e_3, ... e_{n-1} [9]. Meta-paths have previously been used to find similarity between 2 nodes and techniques like PathSim [14] and HeteSim [13] have been used for the same. Sun et al [14] also show that finding all meta-paths and then finding the most relevant out of them is an NP-hard problem. So,the major problem that comes in using these techniques is: (i) finding of all meta-paths between 2 nodes (ii) Defining a "relevant" meta-path.
As most of the modern day networks -especially OSNs- are extremely dense and contain a large amount of nodes, finding all meta-paths even upto a restricted depth can turn out to be pretty resource intensive. During the initial phases of the project, meta-paths upto a depth 4 were tried to be found, but owing to the high amounts of time the process was taking, the idea had to be dropped. Finding useful meta-paths is also a problem that has gained a lot of traction and numerous techniques have been suggested to find these as well [9]. In this project, a technique HMPS is put forward. This technique tries to find a Hierarchical Meta-path connecting 2 users in order to find similarity between them. This technique is explained in the methodology.

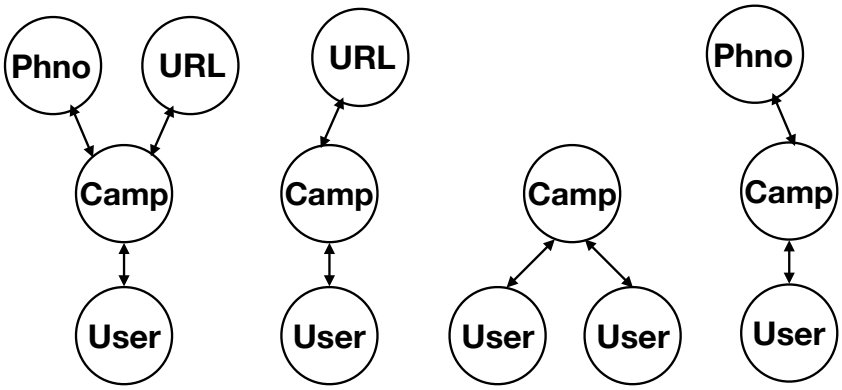

Figure 2.1: Meta-paths in the network

Chapter 3

Methodology

3.1 Collective Classification

Collective classification refers to the combined classification of nodes based on correlations between unknown and known labels [5, 12]. This is different from ensemble learning [3, 4]. Given the labels of the instances in training set $Tr \subset All$, the task of collective classification in HIN is to infer the labels of the testing set ($Te = All - Tr$). We address collective classification problem using HMPS to find users (unknown labels) that are *similar* to spammers (known labels). We employ collective classification approach as it has been shown to achieve better accuracy as compared to independent evaluation [12].

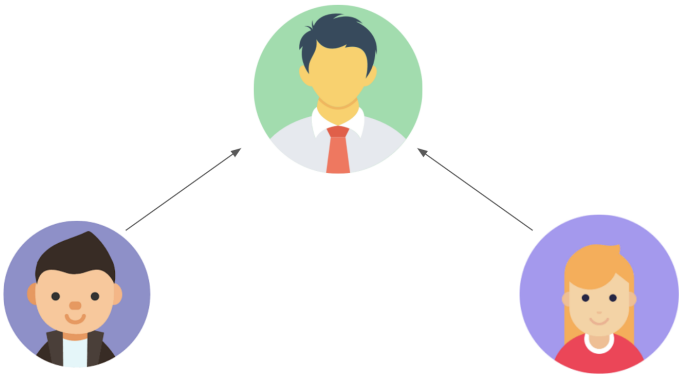

Figure 3.1: Perception of similarity in Humans

3.2 Hierarchical Meta-Path Score (HMPS)

3.2.1 Intuition

As discussed, Heterogeneous Networks have been used to find similarity between nodes in a network. We combined the above network concepts with the way humans view similarity to come up with HMPS.

Humans tend to have a hierarchical view of similarity, we say that siblings are more similar to each other than any other randomly selected people as we know that they have common father. Similarly, in case of academia, we may say that two researchers working in the same domain are much more similar than any two randomly chosen ones; in this case the domain can be seen as a hierarchical step above the researchers.

