

Automatic Detection of Fake Profiles in Online Social Networks

R. Nithin Reddy(108CS043) &
Nitesh Kumar(108CS064)



Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

Automatic Detection of Fake Profiles in Online Social Networks

*Thesis submitted in partial fulfillment
of the requirements for the degree of*

Bachelor of Technology

in

Computer Science and Engineering

by

Ranabothu Nithin Reddy

(Roll: 108CS043)

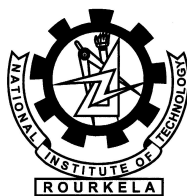
Nitesh Kumar

(Roll: 108CS064)

under the guidance of

Prof. Korra Sathya Babu

NIT Rourkela



Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela-769 008, Orissa, India



Department of Computer Science and Engineering
National Institute of Technology Rourkela

Rourkela-769 008, Orissa, India.

May 14, 2012

Certificate

This is to certify that the thesis entitled *Automatic detection of Fake Profiles in Online Social Networks* submitted by *Ranabothu Nithin Reddy and Nitesh Kumar*, in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Computer Science and Engineering at National Institute of Technology Rourkela is an authentic work carried out by them under my supervision and guidance. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university / institution for the award of any Degree or Diploma.

Prof. Korra Sathya babu

Assistant Professor

CSE department of NIT Rourkela

Acknowledgment

We are grateful to numerous local and global peers who have contributed towards shaping this project. At the outset, we would like to express our sincere thanks to Prof. K. Sathya Babu for his advice during our project work. As our supervisor, he has constantly encouraged us to remain focused on achieving our goal. His observation and comments helped us to establish the overall direction of the research and to move forward with investigation in depth. He has helped us greatly and been a source of knowledge.

We thank Barracuda Labs for supporting us by providing the necessary dataset required for the project. We sincerely thank everyone who has provided us with inspirational words, a welcome ear, new ideas constructive criticism, and their invaluable time. We must acknowledge the academic resources that we have acquired from NIT Rourkela.

Last but not the least, we would like to dedicate this project to our families, for their love, patience and understanding.

R. Nithin Reddy

Nitesh Kumar

Abstract

In the present generation, the social life of everyone has become associated with the online social networks. These sites have made a drastic change in the way we pursue our social life. Making friends and keeping in contact with them and their updates has become easier. But with their rapid growth, many problems like fake profiles, online impersonation have also grown. There are no feasible solution exist to control these problems. In this project, we came up with a framework with which automatic detection of fake profiles is possible and is efficient. This framework uses classification techniques like Support Vector Machine, Nave Bayes and Decision trees to classify the profiles into fake or genuine classes. As, this is an automatic detection method, it can be applied easily by online social networks which has millions of profile whose profiles can not be examined manually.

Contents

Certificate	ii
Acknowledgement	iii
Abstract	iv
List of Figures	vii
1 Introduction	1
1.1 History	1
1.2 Social Impact	2
1.3 Statistics	3
1.4 Issues	3
1.5 Motivation and Objective	3
2 Literature Review	4
2.1 Social Engineering	4
2.2 Online impersonation to defame a person	6
2.3 Advertising and Campaigning	7
2.4 Social Bots	9
2.5 Facebook Immune System(FIS)	9
3 Proposed Work	12
3.1 Overview	12
3.2 Proposed framework	14

3.3	Classification	16
3.4	Naive Bayes Classification	17
3.5	Decision Tree	18
3.5.1	Building a decision tree	19
3.6	Support Vector Machine	21
4	Implementation and Results	23
4.1	Datasets needed	23
4.2	Scrapping data	23
4.3	Attributes that we have considered	24
4.3.1	Why only these attributes ?	24
4.4	Evaluation parameters	25
4.5	Results	25
5	Conclusion and Future Work	29
	Bibliography	30

List of Figures

2.1	Example of social engineering	5
2.2	Example of online impersonation	6
2.3	Social influence via online social network	7
2.4	Example of avertising and campaigning	8
2.5	The adversarial cycle	10
3.1	Framework for detection of fake profiles and learning	15
3.2	General approach for building a classification model	16
3.3	The decision tree produced from the training dataset	19
3.4	Support Vector Machine classification for 2 dimensional data	22
4.1	Efficiency vs No. of profiles in training dataset	26
4.2	Efficiency vs No. of attributes considered in a profile	27
4.3	False Positive vs No. of profiles in training dataset	27
4.4	False Negative vs No. of profiles in training dataset	28

Chapter 1

Introduction

A social networking site is a website where each user has a profile and can keep in contact with friends, share their updates, meet new people who have the same interests. These Online Social Networks (OSN) uses web2.0 technology, which allows users to interact with each other.

These social networking sites are growing rapidly and changing the way people keep in contact with each other. The online communities bring people with same interests together which makes users easier to make new friends

1.1 History

These social networking sites starting with <http://www.sixdegrees.com> in 1997, then came <http://www.makeoutclub.com> in 2000. Sixdegrees.com couldnt survive much and closed very soon but new sites like myspace, LinkedIn, Bebo became successful and facebook was launched in 2004 and presently it is the largest social networking site in the world.

1.2 Social Impact

In the present generation, the social life of everyone has become associated with the online social networks. These sites have made a drastic change in the way we pursue our social life. 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 now-a-days teachers 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 using this. Most of the OSN are free but some charge the membership fee and uses this for business purposes and the rest of them raise money by using the advertising. This can be used by the government to get the opinions of the public quickly.

The examples of these social networking sites are sixdegrees.com, The Sphere, Nexopia which is used in Canada, Bebo, Hi5, Facebook, MySpace, Twitter, LinkedIn, Google+, Orkut, Tuenti used in Spain, Nasza-Klasa in Poland, Cyworld mostly used in Asia, etc. are some of the popular social networking sites.

1.3 Statistics

These online social networks are growing rapidly and there are more than 160 major social network websites exist in the world. 300 million active accounts in Facebook, 50

1.4 Issues

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.

1.5 Motivation and Objective

In todays online social networks there have been a lot of problems like fake profiles, online impersonation, etc. Till date, no one has come up with a feasible solution to these problems.

In this project we intend to give a framework with which the automatic detection of fake profiles can be done so that the social life of people become secured and by using this automatic detection technique we can make it easier for the sites to manage the huge number of profiles, which cant be done manually.

This thesis consists of 5 sections, in which Introduction was given in section 1, section 2 gives the literature review, section 3 gives the proposed framework, section 4 gives the implementation details, section 5 gives the conclusion and future work.

Chapter 2

Literature Review

Fake profiles are the profiles which are not genuine i.e. they are profiles of persons who claim to be someone they are not, doing some malicious and undesirable activity, causing problems to the social network and fellow users.

Why do people create fake profiles ?

