INTERNATIONAL JOURNAL OF RESEARCH AND DEVELOPMENT IN ENGINEERING SCIENCES
e-ISSN: 2582 - 4201 Volume: 2 | Issue: 4 | Year: 2020

ANALYSIS ON DETECTION OF SPAMMERS AND FAKE USERS ON SOCIAL NETWORKS

Dr. Molli Srinivasa Rao

Professor and HOD,
Department of Computer Science and Engineering,
Viswanadha Institute of Technology and Management, Visakhapatnam
Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh, India

P.SATYA SWATHI [2], **P. DIVYA SANTHI** [3], **A.MARYSUPRIYA**[4], **G.YAMINI** [5]

[2,3,4,5] IV B.Tech Students,
Department of Computer Science and Engineering,
Viswanadha Institute of Technology and Management, Visakhapatnam
Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh

Abstract: - *Social networking sites engage millions of users around the world. The users' interacts with these social sites, such as Twitter and Facebook have a tremendous impact and occasionally undesirable repercussions for daily life. The prominent social networking sites have turned into a target platform for the spammers to disperse a huge amount of irrelevant and deleterious information. Twitter, for example, has become one of the most extravagantly used platforms of all times and therefore allows an unreasonable amount of spam. Fake users send undesired tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption. Moreover, the possibility of expanding invalid information to users through fake identities has increased those results in the unrolling of harmful content. Recently, the detection of spammers and identification of fake users on Twitter has become a common area of research in contemporary online social Networks (OSNs). In this we perform a review of techniques used for detecting spammers on Twitter.*

Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features. We are hopeful that the presented study will be a useful resource for researchers to find the highlights of recent developments in Twitter spam detection on a single platform.

Key words: *Machine Learning, Internet, Spam, Filters, OSNs, URL, Social Networking*

1. INTRODUCTION

Spam is a real threat to usefulness of the web. Spammers mask their content as useful or relevant content and hence are delivered to the user. The legitimate users consume this spam data considering it relevant to their information needs. Clay Shirky remarked that a communication channel isn't worth its salt until the spammers descend.

Spams are not easy to stop. For several years, email services like Gmail, Microsoft and others have been successfully detecting spam emails but still spam emails are in circle on the web. These services have been reporting that email spamming has been up to 90 to 95 percent of the total email exchanges. Even after successful detection of spams, companies are unable to stop spammers which ensures about the economic benefits spammers get when they trap a user clicking on a spam link. The severity of the threat posed by spamming has increased with the emergence of online social networks and twitter is one of the most popular online social networks which has been highly affected by spam [1].

Twitter spamming is more threatening because its more targeted towards the trending topics of the twitter and hence bit easier to get penetrated especially because of hash-tag operator. Another fact that makes twitter a rather easier and fruitful target for spammers is its variety of audience. twitter users span across all sectors of life i.e. it can be the teachers or students, celebrities or politicians, marketers or customers or even general public. They belong to all age groups but most widely age group that uses twitter is between 55 to 64 years.

There are about 60% users that access twitter from their cell phones. Twitter has 288 million monthly active members that make it widely growing social networking site. There are around 400 million tweets posted on daily bases, the average posts on twitter is 208 tweets per users account.

Due to this continuous distribution of information, a user faces many problems with search results that shares recurring and irrelevant information. This also can be very worrying at the times since a user has to scroll through the all information in direction to get an overall view of topic.

Spam detection on the twitter network is difficult due to the noticeable usage of URLs, abbreviations, informal language and modern language concepts. Old-style methods of detecting spam information fall short here. To date, study has been available on many techniques for detecting spams on twitter and blogs by using different features.

After knowing the existing importance of spams on twitter, we take inspiration or motivation from this user need and decided to design and develop improved techniques to detect spams on twitter. In this project, we propose a spam detection approach for detecting spam tweets. This approach is based on sentimental features of a tweet. The idea is to exploit the philosophy that spammer use to force a user to click on a particular link.

$$R(j) = \frac{n_i(j)}{n_i(j) + n_o(j)} \quad (1)$$

They definitely seek help of some motivational words (like 'the best web site', 'excellent service', etc) to make people believe in a certain tweet Results show that this exploitation of sentimental features proves fruitful. Another approach is discussed for spam detection in twitter network. They study the propagation of spam in the network. And they want to find out whether there is a pattern that spammers used for spam proliferation through the network and to determine whether the accounts are either been compromised or overtaken by spammers or certain accounts are purely created for spam activities in the network. They examine the characteristics of the graph of spam tweets and run Trust Rank technique on the collected data. Also introduced is the features for spam tweets detection without earlier statistics of the user and use statistical presentation for the analysis purpose of language to identify spam in twitter topics.