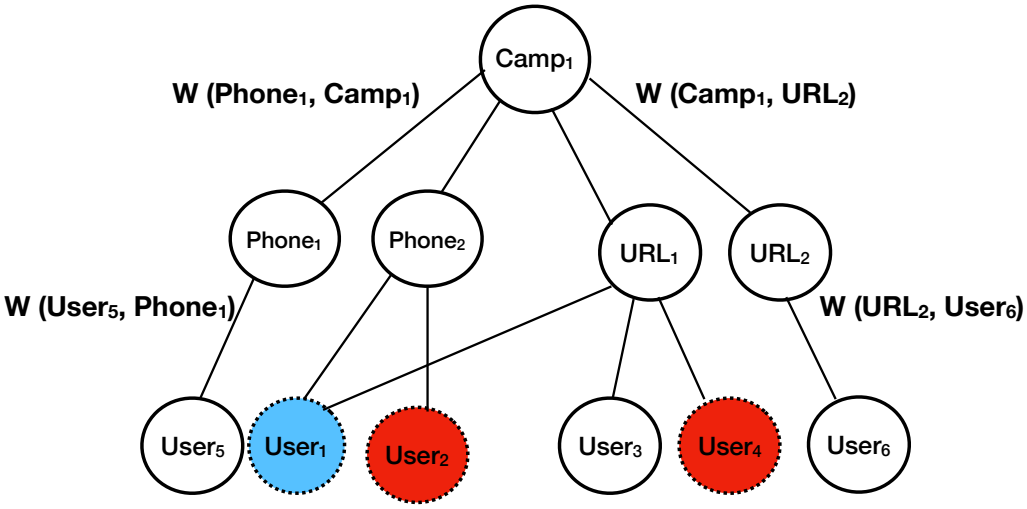

Figure 3.2: The hierarchical structure.

3.2.2 Formalisation

For calculating HMPS, we make our HIN such that an edge exists between: (i) Users and Phone number they shared, (ii) Users and URLs they shared, (iii) Phone number and Campaign node it belongs to, (iii) URL and Campaign node it belongs to. We can represent these nodes in a Hierarchical structure as in Fig. 3.2

The above structure hinges on the hypothesis that 2 Twitter users sharing the same phone number for a repeated number of times are bound to be similar to each other and if one is a spammer, it is highly likely that the other person would be as well. Further, we add another step to the hierarchy by adding the campaign node, as a result 2 users are similar if their post content is similar (definition of campaign).

To find the similarity between 2 leaf nodes (Twitter users) in the HMPS structure, we adopt something similar to finding LCA of the 2 users, except that we don't restrict ourselves to least

ancestor and check if we get a higher score by going to an ancestor on a higher level.
To give score to similarity we give weight to each edge as explained later. We then use the HMPS algorithm (which owes similarity to LCA) to find the path from one user to the other. For each unknown user, we find its similarity to all the spam users. We sum up the score to finally get what we the Hierarchical Meta-Path Score.
The following is the algorithm to calculate HMPS:

Algorithm 1 HMPS for Collective Classification

for $Camp_i \in Campaigns$ **do**
  S = Set of known spammers in $Camp_i$ ($m = |S|$); U = Set of unknown users in $Camp_i$; n = Total number of users in $Camp_i$
  $score_i \leftarrow \sum_{j=1}^{m} \text{HMPS}(U_i, S_j, Camp_i) \forall i \in [1, n]$
end for
procedure HMPS(u, s, $camp$)
  $res = 0$
  for $i \in Parent(u)$ **do** ▷ $Parent(u)$ = Immediate antecedent of u
    for $j \in Parent(u)$ **do**
      if $i == j$ **then** ▷ $W(s, j)$=weight of the edge $\langle s, j \rangle$ in the hierarchical structure
        if $W(u, i).W(s, j) > res$ **then**
          $res \leftarrow W(u, i).W(s, j)$
        end if
      else
        if $W(u, i).W(s, j).W(i, camp).W(j, camp) > res$ **then**
          $res \leftarrow W(u, i).W(s, j).W(i, camp).W(j, camp)$
        end if
      end if
    end for
  end for
  return res
end procedure

Here,

$$W(x, y) = \frac{\text{Number of Posts by x or associated with x connected to y}}{\text{Total number of posts associated to y}}$$

So, for example, if x were a user U and y a phone number P.

$$W(U, P) = \frac{\text{Number of times U posted the phone number P}}{\text{Total number of times P was posted in the entire dataset}}$$

3.3 Active Learning with Feedback

One of the objective of the project was to try to make the classification of spammers on OSNs run without the need of manual annotations. For this, the accounts suspended by Twitter were taken as the ground truth. As now only one class of values were present, One Class Classifier was used to classify user accounts. As different spam campaigns tend to have different network connections and hence different range of HMPS score, it was necessary to have a different classifier for every campaign. One Class SVM was used for this purpose. Two Class classifiers would have posed the problem of imbalanced dataset and the need to manually annotate the negative