- Social Engineering
- Online impersonation to defame a person
- Advertising and campaigning a person, etc

2.1 Social Engineering

Social Engineering in terms of security means the art of stealing confidential information from people or gaining access to some computer system mostly not by using technical skills but by manipulating people themselves in divulging information. The hacker doesn't need to come face to face with the user to do this.

The social engineering techniques are like Pretexting, Diversion theft, phishing, baiting, quid pro quo, tailgating, etc.

Eg: Creating a profile of some person X not in some online social networking site like

facebook. Adding the friends of the X in facebook and making them believe that its the profile of X. They can get the private information meant for only X by communicating with Xs friends in facebook.



Figure 2.1: Example of social engineering

Fig 2.1 shows the screenshot from yahoo news which shows the best example of social engineering done using an online social network facebook, in which some spies created a fake facebook account in the name of James Stavridis, the chief of NATO. They sent requests to many other officials in NATO and some officials in other important organizations and are able to extract a lot of important information.

2.2 Online impersonation to defame a person

The other reason why people create fake profiles is to defame the persons they do not like. People create profiles in the name of the people they dont like and post abusive posts and pictures on their profiles misleading everyone to think that the person is bad and thus defaming the person.

Hyderabad: Police arrests man who created fake Facebook account to defame woman. NEWS

🕒 Feb 26, 2012 📍 Nalgonda

Other

Action Taken: Following a complaint, Cyber Crime sleuths analysed the Internet Protocol (IP) log details and identified the accused and arrested him from Nalgonda.

The accused Mohammed Osman Ali, resident of Nalgonda district created a fake Facebook account of the victim and transmitted vulgar comments stating that the victim was going to deliver a baby shortly, officials of Cyber Crimes Cell in the CID said.

After his proposal was rejected, he bore a grudge against the complainant and created a fake Facebook profile in her name by uploading her photo, mobile number and personal details with obscene text, CID officials said.

Source: Press Trust of India February 25, 2012

Figure 2.2: Example of online impersonation

Fig 2.2 shows the screenshot from a website which shows that a man named Mohamad Osman Ali has created a fake profile of a woman in facebook and tried to defame her. The police finally caught and arrested him. This shows a very serious problem

existing now-a-days.

2.3 Advertising and Campaigning

Imagine a scenario where a movie is released and one of your friends in facebook posted that the movie was awesome. This makes a first impression on you that the movie is good and you would want to watch it. This is how advertising and campaigning works through OSN.

The review posted by a genuine user is always desirable but these reviews when posted by fake profiles and completely undesirable.

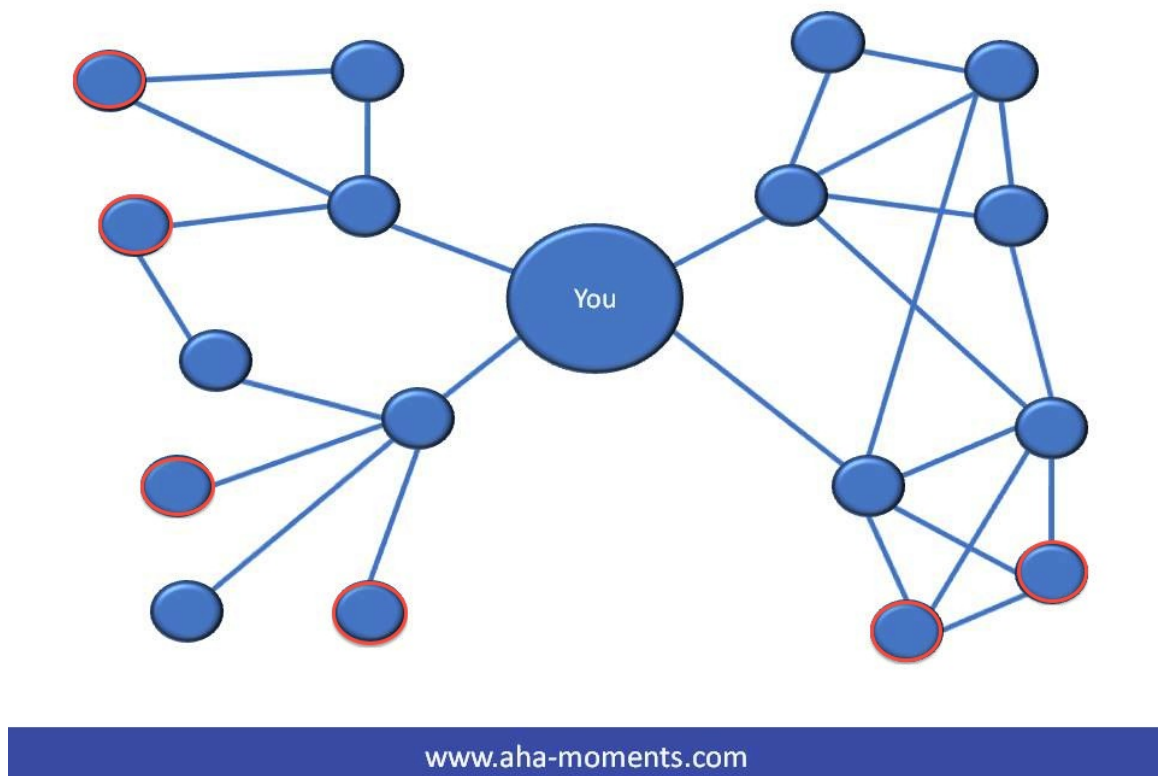


Figure 2.3: Social influence via online social network

Assume that Fig 2.3 shows a social graph where the blue nodes shown are real profiles, the red circled profiles show fake profiles and the edges show the connections between them. If the fake profiles start advertising a brand or campaigning for some politician then the users connected to the fake profiles are misled in believing them. Inturn the profiles who didnt add the fake profiles are effected using the mutual connections.



Figure 2.4: Example of avertising and campaigning

Fig 2.4 shows a screenshot, which shows the post in Newyork Times showing the most successful internet campaigning done by Obama which collected around 500 million dollars of election fund for him. Obama might not have used fake profiles in his internet campaigning but this shows the power of internet campaigning. Imagine a case where a non deserving candidate used this fake profiles to campaign. That is a

very highly undesirable situation.

2.4 Social Bots

Social bots are semi-automatic or automatic computer programs that replicate the human behavior in OSN. These are used mostly by hackers now-a-days to attack online social networks. These are mostly used for advertising, campaigning purposes and to steal users personal data in a large scale.

These social bots communicate with each other and are controlled by a program called botmaster. The botmaster may or may not have inputs from a human attacker. The social bots look like human profiles with a randomly chosen human name, randomly chosen human profile picture and the profile information posted randomly from a list prepared from before by the attacker. These social bots send requests to random users from a list. When someone accepts the request, they send requests to the friends of the user who accepted the request, which increases the acceptance rate due to existence of mutual friends.