- **User-Based Features**

In Twitter, you can build your own social network by following friends and allowing others to follow you. Spam accounts try to follow large amount of users to gain their attention. The Twitter's spam and abuse policy [6] says that, "if you have a small number of followers compared to the amount of people you are following", then it may be considered as a spam account.

Three user-based features, namely the number of friends, the number of followers, and the reputation of a user are computed for spam detection in. The reputation of a user is defined in as where ni(j) represents the number of follower's user j has and no(j) represents the number of friends ("following") user j has. However, in our work, we only use the number of followers and the number of "following" as part of our user-based features.

- **Content-Based Features**

For content-based features, we use some obvious features e.g. the average length of a tweet. Additional content-based features are described in subsequent subsections.

  - **Number of URLs**

Since Twitter only allows a message with a maximum length of 140 characters, many URLs included in tweets are shortened URLS. Spammers often include shortened URLs in their tweets to entice legitimate users to access them. Twitter filters out the URLs linked to known malicious sites. However, shortened URLs can hide the source URLs and obscure the malicious sites behind them. While Twitter does not check these shorten URLs for malware, any user' updates that consist mainly of links are considered spam according to Twitter's policy. In the authors use

the percentage of tweets containing HTTP links in the user's 20 most recent tweets. If a tweet contains the sequence of characters "http;// or www., this tweet is considered containing a HTTP link. In our work, we use the number of HTTP links that are contained in a user's 100 most recent tweets [2].

2. LITERATURE SURVY

Prior to 2004, email spam classification research was suffering from considerable diversity and controversy in the methods employed for spam filtering and the ways by which these methods were evaluated. It was unclear, which method was best and showed promise for improvement. Three different communities were focussing on these issues (Lynam, 2009):

- The community of developers and practitioners with the motive of developing tools for instantaneous deployment;
- The community of spam filter vendors with the motive of selling spam filters;
- The community of researchers with the motive of inventing new facts and validating existing theories and algorithms.

Several spam filtering methods were experimented and investigated by users, practitioners, vendors and researchers and classified into three groups:

- Manual inspection,
- System oriented approaches,
- Content-based filtering.

Apart from all, Content-based filters can be further classified as -

- Ad-hoc Rule-based filters,
- Practical learning filters,
- Machine learning research.

Shen *et al.* investigated issues of detecting spammers on Twitter. The proposed method combines characteristics withdrawal from text content and information of social networks. The authors used matrix factorization to determine the underline feature matrix or the tweets and then came up with a social regularization with interaction coefficient to teach the factorization of the underline matrix. Subsequently, the authors combined knowledge with social regularization and factorization matrix processes, and performed experiments on the real-world Twitter dataset, i.e., UDI Twitter dataset.

Washha described the Hidden Markov Model for filtering the spam related to recent time. The method supports the accessible and obtainable information in the tweet object to recognize spam tweets and the tweets that are handled previously related to the same topic.

Jeong analyzed the follow spam on Twitter as an alternative of dispersion of provoking public messages, spammers follow authorized users, and followed by authorized users. Categorization techniques were proposed that are used for the detection of follow spammers. The focus of the social relation is cascaded and formulated into two mechanism, i.e., social status filtering and trade significance profile filtering, where each of which uses two-hop sub networks that are centered at each other.

Assemble techniques and cascading filtering are also proposed for combining the properties of both trade significance profile and social status. To check whether a user is fake or not, a two-hop social network for each user is focused to gather social information from social networks [3].

Meda presented a technique that utilizes a sampling of non-uniform features inside a machine learning system by the adaptation of random forest algorithm to recognize spammer insiders. The proposed framework focuses on the random forest and non-uniform feature sampling techniques [4].

The random forest is a learning algorithm for the categorization and regression that works by assembling several decision trees at preparation time and selecting the one with the majority votes by individual trees. The scheme integrates bootstrap aggregating technique with the un-planned selection of features.

3. EXISTING SYSTEM

In the existing system there is no filtering system based on a pre-processing schedule and on Naïve Bayes algorithm to discard the tweets containing inaccurate information,. Less security due No URL Based Spam Detection.

Disadvantages

- There is no filtering system based on a preprocessing schedule and on Naïve Bayes algorithm to discard the tweets containing inaccurate information,.
- Less security due No URL Based Spam Detection.

