Community Detection – Detection of Fake Profiles on Facebook

Information Retrieval Project

Aman Shenoy – 2016A8PS0393P
Teertha Raj Chatterjee – 2016A1PS0689P
Prajjwal Mahajan – 2016A7PS0123P

Abstract— **The social network, a crucial part of our life is plagued by online impersonation and fake accounts. According to the 'Community Standards Enforcement Report' published by Facebook on March 2018, about 583 million fake accounts were taken down just in quarter 1 of 2018 and as many as 3-4% of its active accounts during this time were still fake. In this project, we propose a model that could be used to classify an account as fake or genuine. This model uses Support Vector Machine as a classification technique and can process a large dataset of accounts at once, eliminating the need to evaluate each account manually. The community of concern to us here is Fake Accounts and our problem can be said to be a classification or a clustering problem. As, this is an automatic detection method, it can be applied easily by online social networks which has millions of profiles, whose profiles cannot be examined manually.**

I. Introduction

Problem Statement – *Given a dataset of social media accounts, propose a framework that is able to classify an account as genuine or fake.*

Background of the Problem –

a.) ***Description of the selected application domain*** –

The Application Domain of the following project was Community Detection. Community detection is key to understanding the structure of complex networks, and ultimately extracting useful information from them. Applications are diverse: from healthcare to regional geography, from human interactions and mobility to economics. In this project we will pertain to human interaction. The selected application domain was detecting fake accounts on Facebook.

In the present generation, the social life of everyone has become associated with the online social networks. Adding new friends and keeping in contact with them and their updates has become easier. The online social networks have impact on the science, education, grassroots organizing, employment, business, etc. Researchers have been studying these online social networks to see the impact they make on the people. Teachers can reach the students easily through this making a friendly environment for the students to study, teachers nowadays are getting themselves familiar to these sites bringing online classroom pages, giving homework, making discussions, etc. which improves education a lot. The employers can use these social networking sites to employ the people who are talented and interested in the work, their background check can be done easily.

b.) ***Motivation*** - The social networking sites are making our social lives better but nevertheless there are a lot of issues with using these social networking sites. The issues are privacy, online bullying, potential for misuse, trolling, etc. These are done mostly by using fake profiles.

In this project, we came up with a framework through which we can detect a fake profile using machine learning algorithms so that the social life of people become secured.

c.) ***Technical Issues included in our work*** – Because of Privacy Issues the Facebook dataset is very limited and a lot of details are not made public.

II. Literature Survey

We first outline the threat model we assume in this work. We then present required background and related work on abuse mitigation and the ground-truth, and social infiltration of fakes in OSNs.

A. ***Threat model*** - We focus on OSNs such as Facebook, RenRen, and Tuenti, which are open to everyone and allow users to declare bilateral relationships (i.e., friendships).

Capabilities. We consider attackers who are capable of creating and automating fake accounts on a large scale. Each fake account, also called a socialbot, can perform social activities similar to those of real users. This includes sending friend requests and posting social content. We do not consider attackers who are capable of hijacking real accounts, as there are existing detection systems that tackle this threat (e.g., COMPA). We focus on detecting fake accounts that can befriend a large number of benign users in order to mount subsequent attacks, as we describe next.

Objectives. The objective of an attacker is to distribute spam and malware, misinform, or collect private user data on a large scale. To achieve this objective, the attacker has to infiltrate the target OSN by using the fakes to befriend many real accounts. Such an infiltration is required because isolated fake accounts cannot directly interact with or promote content to most users in the OSN. This is also evident by a thriving underground market for social infiltration. For example, attackers can now connect their fake accounts with 1K users for $26 or less.

B. *Abuse mitigation and the ground-truth*

Due to the inapplicability of automated account suspension, OSNs employ abuse mitigation techniques, such as CAPTCHA challenges and photo-based social authentication, so as to rate-limit accounts that have been automatically flagged as fake or suspicious. Moreover, these accounts are pooled for manual inspection by experienced analysts who build a ground truth of real and fake accounts along with their features, before suspending or removing verified fakes.

While maintaining an up-to-date ground-truth is important for retraining deployed classifiers and estimating how effective they are in practice, it is rather a time- consuming and non-scalable task. For example, on an average day, each analyst at Tuenti inspects 250–350 accounts an hour, and for a team of 14 employees, up to 30K accounts are inspected per day. It is thus important to rank user accounts in terms of how likely they are to be fake in order to prioritize account inspection by analysts.