Recently a researcher from university of british Columbia made a social botnet of 103 bots in facebook and added 3000 friends in just 8 weeks. He was able to extract around 250 GB of personal data of users. This shows the extent of the applications of social bots by the attackers.

2.5 Facebook Imune System(FIS)

When we consider facebook, it has its own security system to protect its users from spamming, phishing, etc. and this is called facebook immune system. FIS does real

time checks on every single click and every read and write operation done by it. This is around 25 Billion checks per day and as high as 620,000 checks per minute at peak as of may, 2011.

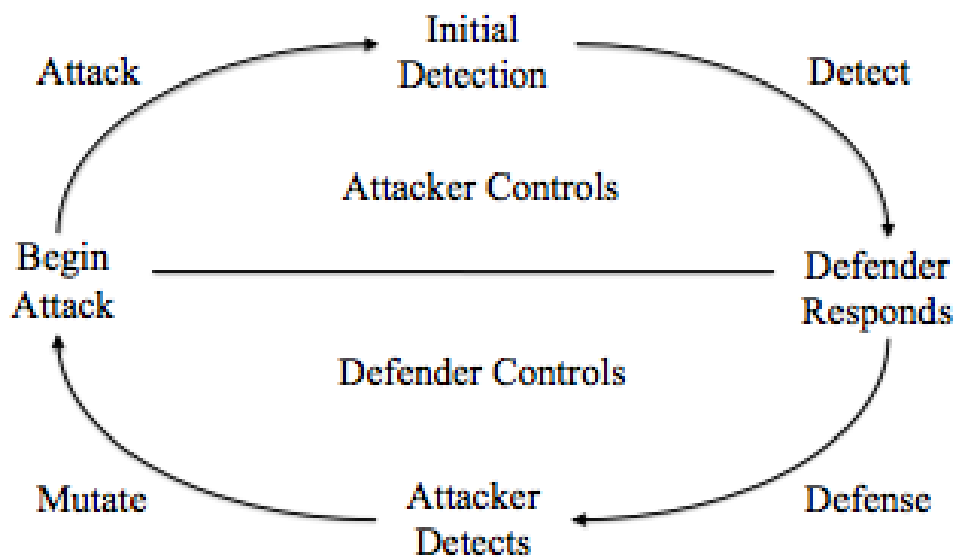


Figure 2.5: The adversarial cycle

Fig 2.5 shows the adversarial cycle in which the top part is controlled by the attacker and the bottom part shows the response by the FIS to control the attack, which when detected by the attacker, he/she mutates the attack and attacks it again. This goes on like a cycle and is never ending. FIS is able to detect the spam, malware and phishing produced by the compromised ad fake accounts. They are actually able to reduce the spam to less than 0.4

FIS is not successful in detecting the social bots and the fake accounts created by humans. This can be seen by the example mentioned above where a researcher created

103 social bots to collect a lot of personal data of users and facebook couldnt detect this attack.

Chapter 3

Proposed Work

3.1 Overview

Each profile (or account) in a social network contain lots of information such as gender, no. of friends, no. of comments, education, work etc. Some of these information are private and some are public. Since private information is not accessible so, we have used only the information that are public to determine the fake profiles in social network. 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 these 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. First all the features are selected on which the classification algorithm is applied.

The classification algorithm is discussed in the section 3.3. Proper care should be taken while choosing the features such as features should not be dependent on other features and those features should be chosen which can increase the efficiency of the classification. The features that we have chosen is discussed in section 4.3.

2. After proper selection of attributes, the dataset of previously identified fake and real profiles are needed for the training purpose of the classification algorithm. We have made the real profile dataset whereas the fake profile dataset is provided by the Barracuda Labs, a privately held company providing security, networking and storage solutions based on network appliances and cloud services. The collection of dataset is discussed in the section 4.1.
3. The attributes selected in step 1 are needed to be extracted from the profiles (fake and real). The process of extraction of the features that we have used is discussed in the section 4.2. For the social networking companies which want to implement our scheme dont need to follow the scrapping process, they can easily extract the features from their database. We applied scrapping of the profiles since no social network dataset is available publicly for the research purpose of detecting the fake profiles.
4. After this the dataset of fake and real profiles are prepared. From this dataset, 80% of both profiles (real and fake) are used to prepare a training dataset and 20% of both profiles are used to prepare a testing dataset. We find the efficiency of the classification algorithm using the training dataset containing 922 profiles and testing dataset containing 240 profiles.
5. After preparation of the training and the testing dataset, the training dataset is feed to the classification algorithm. It learns from the training algorithm and is expected to give correct class levels for the testing dataset.
6. The levels from the testing dataset are removed and are left for determination by

the trained classifier. The efficiency of the classifier is calculated by calculating the no. of correct prediction divided by total no. of predictions. The result of classification is shown in the section 4.5. We have used three classification algorithms and have compared the efficiency of classification of these algorithms

3.3.

3.2 Proposed framework

The proposed framework in the figure 3.1 shows the sequence of processes that need to be followed for continues detection of fake profiles with active leaning from the feedback of the result given by the classification algorithm. This framework can easily be implemented by the social networking companies.

1. The detection process starts with the selection of the profile that needs to be tested.
2. After selection of the profile, the suitable attributes (i.e. features) are selected on which the classification algorithm is implemented.
3. The attributes extracted is passed to the trained classifier. The classifier gets trained regularly as new training data is feed into the classifier.
4. The classifier determines the whether the profile is fake or real.
5. The classifier may not be 100% accurate in classifying the profile so; the feedback of the result is given back to the classifier. For example, if the profile is identified as fake, social networking site can send notification to the profile to submit

identification. If the valid identification is given, feedback is sent to the classifier that the profile was not fake.

6. This process repeats and as the time proceeds, the no. of training data increases and the classifier becomes more and more accurate in predicting the fake profiles.