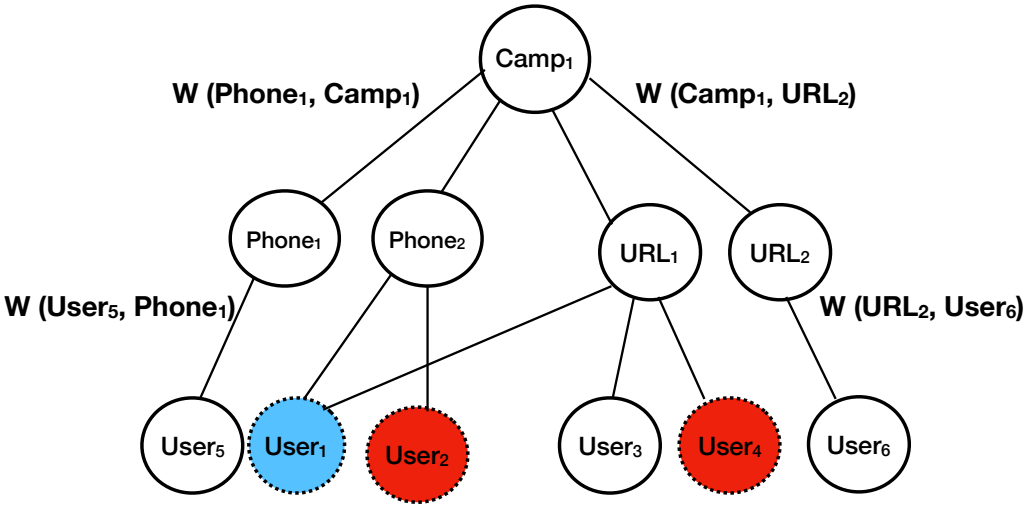

Figure 3.3: Schematic diagram of active learning with feedback

class, both of these problems was solved using the One Class classifier.

As the training dataset in this case for each campaign was extremely less, feedbacks were provided to the classifiers. A user that was predicted by a classifier as spam in one campaign with a high value of decision function i.e. with high confidence was given to the next classifier as a known spam node. This was done for multiple iterations till no new high confidence spam users were predicted. The definition of high confidence was varied to find the one that worked the best. It was found that the results were best when we took the users whose decision function was more than the maximum of the decision functions of the training set for the classifier. Here the decision function essentially depicts the distance of the point from the SVM decision boundary. The criterion for selecting users can be formulated as:

Given (a) a one-class classifier C, represented by the function $f(x)$ which, for an instance x, provides the distance of x from the classification boundary, and (b) X, a set of unlabeled instances, we take the maximum distance among all the training samples from the decision boundary, $T^c_{max} = \max_{x \in X} f(x)$. Now, from the unknown set X_u which are labeled by C, we choose those instances X'_u such that $\forall x \in X'_u : f(x) \geq T^c_{max}$. Note that the threshold T^c_{max} is specific to a campaign.

Chapter 4

Results

4.1 Basline Methods

We compare our method with three state-of-art methods proposed in the literature for spam detection in general. However, none of them focused on phone number specific spammers whose dynamics are different. Since we did not obtain the source code, we implemented the methods on our own. Note that all the baselines originally used 2-class classifiers. However, in this paper, we show the results of the baselines both for one-class and 2-class classifications after suitable hyper-parameter optimization.

Baseline 1: We consider the spam detection method proposed by Benevenuto et al. [2] as our first baseline. They proposed the following OSN-based features (referred as **OSN1**) per user: fraction of tweets with URLs, age of the user account, average number of URLs per tweet, fraction of followers per followee, fraction of tweets the user replied, number of tweets the user replied, number of tweets the user receives a reply, number of friends and followers, average number of hashtags per tweet. They showed that the SVM-based classifier performs best.

Baseline 2: We consider the method proposed by Khan et al. [7] to segregate spammers from genuine experts on Twitter as our second baseline. They suggested the following features (referred as **OSN2**): authority and hub scores of users in the follower-followee network, fraction of the user's tweets that contain the URLs, average number of URLs in a tweet, average number of URLs per number of words in a tweet of the user, average number of hashtags per number of words in a tweet, and average number of hashtags in a tweet. They showed that Logistic Regression performs best.