Problem Statement

Spamming is one of the major attacks that accumulate the large number of compromised machines by sending unwanted messages, viruses and phishing through emails. We have chosen this project because now days there are lot of people trying to fool you just by sending you fake e-mails like you have won 1000 dollars, this much amount is deposited in your account once you open this link then they will track you and try to hack your information. Sometimes relevant e-mails are considered as spam emails [4].

- Unwanted email irritating Internet consumers.
- Critical email messages are missed and/or delayed.
- Consumers change ISP's all the time looking for consistent email delivery.
- Loss of Internet performance and bandwidth.

SYSTEM DESIGN

Rational Rose

Rational Rose is object-oriented Unified Modeling Language (UML) software design tool intended for visual modeling and component construction of enterprise-level software applications. In much the same way a theatrical director blocks out a play, a software designer uses Rational Rose to visually create (model) the framework for an application by blocking out classes with actors (stick figures), use case elements (ovals), objects (rectangles) and messages/relationships (arrows) in a sequence diagram using drag- and-drop symbols.

Rational Rose documents the diagram as it is being constructed and then generates code in the designer's choice of C++ , Visual Basic , Java , Oracle8, Cobra or Data Definition Language [5].

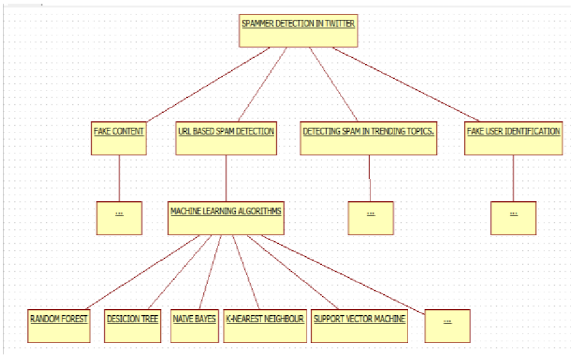


Figure.1: Overview of Implementing

Machine Learning Algorithms in Spammers Detection

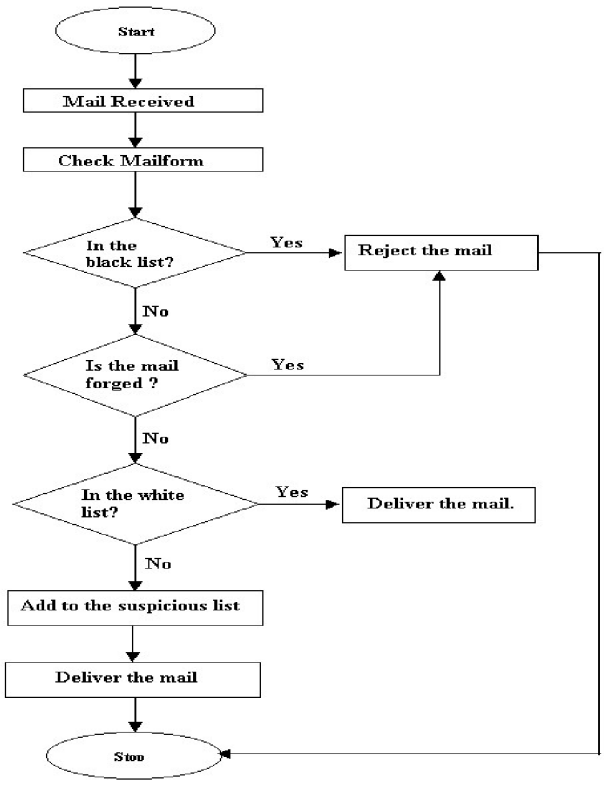


Figure.2: Flowchart of spam email filtering

4. PROPOSED SYSTEM

1) The average numbers of verified accounts that were either spam or non- spam and the number of followers of the user accounts.
2) The fake content propagation was identified through the metrics that include:
3) Social reputation,

    a) Global engagement,
    b) Topic engagement,
    c) Likability, and
    d) Credibility.

After that, the authors utilized regression prediction model to ensure the overall impact of people who spread the fake content at that time and also to predict the fake content growth in future.

Advantages

The average numbers of verified accounts that were either spam or non-spam and (ii) the number of followers of the user accounts.

The fake content propagation was identified through the metrics that include:

(i) Social reputation,
(ii) Global engagement,
(iii) Topic engagement,
(iv) Likability, and
(v) Credibility.

The Graphical Model

The naive Bayes classifier is a simple probabilistic model with strong independence assumptions. In particular, the underlying Bayesian network consists of a single node Y (whether the email is spam, for instance) that determines a set of features $F1,...,Fn$ independently (with Fi corresponding to the word at position i, for instance).