C. *Social infiltration*

In early 2011, a study was conducted to evaluate how easy it is to infiltrate large OSNs such as Facebook. In particular, they used 100 automated fake accounts to send friend requests to 9.6K real users, where each user received exactly one request.

Main results. They found that users are not careful in their befriending decisions, especially when they share mutual friends with the requester. This behaviour was exploited by the fakes to achieve large-scale social infiltration with a success rate of up to 80%, in which case the fakes shared at least 11 mutual friends with the victims. In particular, they reported two main results that are important for designing fake account detection systems. First of all, some users are more likely to be victims than others. The more friends a user has, the more likely the user is to accept friend requests sent by fakes posing as strangers, regardless of their gender or number of mutual friends. Second, attack edges are generally easy to establish in OSN such as Facebook. An attacker can establish enough attack edges such that there is no sparse cut separating real accounts from fakes.

Implications. The study suggests that one can predict victims of fake accounts from user-level activities using low-cost features (e.g., number of friends). In addition, the study shows that graph-based detection mechanisms that rely solely on the graph structure are not effective under social infiltration. As social infiltration is prominent in other OSNs new proposals for graph-based detection should extend their threat model and include attackers who can infiltrate on a large scale.

III. Research Gap

Building efficient, scalable solutions for big data services like Facebook, requires a representative sample of data for experimentation, and for drawing valid conclusions. However, getting a representative sample of the Facebook sub graph is a hard problem in itself. One of the major reasons for researchers being unable to get a convincing data sample is that Facebook's fine-grained privacy settings make majority of its content private, and publicly inaccessible. About 72% Facebook users set their posts to private. This private nature of Facebook has been a major challenge in collecting and analysing its content in the computer science research community.

In addition, existing techniques related to spread and mitigation of malicious content on Facebook haven't been studied comprehensively. Most of the techniques proposed for detecting malicious posts on Facebook lack comprehensive evaluation, which is essential to prove their worth and research contribution. There hardly exists any research in the computer science community which characterizes or analyses malicious content on Facebook on a large scale. The only large-scale study on Facebook was on a dataset of 187 million wall messages which were collected from a random sample of 3.5 million users by crawling their Facebook walls in 2009. It would be interesting to study how malicious content identified from a random sample of Facebook differs from malicious content on Facebook during events. It is possible that the characteristics of malicious Facebook content vary across different events and differ from malicious Facebook content in general. It would also be interesting to study if malicious content has evolved over time on Facebook.

We also saw some research attempts towards studying events from Facebook data. However, Twitter has largely been the focus of researchers for studying events. We saw how Twitter was found to be a vital actor during sporting events, political campaigns, forest fires and even earthquakes. Content on other social networks has, however, not been given much attention in this respect. It is reasonable to assume that other social networks including Facebook also carry event related content, which can be of importance to the population of Internet users where Twitter is not as widely used as some other social networks. Even though researchers found high overlap between Twitter and Facebook streams during events we are yet to see dedicated attempts at studying Facebook content during events.

IV. System Description

Overview - Each profile (or account) in a social network contain lots of information such as name, gender, number of friends, number of followers, number of likes, location etc. Some of this information are private and some are public. We have used information that is public, to determine the fake profiles in social Network as private information is not accessible. However, if our proposed scheme is used by the social networking companies itself then they can use the private information of the profiles for detection without violating any privacy issues. We have considered this information as features of a profile for classification of fake and real profiles.

The steps that we have followed for detection of fake profiles are as follows -

1. Features are selected to apply classification algorithms. The classification algorithm is discussed further. Attributes are selected as features if they are not dependent on other attributes and they increase efficiency of the classification. The features that we have chosen are discussed further.

2. After selection of attributes, the dataset of profiles that are already classified as fake or genuine are needed for the training purpose of the classification algorithm. We have used a publicly available dataset of 1337 fake users and 1481 genuine users consisting of various attributes including name, status count, number of friends, followers count, favourites, languages known etc.