Baseline 3: We consider the method proposed by Adewole et al. [1] to detect spam messages and spam user accounts as our third baseline. They proposed the following list of profile and content-based features (referred as **OSN3**): length of the screen name based on characters, the presence or absence of profile location, whether the user includes URL or not in his profile, age of the account in days, number of followers of the user, number of friends / followers of the user, total statuses of the account, number of tweets the user has favorited, indicating presence

or absence of profile description, whether the user has not modified the theme of their profile, presence or absence of time zone, whether the account has been verified or not, whether the user has not changed the default profile egg avatar, number of the public lists the user is a member, whether or not the user has enabled the possibility of geo-tagging their tweets, normalized ratio of followers to friends, ratio of the number of follower to friends, ratio of the number of friends to followers, (total, unique, and mean) number of tweets, hashtags, URLs, mentions, favorite count, and retweets, ratio of (hashtags, URLs, mentions, retweets) to total number of tweets, (hashtag, URLs, mention, retweet, tweet-length) deviation, average number of daily tweets, average tweet length, popularity ration, number of duplicate tweets, and maximum value of hashtag frequency. They showed that Random Forest performs best for the classification task.

Note that previous work considered only those campaigns which involve only URLs [1, 2, 7]. In our work, a phone number, being a stable resource, helped in forming campaigns better. Besides, most of the OSN features used in the baselines are easy to evade by spammers, whereas HMPS-based feature is difficult to manipulate.

4.2 Experimental Setup

Our proposed classification method is run separately with different features (HMPS, OSN1, OSN2, and OSN3) and their combinations. We use the standard grid-search technique to tune the hyper-parameters. For evaluation, we design two experimental settings:

(i) Setting 1: Our primary goal is to detect user accounts which are suspended by Twitter because they are spam accounts. Therefore, the set of suspended accounts constitutes the ground-truth for the classifiers. Out of all suspended accounts present in our dataset, we adopt leave-one-out cross-validation technique (due to the very limited number of suspended accounts) and report the average accuracy of the classifiers. Note that in this setting, we use one-class classifier for all the competing methods.

(ii) Setting 2: We believe that our method is capable of detecting those accounts which are spammers, but not suspended by Twitter yet. Therefore, we further invited **human annotators**[1] to annotate some non-suspended accounts as spammers or non-spammers. This will further help us to run the baseline methods which originally used binary classifiers. Since it is not possible to label all non-suspended users, we adopt a convenient sampling approach. We define user bins according to the number of campaigns the non-suspended users exist. Our sampling approach preferentially chooses users who are part of multiple campaigns to maximize the evidence per campaign – the probability of choosing a user belonging to multiple campaigns is higher than that for a user who is a part of a single campaign. Following this approach, we picked 700 users from 3,370 campaigns. Each user was labeled by three human annotators as spammers or non-spammers, and then the majority vote was considered as the final class. The inter-annotator agreement was 0.82 according to Cohen's kappa measure.

[1] All annotators were security researchers between the age group of 25 - 35 years.

Out of 700 manually annotated accounts, we hold out 20% of the dataset to be used as the test set in Setting 2. We repeat this experiment 50 times and report the average accuracy. Here also, we use one-class classifier for all the competing methods and consider 'spammer' as our target class.

Evaluation metrics: For comparative evaluation, we use the standard information retrieval metrics – Precision, Recall, F1-score, Area under the ROC curve (AUC).

4.2.1 Comparative Evaluation

Table 4.1 shows the performance of the competing methods for both settings. We report the results of our active-learning based one-class classifier with different feature combinations. [2] For setting 1 (leave-one-out), we report the performance w.r.t the *accuracy* (fraction of known spammers identified by the method) and observe that our method performs significantly well with only HMPS feature – it achieves an accuracy of 0.77, outperforming all baseline methods. However, incorporating OSN2 features along with HMPS further enhances 9.1% performance of our classifier, achieving an accuracy of 0.84.

A similar pattern is observed for setting 2. However, here our model with only HMPS turns out to be even stronger classifier, outperforming all others in terms of precision (0.99), F1-score (0.93) and AUC (0.88). Here also, incorporating most of the OSN features with HMPS does not enhance the performance of our method (or sometimes deteriorates the performance), except ONS2 which seems to be quite competitive. However, baseline 2 seems to be the best method w.r.t recall (0.92); but it significantly sacrifices the performance w.r.t. precision, F1-score, and AUC.

Nevertheless, **we consider the following setting as our default method since it outperforms other methods in almost all experimental setup: HMPS + OSN2 + one-class classifier + active learning**. Baseline 2 is considered as the best baseline method in the rest of the paper.