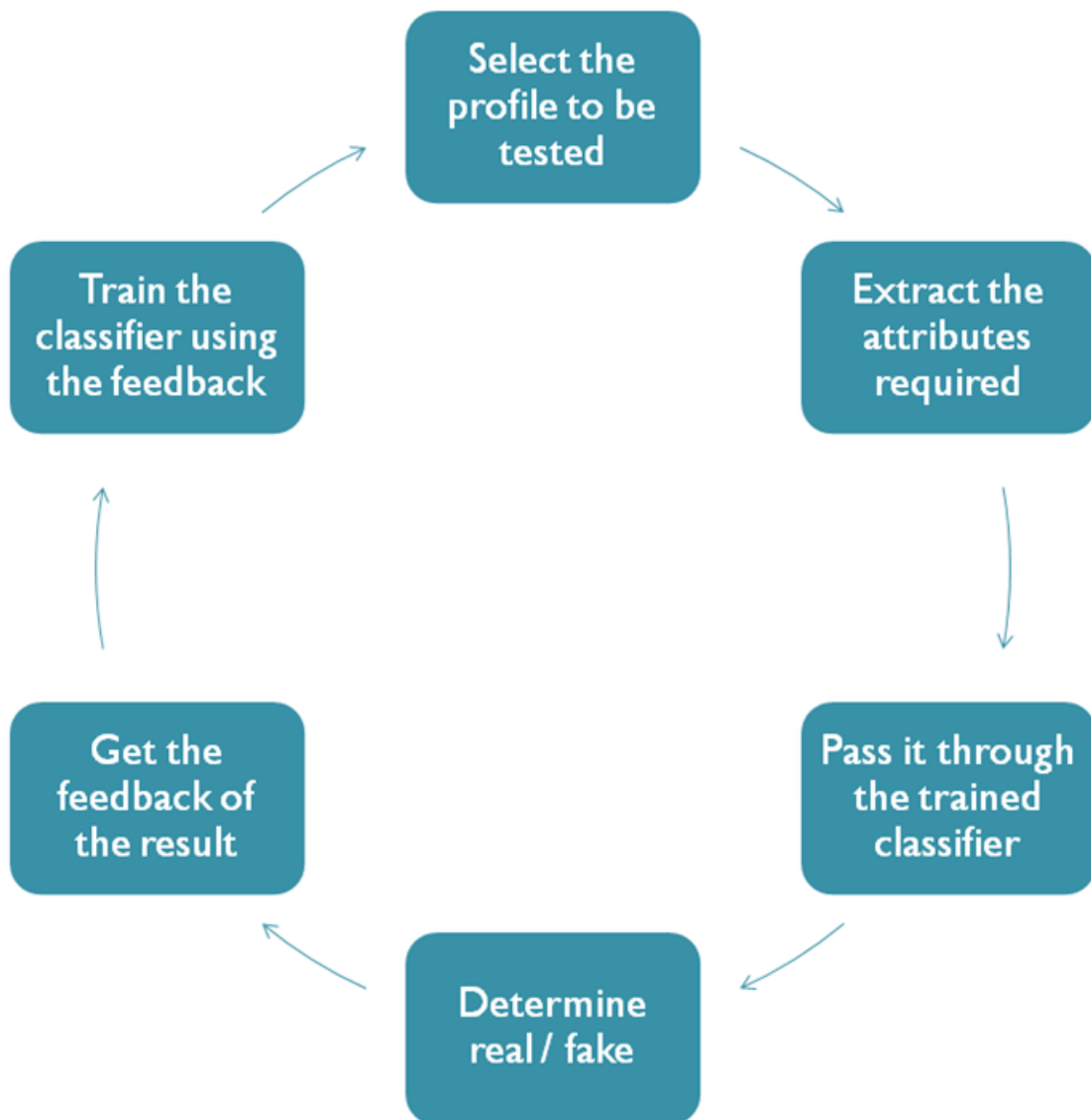


Figure 3.1: Framework for detection of fake profiles and learning

3.3 Classification

Classification is the process of learning a target function f that maps each records, x consisting of set of attributes to one of the predefined class labels, y . A classification technique is a approach of building classification models from an input data set. This technique uses a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the training set. The model generated by the learning algorithm should both fit the input data correctly and correctly predict the class labels of the test set with as high accuracy as possible. The key objective of the learning algorithm is to build the model with good generality capability. The figure 3.2 shows the general approach for building a classification model.

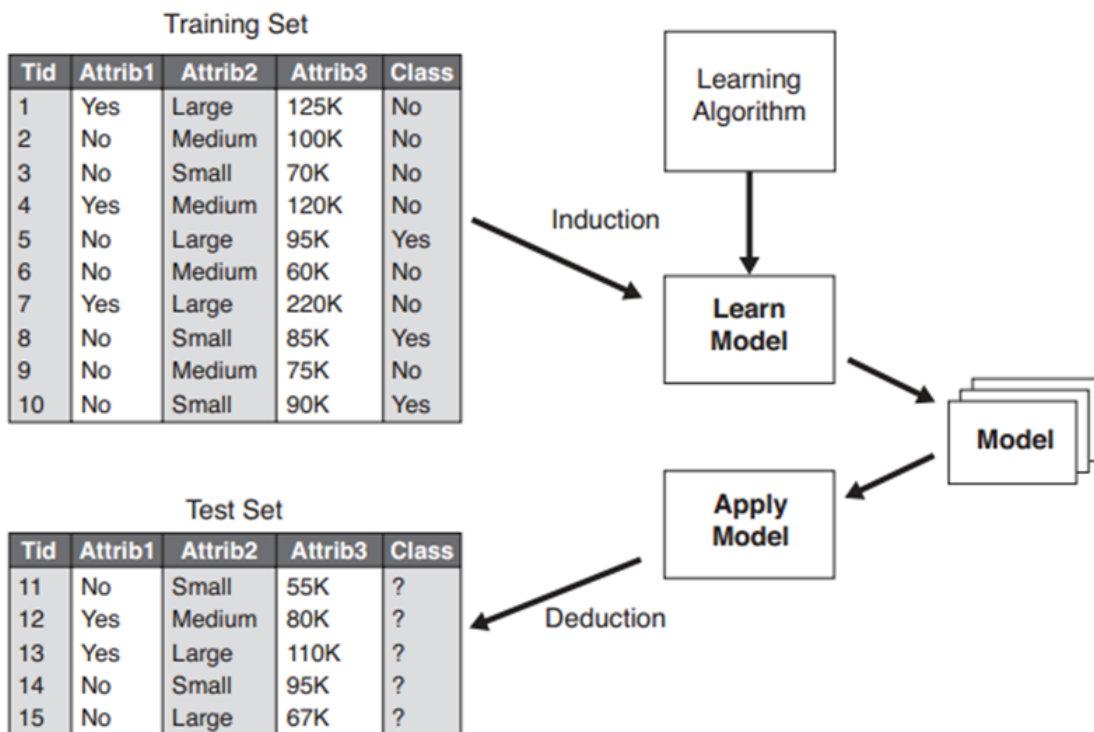


Figure 3.2: General approach for building a classification model

The classifiers that we have implemented for classifying the profiles are:

- Naive Bayes Classification
- Decision Tree Classification
- Support Vector Machine

All these algorithms are the standard algorithm and is widely used in problems such as detecting spam email messages, categorizing cells as malignant or benign based upon the results of MRI scans, classifying galaxies based upon their shapes etc.

3.4 Naive Bayes Classification

In Bayesian classification we have a hypothesis that the given data belongs to a particular class. We then calculate the probability for the hypothesis of being true. This is among the most practical approaches for certain types of problems. The approach requires only one scan of the whole data. Also, if at some stage additional training data is added then each training example can incrementally increase or decrease the probability that the hypothesis is correct.