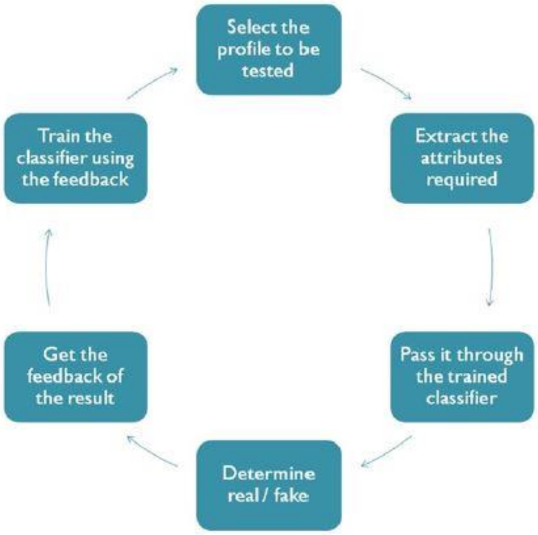
3. The selected attributes are extracted from profile for the purpose of classification.

4. After this the dataset of fake and real profiles are prepared. From this dataset, 80% of both profiles (genuine and fake) are used to prepare a training dataset and 20% of both profiles are used to prepare a testing dataset.

5. The training dataset is then fed to the classification algorithm. It learns from the training dataset and is expected to give correct class labels for the testing dataset.

6. The labels from the testing dataset are removed and are left for determination by the trained classifier. The result of classification algorithm is shown further. We have used a classification algorithm and checked the efficiency of this algorithm.

Proposed Framework **-** The proposed framework in the figure shows the sequence of processes that need to be followed for continuous detection of fake profiles with active learning from the feedback of the result given by the classification algorithm.


Select the profile to be tested
Extract the attributes required
Pass it through the trained classifier
Determine real / fake
Get the feedback of the result
Train the classifier using the feedback

This is a framework that can easily be implemented by social networking companies as they have access to user information.

1. Classification starts from the selection of profile that needs to be classified.

2. Once the profile is selected, the useful features are extracted for the purpose of classification.

3. The extracted features are then fed to trained classifier.

4. Classifier is trained regularly as new data is fed into the classifier.

5. Classifier then determines whether the profile is genuine or fake.

6. The result of classification algorithm is then verified and feedback is fed back into the classifier.

7. As the number of training data increases the classifier becomes more and more accurate in predicting the fake profiles.

Classification - Classification is the process of categorizing a data object into categories called classes based upon features/attributes associated with that data object. Classification uses a classifier, an algorithm that processes the attributes of each data object and outputs a class based upon this information. In this project, we use Support Vector Machine as a classifier. Support Vector Machine is an elegant and robust technique for classification on a large data set not unlike the data sets of Social Network with several millions of profiles.

SUPPORT VECTOR MACHINE

Support Vector Machine is a binary classification algorithm that finds the maximum separation hyperplane between two classes. It is a supervised learning algorithm that given enough training examples, divides two classes fairly well and classifies new examples.

Maximum Margin Classification - Given a weight vector w and bias weight b, we formulize the classification methodology as:

$$w^T x + b > 0 \ \Rightarrow \text{positive class}$$
$$w^T x + b < 0 \ \Rightarrow \text{negative class}$$

This equation gives a separator and it is intuitive that depending upon the choice of the above-mentioned parameters; we can have several separators for the same dataset.

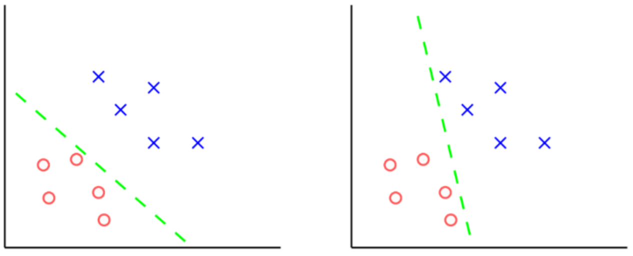


Figure 1 : Several decision boundaries may be possible for the same dataset

We can compute the distance of a training example from the decision boundary as:

$$distance(r) = \frac{w^T x_i + b}{||w||}$$

The training examples from each class closest to the separator are called support vectors and the distance between support vectors is called margin(ρ).

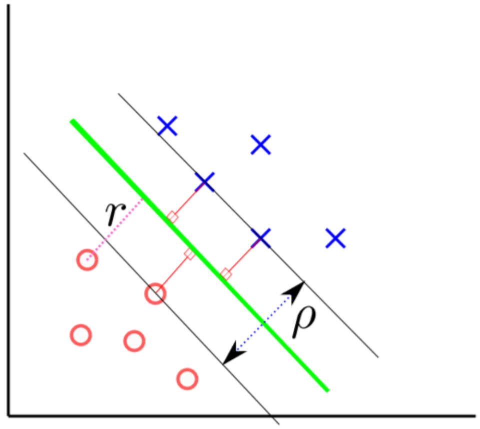


Figure 2 : Support vectors and Margin

We can judge how good or bad a separator is based upon its margin. The greater the margin, the better the separator. So, formal optimization problem is:

$$minimize\ \frac{1}{2}\left||w|\right|^2\ subject\ to\ y_i(w^T x_i + b) > 1\ for\ all\ i$$

Such a classification methodology that ensures that the separator maximizes the margin is called maximum margin classification and is a salient feature of Support Vector Machine.