Table 4.1: Results for the classifier

Method	Feature	Setting 1	Setting 2			
		Accuracy	P	R	F1	AUC
Baseline 1	OSN1	0.62	0.86	0.71	0.77	0.48
Baseline 2	OSN2	0.58	0.84	**0.92**	0.87	0.52
Baseline 3	OSN3	0.62	0.86	0.66	0.74	0.47
Our	HMPS	0.77	**0.99**	0.87	**0.93**	**0.88**
	HMPS + OSN1	0.76	0.89	0.90	0.89	0.72
	HMPS + OSN2	**0.84**	0.98	0.88	**0.93**	0.87
	HMPS + OSN3	0.70	0.88	0.73	0.80	0.59
Our	HMPS + OSN2 - Active Learning	–	0.42	0.98	0.55	0.51

Justification behind superior performance of HMPS: All of the baseline methods rely

[2] We tried with other combinations as well such as HMPS+OSN1+OSN2, HMPS+OSN2+OSN3 etc. The results were not encouraging enough to be reported in the paper.

on the features that can be changed over time. These methods either consider URL attributes (baselines 1 and 3) within the tweets or changes in profile characteristics between a legitimate and spam user account (baselines 2 and 3). Given these specificities, it is easy for a spammer to manipulate these features. In contrast, HMPS relies on the monetization infrastructure (phone numbers) to identify campaigns and spammers. As discussed earlier, we aggregate tweets as part of the same campaign when they use multiple phone numbers wrapped around similar text. As a result, our method is resilient to spammers' manipulation. Furthermore, to understand how HMPS helps in improving the detection of spammers over the baselines, we manually analyze a sample of 'spammers'. Some of the users not identified by baselines 1 and 3 as spammers have a balanced number of friends and followers and a low number of tweets. In addition, users were not using URLs to spread the campaign. Therefore, all URL-based features do not aid in the detection task.

Baseline 2 measures the authority and hub scores based on the tweets with hashtags. As a result, it wrongly detects some benign users as spammers that were retweeting posts related to (say,) blood donation campaigns. When baseline 2 is combined with HMPS, the false positive rate is reduced since these users are not found in the spammer network.

In addition, HMPS can find spammers that are not suspended by Twitter yet. For instance, Figure 4.1 shows a spammer account that clearly violates the Twitter policy by promoting and posting repeated, pornographic content. [3] Surprisingly, this account has not been suspended by Twitter yet. However, we found similar such accounts suspended by Twitter. Interestingly, our system was able to identify this account as a spammer.

These examples show that HMPS can identify spammers that use phone numbers, which are not detected by the baseline systems and / or Twitter, and is, therefore, more effective in detecting spammers that spread phone numbers to promote campaigns.

Figure 4.1: An example spammer account (bio shown in (a), timeline shown in (b)) that has not been suspended by Twitter yet, but our system could detect it as spammer.

[3] https://support.twitter.com/articles/18311

Chapter 5

Conversation Network

5.1 Definition

In order to map the user relations better, we tried using the conversation networks. Contrary to our previous assumptions, this network was homogeneous, with nodes being the users. As in Fig 5.1 we created edges from User A to B in 3 instances:

1. User A RT'ed User B's post
2. User A Replied to User B's post
3. User A mentioned User B in his post

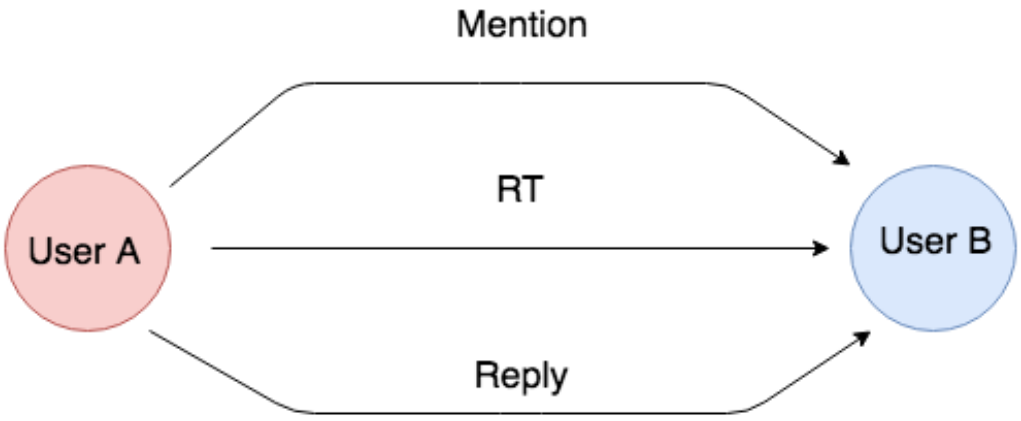

Figure 5.1: Types of edges between users

5.2 Methodology

We tried to use this network to transfer intelligence to find if we can call 2 users involved in active conversations as similar. We created these edges and found strongly connected components. For all the connected components we tried varying the threshold of weight where we add the edge. This means, we created an edge between 2 users only if the weight of the edge was greater than the threshold. We gave the weight to the conversation of 2 users on the basis on the conversation going on in between them. So,