Before we go further, we define the Bayes theorem as,

$$P(A | B) = \frac{P(B | A).P(A)}{P(B)}$$

Where $P(A)$ refers to the probability that event A will occur. $P(A | B)$ stands for the probability that event A will happen, given that event B has already happened.

The naive Bayes classifier exploits the Bayess rule and assumes independence of attributes. It assigns an instance S_k with attribute values $(A_1 = v_1, A_2 = v_2, \dots, A_m = v_m$

) with maximum $Prob(C_i | (v_1, v_2, \dots, v_m))$ for all i . For example the probability of assigning to class C_i and C_j is calculated for an instance S_k as,

Likelihood of S_k belonging to C_i

$$Prob(C_i | (v_1, v_2, \dots, v_m)) = \frac{P((v_1, v_2, \dots, v_m) | C_i)P(C_i)}{P((v_1, v_2, \dots, v_m))}$$

Likelihood of S_k belonging to C_j

$$Prob(C_j | (v_1, v_2, \dots, v_m)) = \frac{P((v_1, v_2, \dots, v_m) | C_j)P(C_j)}{P((v_1, v_2, \dots, v_m))}$$

Therefore, when comparing $Prob(C_i | (v_1, v_2, \dots, v_m))$ and $P(C_j | (v_1, v_2, \dots, v_m))$, we only need to compute $P((v_1, v_2, \dots, v_m) | C_i)P(C_i)$ and $P((v_1, v_2, \dots, v_m) | C_j)P(C_j)$.

Under the assumption of independent attributes,

$$\begin{aligned} P((v_1, v_2, \dots, v_m) | C_j) &= P(A_1 = v_1 | C_j).P(A_2 = v_2 | C_j)...P(A_m = v_m | C_j) \\ &= \prod_{h=1}^m P(A_h = v_h | C_j) \end{aligned}$$

Furthermore,

$$P(C_i) = \frac{\text{no. of training samples belonging to } C_i}{\text{total no. of training samples}}$$

All the probabilities are calculated using the data of training dataset.

3.5 Decision Tree

A decision tree is a popular classification method that generates tree structure where each node denotes a test on an attribute value and each branch represents an outcome of the test. The tree leaves represent the classes. The figure 3.3 shows the decision tree evaluated from our training dataset used in the project. It displays the relationships found in the training dataset. This technique is fast unless the training data is very

large. It does not make any assumptions about the probability distribution of the attributes value. The process of building the tree is called induction.

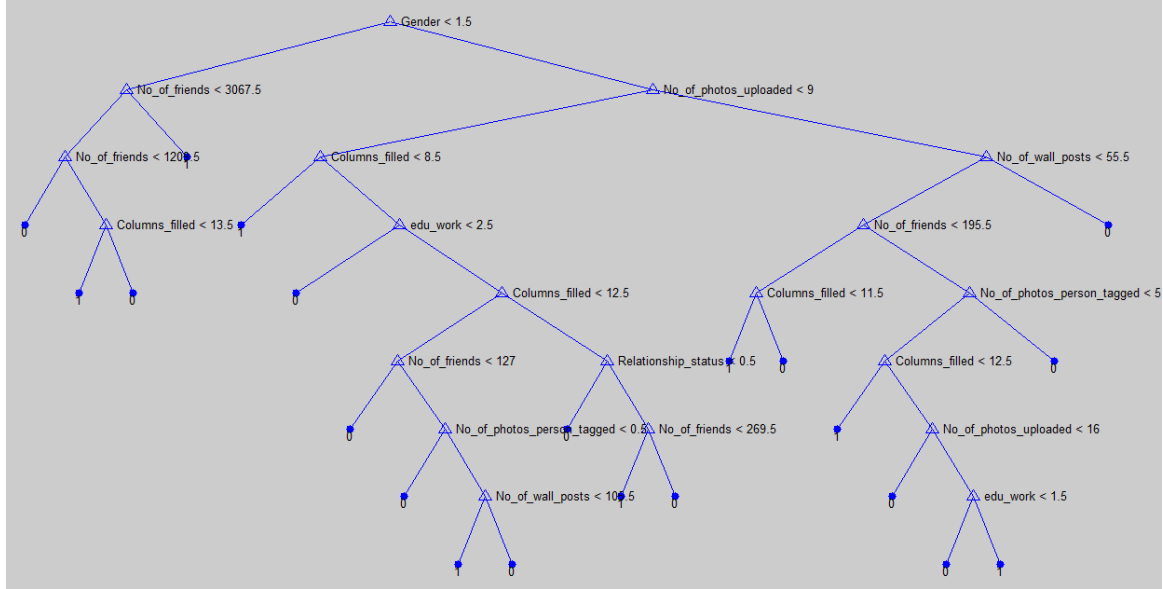


Figure 3.3: The decision tree produced from the training dataset

3.5.1 Building a decision tree

The decision tree algorithm is a top-down greedy algorithm which aims to build a tree that has leaves as homogenous as possible. The major step in the algorithm is to continue dividing leaves that are not homogeneous into leaves that are as homogeneous as possible until no further division is possible. The algorithm is described below:

1. If some of the attributes are continuous-valued, they should be discretized into categories.
2. If all instances in training dataset are in the same class, then stop.
3. Split the next node by selecting an attribute from the independent attributes that best divides the objects in the node into subsets and create decision tree

node.

4. Split the node according to the value of attribute selected in step 3
5. Stop if any of the following conditions meets, otherwise continue step 3:
 - (a) If this partition divides the data into subsets that belong to a single class and no more node needs splitting
 - (b) If there are no remaining attributes for further division.

The major step in the decision tree building algorithm is Step 3, where an attribute that best splits the data needs to be selected.

The discriminatory power of each attribute is evaluated using following rules:

- Rules based on Information Gain Measure
- Rules based on Gini Index

Information Gain Measure

Information also known as entropy measures the lack of order in a system. The information of a data set S with m classes is defined as,

$$I = - \sum_{k=1}^m P_k \cdot \log_2(P_k)$$

where P_k is the relative frequency of class k . The information gain for Sample S using attribute A is $Gain(S, A) = I - \sum_{i \in values(A)} (t_i/s) I_i$, where I is information before split and I_i is the information of node i . The attribute with the highest information gain is selected.

Gini's Index Measure

Gini's index is a ratio measure with values in the interval $[0,1]$ used to measure the discriminatory power of rating systems. For a data set S with m distinct classes, the simple Gini index is,

$$Gini(S) = 1 - \sum_{k=1}^m P_k^2$$

where P_k is the probability that an item belongs to class k (P_k is the relative frequency of class k). When the number of classes is large, some of the P_k can be small. The maximum of $Gini(S)$ occurs when each of the probabilities are equal with maximum value $1 - (1/m)$. The minimum occurs when all instances belong to the same class, with minimum value 0.