Non-Linear Decision Boundaries **-** Certain function transforms a low dimensional input space into a higher dimensional output that is informative about the association of a training example with each class.). Mathematically,

$$For\ input\ space\ X\ and\ output\ space\ V, a\ kernel\ function\ k: X\ \times X\ \ \rightarrow R\ is:$$

$$k(x, x') = \{\varphi(x), \varphi(x')\}_v$$

$$where\ \varphi: X \rightarrow V\ is\ a\ feature\ map\ and\ \{,\}is\ a\ proper\ inner\ product$$

Several kernel functions exist, each suitable to a different application scenario. Kernels allow SVM to have non-linear separator (called decision boundary) while using a linear algorithm.

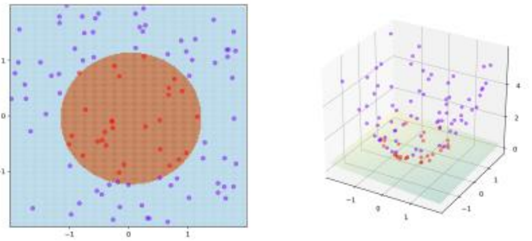


Figure 3 : SVM with kernel given by φ((a, b)) = (a, b, a2 + b2) and thus K(x, y) = x•y + x2 y2. The training points are mapped to a 3-dimensional space where a separating hyperplane can be easily found.

Dataset - We needed dataset of fake and genuine profiles. Various attributes included in dataset are number of friends, followers, status count. Dataset is divided into training and testing data. Classification algorithms are trained using training dataset and testing dataset is used to determine efficiency of algorithm. From the dataset used, 80% of both profiles (genuine and fake) are used to prepare a training dataset and 20% of both profiles are used to prepare a testing dataset.

Data Object Attributes **-** Status Count, Followers count, Friends Count, Favourites Count, Listed Count, Gender, Language Code

V. Evaluation Strategy

Accuracy = Number of correct predictions/ total number of predictions

Percent Error = (1-Accuracy) *100

Confusion Matrix - Confusion Matrix is a technique for summarizing the performance of a classification algorithm. Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

$$\text{True Positive Rate}(TPR) = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate}(FPR) = \frac{FP}{FP + TN}$$

$$\text{True Negative Rate}(TNR) = \frac{TN}{FP + TN}$$

$$\text{False Negative Rate}(FNR) = 1 - TPR$$

Recall- How many of the *true* positives were *recalled* (found), i.e. how many of the correct hits were also found.

$$Recall(R) = \frac{TP}{TP + FN}$$

Precision- Precision is how many of the *returned* hits were *true* positive i.e. how many of the found were correct hits.

$$Precision(P) = \frac{TP}{TP + FP}$$

F1 score- F1 score i s a measure of a test's accuracy. It considers both the precision *p* and the recall *r* of the test to compute the score.

$$F1\ score = \frac{(a^2 + 1)PR}{a^2 P + R}$$

$$where\ a = 1$$

ROC Curve- The *Receiver Operating Characteristic* is the plot of TPR versus FPR. ROC can be used to compare the performances of different classifiers.

VI. Experimental Results and Evaluation

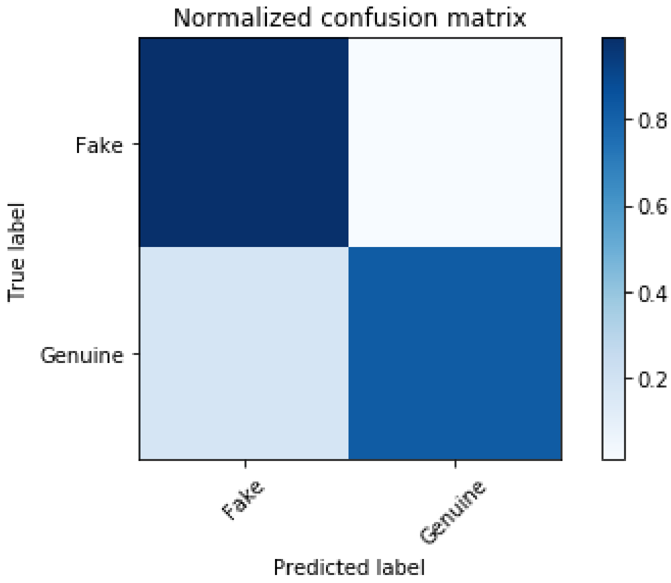
A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known.
For example –