$$W(UserA, UserB) = \frac{\text{Number of Conversations of A with B}}{\text{Total number of Conversation from A}}$$

From Fig 5.2 we tried to find the knee where we can cut the graph and take as the threshold

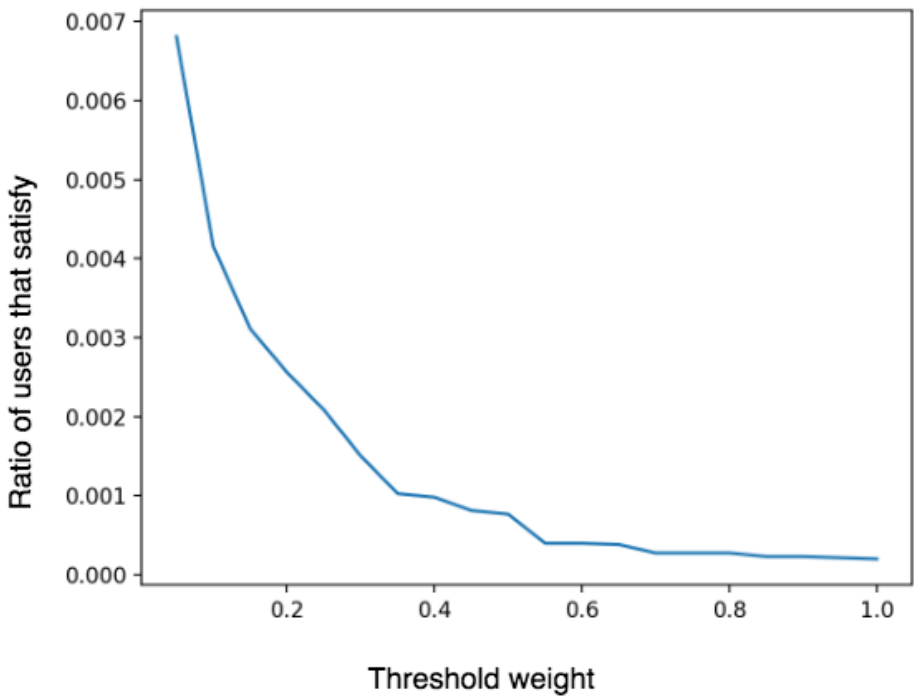

Figure 5.2: Selecting threshold to call users similar

to call users similar. We found a knee near 0.35 and took that as the cut-off. The next task was to check if the proposition held true or not, for this, we decided to check the results of such sampling with respect to those suspended by Twitter. If the users that are predicted by Conversation Networks as similar are a subset of those predicted by HMPS, we can assume the validity of this hypothesis at least to the data subset we considered.

5.3 Results

Unfortunately, the users predicted by the Conversation Networks belonged to different campaigns than the ones HMPS was able to run on. As a result, the only way to check this hypothesis was by manual annotation which didn't want to introduce into the system as we wanted to make a system that worked without the need of manual annotations.

Chapter 6

Iterating HMPS

6.1 Problem

HMPS gives amazing results when run on our dataset, but unfortunately, out of the available 22k campaigns, we were able to run the algo only on just over 3,000 campaigns. To increase the number of campaigns HMPS can encompass, we decided to further give another iteration of HMPS taking as ground truth the new predictions.

6.2 Initial Work

We started by predicting users just like we do using HMPS as described in the above chapters, but post that, to increase the number of campaigns encompassed, we considered the set of users predicted by HMPS as spam as the ground truth. We planned to run such iterations till our results converged.

6.3 Bottlenecks

HMPS uses Phone numbers and URLs to define 2 users' genealogy, to make sure 2 URLs are same, we need to ensure they are not shortened. Unshortening URL, on the other hand, is a complex task given that the edges for campaigns in our dataset can be extremely dense, and for certain campaigns the unshortening may take days to finish. To counter this, we planned to unshorten the URLs till we can for a specified point of time and then break soon. For this we needed to curate all the previous tasks and run them automatically at once.