If the set S has a large number of classes, it can be computed recursively as follows.

If S is partitioned into two disjoint subsets $S = S_1 \cup S_2$, the Gini index can be found as $Gini(S) = (n_1/n)Gini(S_1) + (n_2/n)Gini(S_2)$ where $n_i = |S_i|$ is the cardinality of set S_i , for $i=1,2$. This result can be extended to m -partitions of a set S as $Gini(S) = \sum_{i=1}^m (n_i/n).Gini(S_i)$. The attribute that provide the minimum Gini index is chosen for splits.

We are using Gini index to identify the attribute that best splits the data.

3.6 Support Vector Machine

An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. An SVM classifies data by finding the best hyperplane that separates all data points of one class from those

of the other class. The support vectors are the data points that are closest to the separating hyperplane.

The figure 3.4 illustrates linear classification, with + indicating data points of type 1, and - indicating data points of type 0.

The datasets that we have used cannot be classified using linear classifier. So, non-linear classifier with Gaussian kernel is used. The implementation of SVM is done on Matlab.

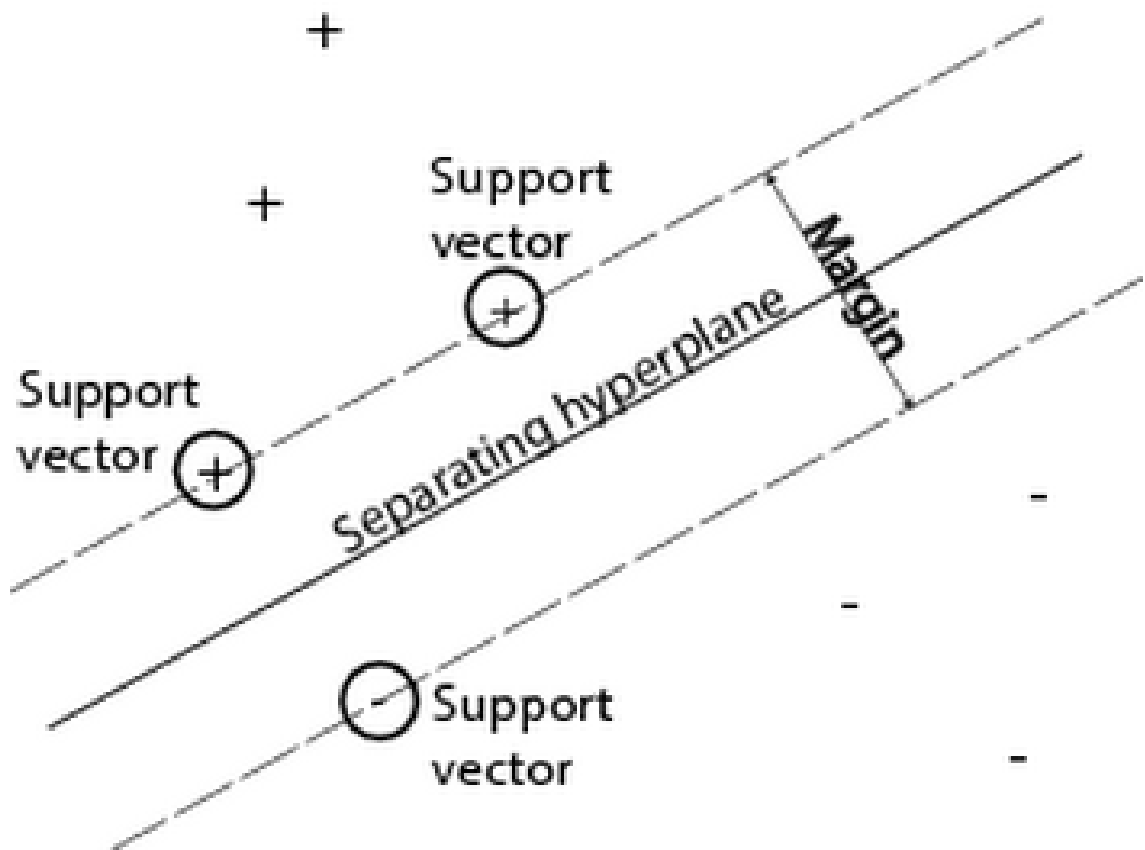


Figure 3.4: Support Vector Machine classification for 2 dimensional data

Chapter 4

Implementation and Results

4.1 Datasets needed

We need dataset with a mixture of real and fake profiles labeled accordingly. The algorithms need to be trained using the training dataset and should be evaluated using the testing dataset. But there are no such datasets available because of privacy issues.

As there is no standard dataset present, we need to prepare the dataset by scrapping the profiles from facebook. To scrap the data from the profiles, we need to be friends with the profiles which are being scrapped. We used the profile facebook/nitreddy with 957 friends to scrap the real profiles.

4.2 Scrapping data

Scripts written in python language were used which logs into facebook automatically and scraps required data. Facebook Graph API is also used along with python to extract some required data. Anti scrap detection techniques were implemented to prevent facebook immune system from detecting. 957 profiles were scrapped out of which some profiles were hiding data from friends also which were removed from the

dataset which left 872 real profiles in the dataset.

Barracuda labs is presently working on facebook spam detection making applications for them. They detected and scrapped 350 fake profiles and analyzed the data. We collected the data from them, filtered the profiles in which data is hidden, leaving 290 fake profiles in the dataset.

4.3 Attributes that we have considered

- No. of friends
- Education and work
- Gender
- No. of columns filled in about me
- Relationship status
- No. of photos of the person tagged *
- No. of wall posts posted by the person *
- No. of photos uploaded by the person *

* Indicates the attributes which are taken into account between 15th May 2011 and 15th September 2011

4.3.1 Why only these attributes ?

In the fake profiles dataset given by Barracuda labs, these were the only attributes we were able to extract.

Some other attributes which can be used in these classification algorithms are :

- Ratio of same gender friends and total friends.
- Ratio of the no. of friend requests sent and accepted
- No. of groups
- No. of likes, etc.

4.4 Evaluation parameters

-

$$\text{Efficiency} = \frac{\text{No. of correct predictions}}{\text{Total No. of Predictions}}$$

-

$$\text{False Positive rate} = \frac{\text{No. of real profiles detected as fake}}{\text{Total No. of fake profiles to be detected}}$$

-

$$\text{False Negative rate} = \frac{\text{No. of fake profiles detected as real}}{\text{Total No. of real profiles}}$$

4.5 Results

From the graph we find that the efficiency of the SVM is highest when the data is well trained and the efficiency of the Nave Bayes is lowest which dont change much when the training dataset increases.

As the no. of attributes increases for the training dataset the efficiency of all the algorithms increases.