	Predicted results	
Actual Results	Negative (Genuine)	Positive (Fake)
Negative (Genuine)	True Negative (TN)	False Positive (FP)
Positive (Fake)	False Negative (FN)	True Positive (TP)

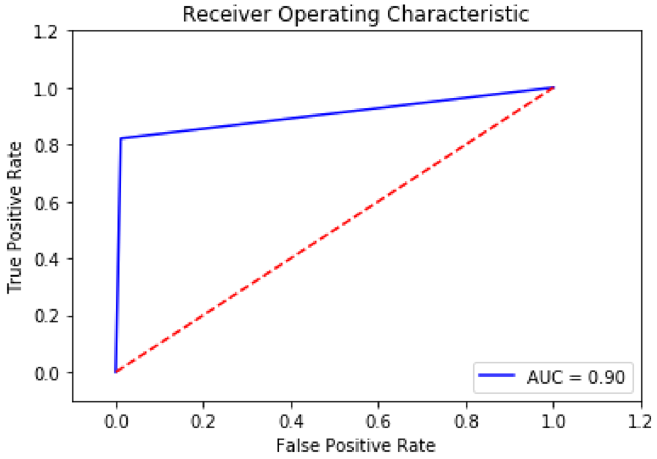
The evaluation statistics and the ground truth for the evaluation is given in the following table:

	Precision	Recall	F1-Score	Support
Fake	0.83	0.99	0.90	268
Genuine	0.99	0.82	0.90	296
Avg/Total	0.91	0.90	0.90	564

The confusion matrix for our evaluation model is presented as follows:



The following figure demonstrates the ROC curve :

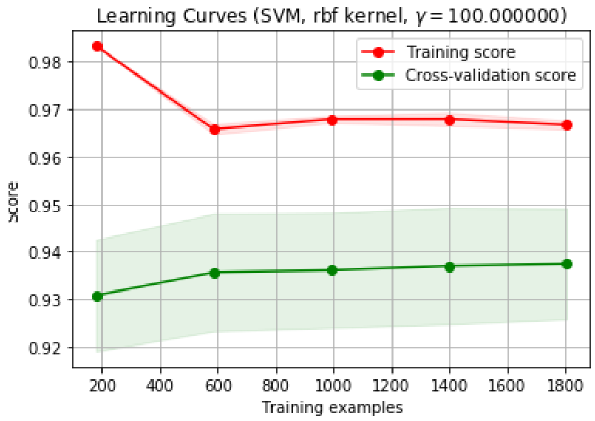


VII. Conclusions and Future Work

The model presented in this project demonstrates that Support Vector Machine (SVM) is an elegant and robust method for binary classification in a large dataset. Regardless of the non-linearity of the decision boundary, SVM is able to classify between fake and genuine profiles with a reasonable degree of accuracy (>90%).

This method can be extended on any platform that needs binary classification to be deployed on public profiles for various purposes. This project uses only publicly available information which makes it convenient for organizations that want to avoid any breach of privacy, but organizations can also use private data available to them to further extend the capabilities of the proposed model.

***Future Work* –** Since we have limited data to train the classifier, our approach is facing a high variance problem which can be observed in the learning curve as follows



High variance problems can usually be mitigated by increasing the size of the dataset which should not be of much concern to Social Networks Organizations which already have fairly large datasets.

References

[1] Ciao Xiao, David Mendell Freeman and Theodore Hwa, "*Detecting clusters of fake accounts in Online Social Networks"*, https://theory.stanford.edu/~dfreeman/papers/clustering.pdf

[2] Yazan Boshmaf, Georgos Siganos and Jorge Leria, *Íntegro: "Leveraging victim prediction for robust fake account detection in large scale OSNs"*, https://www.sciencedirect.com/science/article/pii/S0167404816300633

Yin Zhu, Xiao Wang, Erheng Zhong, Nanthan N. Liu , He Li and Qiang Yang, "*Discovering Spammers in Social Networks*", https://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/viewFile/5073/5135