6.4 Methodology

We started sampling campaigns based on the number of users and tweet present in the campaign. Even though this gave us the measure of quality of a campaign, we still could not be sure if the campaign selected would hav too many URLs to unshorten or not.

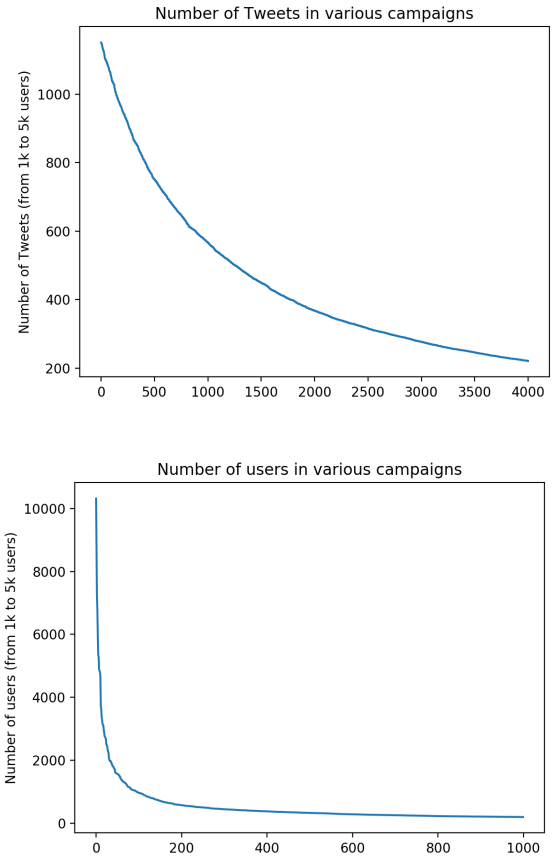

Figure 6.1: Methods to sample campaigns

We further decided to sample campaigns on the basis of URL overlaps. This measure would give us the number of campaigns a URL appears in. From this we can select the campaigns where we have the maximum number of URL overlap frequency, i.e. for every campaign we created a score on the basis of number of campaigns the URL present in it was appearing.

$$Score(Campaign) = \sum_{\forall URL_i \in Campaign} Freq(URL_i)$$

where,

$$Freq(URL) = \text{Number of Campaigns the URL belongs to}$$

We then started sampling campaigns in descending order. As in Fig 6.2 we sampled users in the order of more overlaps to less overlaps.

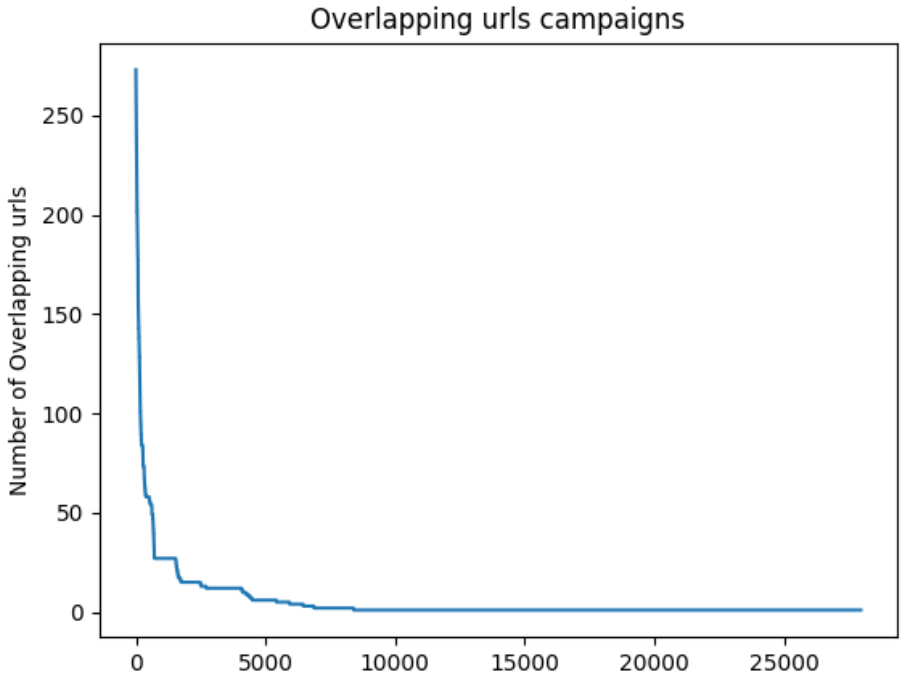

Figure 6.2: Number of Overlapping Users per campaign

6.5 Results

HMPS was run systematically using an automated script that fetched the users predicted in the previous iteration by the classifier back to the HMPS finding script. URL unshortening for campaigns was done in the order provided by campaigns in Fig. 6.2. These steps were performed till convergence.