The false positive rate of the SVM is least that means if a profile is detected fake then the chance of being fake is very high in SVM whereas Nave Bayes shows high false positive rate.

The false negative rate on the other hand is very low for Naive Bayes and the SVM has average false negative rate is the algorithm is well trained.

So, from the results we find that SVM is well suited for classification of the fake profiles in the social networks.

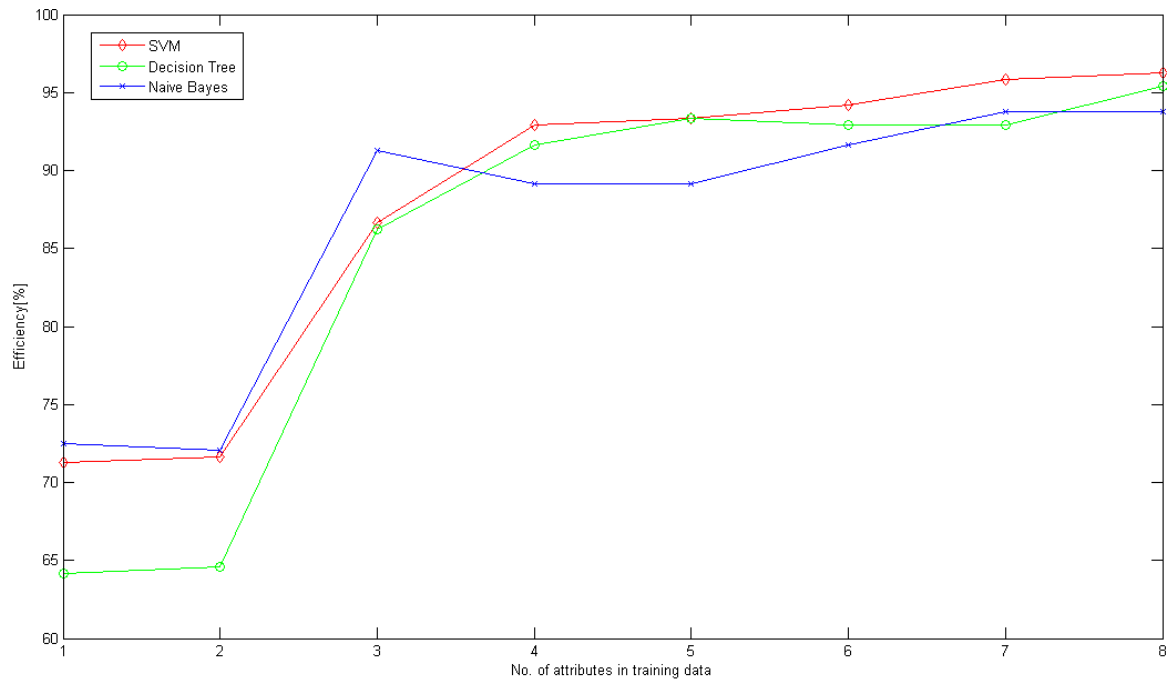


Figure 4.1: Efficiency vs No. of profiles in training dataset

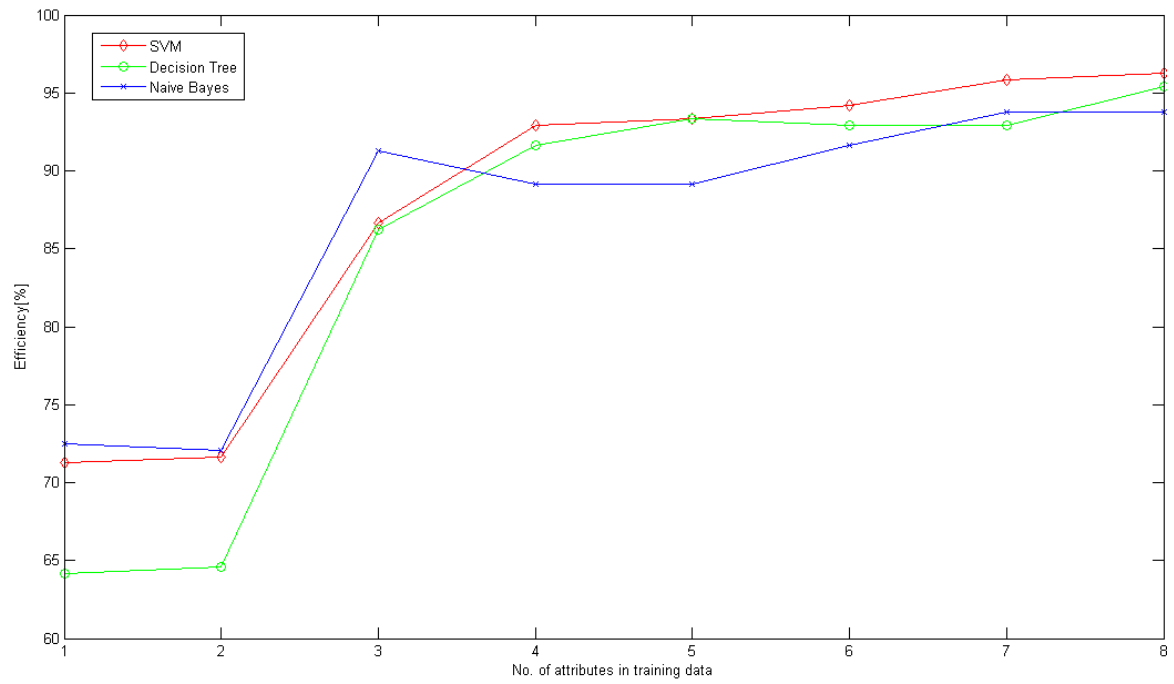


Figure 4.2: Efficiency vs No. of attributes considered in a profile

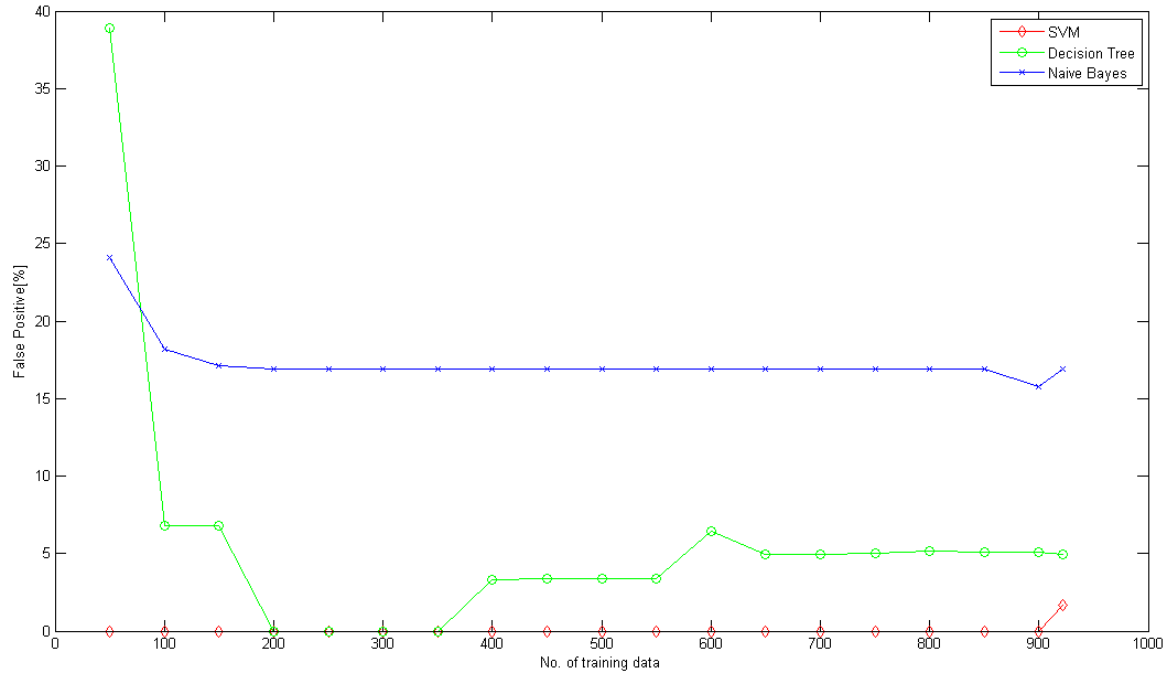


Figure 4.3: False Positive vs No. of profiles in training dataset

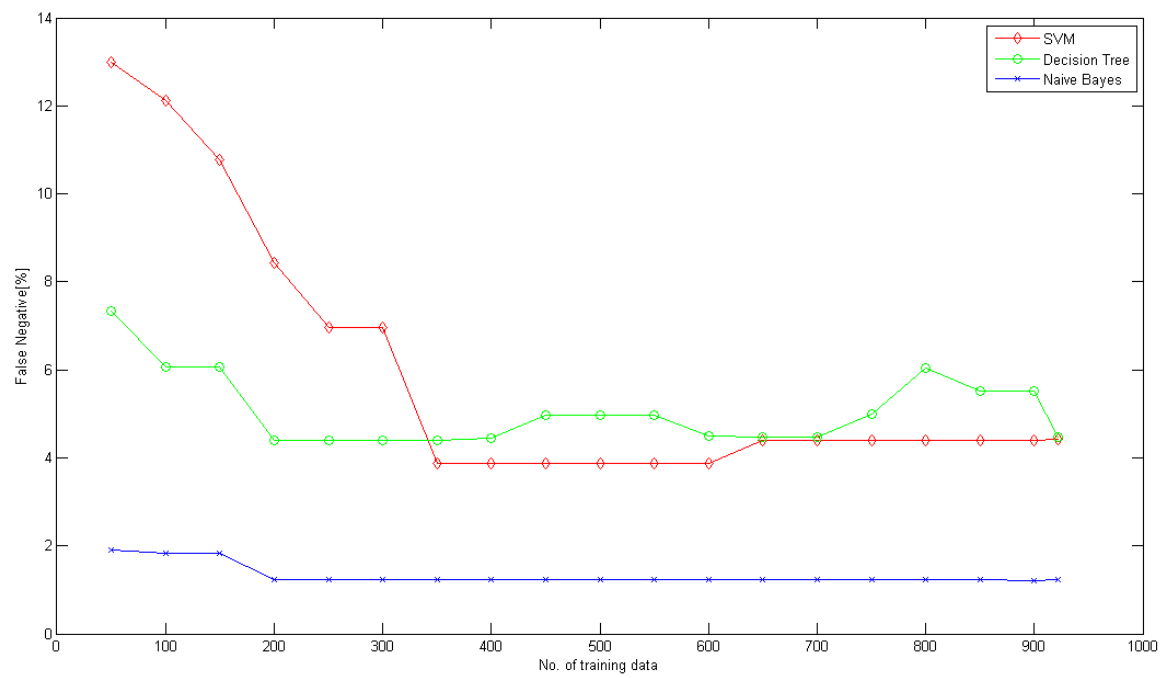


Figure 4.4: False Negative vs No. of profiles in training dataset

Chapter 5

Conclusion and Future Work

We have given a framework using which we can detect fake profiles in any online social network with a very high efficiency as high as around 95%. Fake profile detection can be improved by applying NLP techniques to process the posts and the profile

Bibliography

- [1] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems, SNS*, volume 11, page 8, 2011.
- [2] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 93–102. ACM, 2011.
- [3] C. Wagner, S. Mitter, C. Körner, and M. Strohmaier. When social bots attack: Modeling susceptibility of users in online social networks. In *Proceedings of the WWW*, volume 12, 2012.
- [4] G. Kontaxis, I. Polakis, S. Ioannidis, and E.P. Markatos. Detecting social network profile cloning. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 295–300. IEEE, 2011.
- [5] A. Wang. Detecting spam bots in online social networking sites: a machine learning approach. *Data and Applications Security and Privacy XXIV*, pages 335–342, 2010.
- [6] 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 annual conference on Internet measurement*, pages 35–47. ACM, 2010.
- [7] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 21–30. ACM, 2010.
- [8] S. Krasser, Y. Tang, J. Gould, D. Alperovitch, and P. Judge. Identifying image spam based on header and file properties using c4. 5 decision trees and support vector machine learning. In *Information Assurance and Security Workshop, 2007. IAW'07. IEEE SMC*, pages 255–261. IEEE, 2007.
- [9] G.K. Gupta. *Introduction to Data Mining with Case Studies*. Prentice Hall India, 2008.
- [10] Rajan Chattamvelli. *Data Mining Methods*. Narosa, 2010.
- [11] Spies create fake facebook account in nato chief’s name to steal personal details, <http://in.news.yahoo.com/spies-create-fake-facebook-account-nato-chiefs-name-114824955.html>.

- [12] Man arrested for uploading obscene images of woman colleague, <http://www.ndtv.com/article/andhra-pradesh/man-arrested-for-uploading-obscene-images-of-woman-colleague-173266>.
- [13] How obamas internet campaign changed politics, </bits.blogs.nytimes.com/2008/11/07/how-obamas-internet-campaign-changed-politics>.
- [14] S. Nagaraja, A. Houmansadr, P. Piyawongwisal, V. Singh, P. Agarwal, and N. Borisov. Stegobot: A covert social network botnet. In *Information Hiding*, pages 299–313. Springer, 2011.
- [15] M. Huber, M. Mulazzani, and E. Weippl. Who on earth is mr. cypher: Automated friend injection attacks on social networking sites. *Security and Privacy—Silver Linings in the Cloud*, pages 80–89, 2010.