Finally, the HMPS was able to run of **1,300** more campaigns than the it was able to run previously and then converged. This hints us to the fact that not a lot of campaigns are actually interconnected and to find spammers for other campaigns, we need a ground truth which we cannot get unless we: (i) manually annotate the users or (ii) Twiiter starts suspending them. We cannot do we cannot control (ii) and as we don't want to involve manual annotations (i) is not feasible, thus restricting us to the current number of campaigns.

Chapter 7

New Datasets and Future Work

7.1 Introduction

Results from previous chapters lead us to the fact the we have essentially exhausted our previous dataset and that we cannot get more predictions unless we change our initial problem statement. In order to define a future scope of the project, we started looking for multiple publicly available datasets that we can run HMPS on and show the generalisability of our proposed algorithm.

7.2 Datasets Considered

HMPS graphs and networks were made on the following datasets:

1. Fake like dataset by Sen et al. at Precog Lab IIITD (yet to be made public)
2. Facebook Likers Dataset [11]
3. Yelp Review Dataset [10]
4. Social Honeypot Dataset [6]

The HMPS graph on the above datasets was unsatisfactory and edges weights could not be properly defined due to improper hierarchy.

7.3 Future Datasets

Following Datasets can be further explored to run HMPS:

1. Phishing URL dataset [8]
2. Multiple other Twitter Content polluter Datasets

Bibliography

[1] Adewole, K. S., Anuar, N. B., Kamsin, A., and Sangaiah, A. K. Smsad: a framework for spam message and spam account detection. *Multimedia Tools and Applications* (2017), 1–36.

[2] Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)* (2010), vol. 6, pp. 1–12.

[3] Chakraborty, T., Chandhok, D., and Subrahmanian, V. Mc3: A multi-class consensus classification framework. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2017), Springer, pp. 343–355.

[4] Chakraborty, T., Pierazzi, F., and Subrahmanian, V. Ec2: Ensemble clustering and classification for predicting android malware families. *IEEE Transactions on Dependable and Secure Computing*, 1 (2017), 1–1.

[5] Gupta, S., Khattar, A., Gogia, A., Kumaraguru, P., and Chakraborty, T. Collective classification of spam campaigners on twitter: A hierarchical meta-path based approach. In *Proceedings of the 2018 World Wide Web Conference* (Republic and Canton of Geneva, Switzerland, 2018), WWW '18, International World Wide Web Conferences Steering Committee, pp. 529–538.

[6] Kamath, K. Y., Caverlee, J., Lee, K., and Cheng, Z. Spatio-temporal dynamics of online memes: a study of geo-tagged tweets. In *WWW* (2013).

[7] Khan, U. U., Ali, M., Abbas, A., Khan, S., and Zomaya, A. Segregating spammers and unsolicited bloggers from genuine experts on twitter. *IEEE Transactions on Dependable and Secure Computing* (2016).

[8] Kwon, H., Baig, M. B., and Akoglu, L. A domain-agnostic approach to spam-url detection via redirects. In *PAKDD* (2017).

[9] Meng, C., Cheng, R., Maniu, S., Senellart, P., and Zhang, W. Discovering meta-paths in large heterogeneous information networks. In *Proceedings of the 24th International Conference on World Wide Web* (2015), International World Wide Web Conferences Steering Committee, pp. 754–764.

[10] Rayana, S., and Akoglu, L. Collective opinion spam detection: Bridging review networks and metadata. In *KDD* (2015).

[11] Satya, P. R. B., Lee, K., Lee, D., Tran, T., and Zhang, J. Uncovering fake likers in online social networks. In *CIKM* (2016).

[12] Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine 29*, 3 (2008), 93.

[13] Shi, C., Kong, X., Huang, Y., Philip, S. Y., and Wu, B. Hetesim: A general framework for relevance measure in heterogeneous networks. *IEEE Transactions on Knowledge and Data Engineering 26*, 10 (2014), 2479–2492.

[14] Sun, Y., Han, J., Yan, X., Yu, P. S., and Wu, T. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment 4*, 11 (2011), 992–1003.