For example, if the first word is "Free", the email is more likely to be spam than if it is "Hey" (Or more accurately, spam is more likely to begin with "Free" than Hey) . The graphical model is below:

Various Modules

Module 1.Support Vector Machine (SVM): Support Vector Machine (SVM) is a supervised machine learning algorithm capable of performing classification. The linear SVM classifier works by drawing a straight line between two classes. All the data points that fall on one side of the line will be labelled as one class and all the points that fall on the other side will be labelled as the second.

The LSVM algorithm will select a line that not only separates the two classes but stays as far away from the closest samples as possible. In fact, the “support vector” in “support vector machine” refers to two position vectors drawn from the origin to the points which dictate the decision boundary [6].

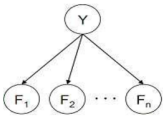


Figure.3: Graphical model

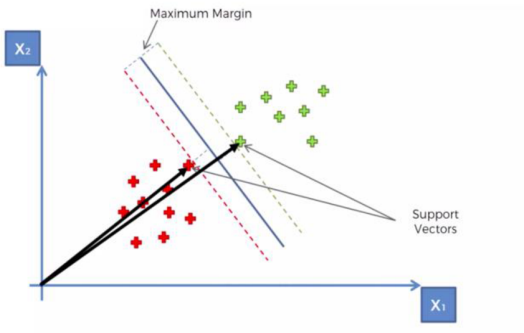


Figure.4: Graphical model

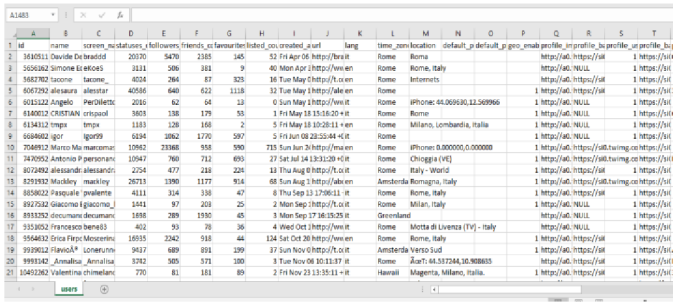
Here we import the above libraries and assign them a name for our comfort to implement in the algorithm We use certain functions in NN algorithm in order to provide the spammers accuracy effectively. The functions are:

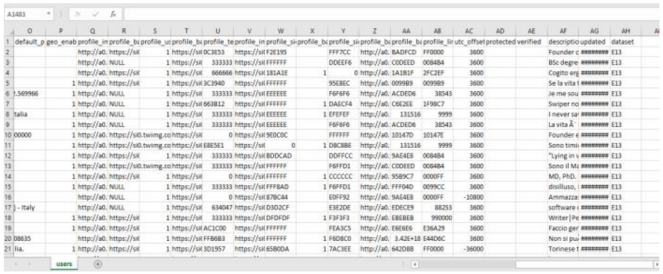
The entire neural network model is based on a layered architecture. Each layer has its own responsibility. These networks are designed to make use of layers of “neurons” to process raw data, find patterns into it and objects which are usually hidden to naked eyes

a) Function to read Datasets from CSV files.
b) Function for predicting Sex using name of the person.
c) Function from feature engineering.
d) Function for plot confusion curve.
e) Function for plotting ROC curve.
f) Function to train data using Neural Network.

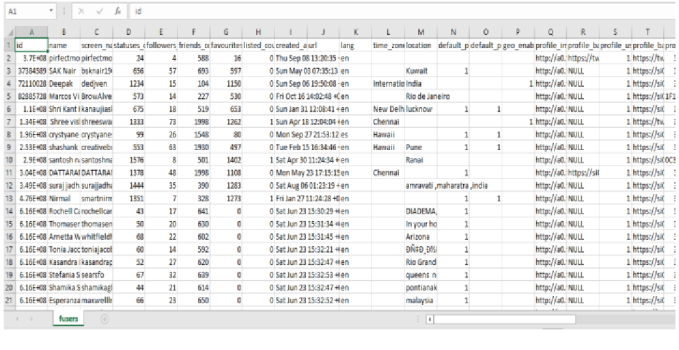
DATA SETS

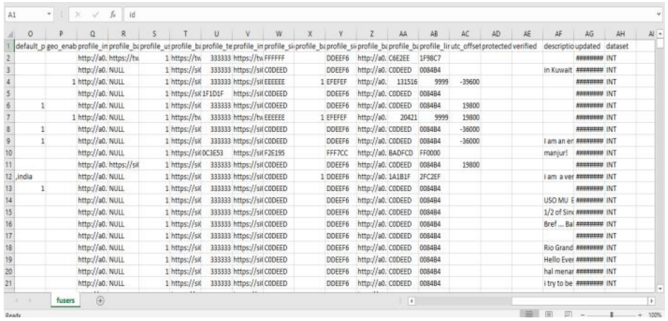
Users' data set:





Fake users data set:





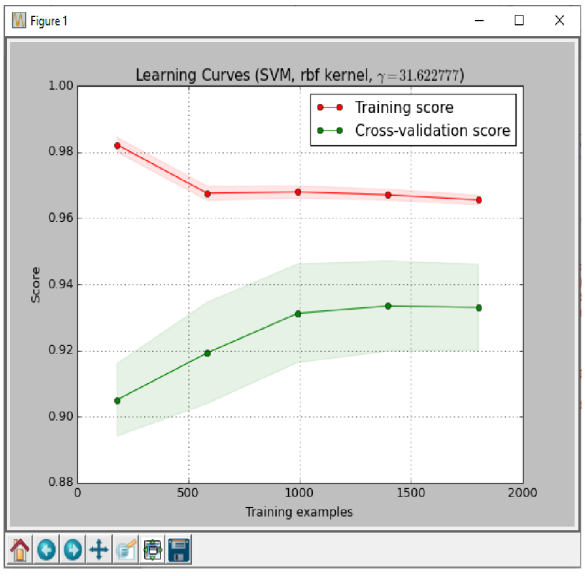
5. RESULTS

Support Vector Machine (SVM)

Output: 1

```
*Python 2.7.15 Shell*
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\Alekhya\Desktop\spammerdetection\SVM.py =========
reading datasets.....

extracting features.....

Index([u'statuses_count', u'followers_count', u'friends_count',
       u'favourites_count', u'listed_count', u'sex_code', u'lang_code'],
      dtype='object')
       statuses_count  followers_count  friends_count  favourites_count  \
count     2818.000000     2818.000000    2818.000000       2818.000000
mean      1672.198368      371.105039     395.363023        234.541164
std       4884.669157     8022.631339     465.694322       1445.847248
min          0.000000        0.000000       0.000000          0.000000
25%         35.000000       17.000000     168.000000          0.000000
50%         77.000000       26.000000     306.000000          0.000000
75%       1087.750000      111.000000     519.000000         37.000000
max      79876.000000   408372.000000   12773.000000      44349.000000

       listed_count     sex_code    lang_code
count   2818.000000  2818.000000  2818.000000
mean       2.818666    -0.180270     2.851313
std       23.480430     1.679125     1.992950
min        0.000000    -2.000000     0.000000
25%        0.000000    -2.000000     1.000000
50%        0.000000     0.000000     1.000000
75%        1.000000     2.000000     5.000000
max      744.000000     2.000000     7.000000
spliting datasets in train and test dataset...

training datasets.....

('The best classifier is: ', SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma=31.622776601683793,
  kernel='rbf', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False))
[0.91796009 0.92904656 0.92461197 0.9556541  0.93777778]
Estimated score: 0.93301 (+/- 0.00651)
```

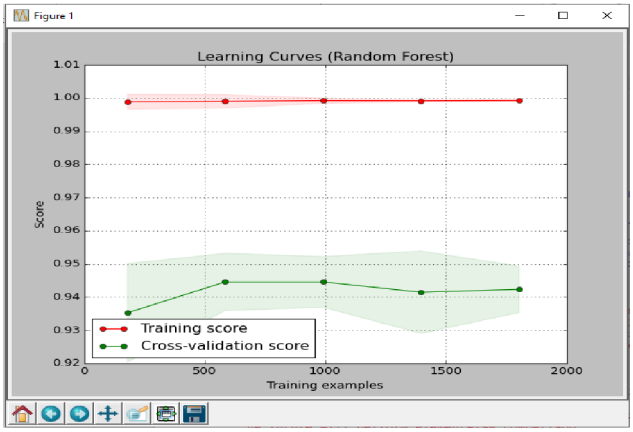
```
Warning (from warnings module):
  File "C:\Python27\lib\site-packages\sklearn\learning_curve.py", line 191
    if np.issubdtype(train_sizes_abs.dtype, np.float):
FutureWarning: Conversion of the second argument of issubdtype from `float` to `
np.floating` is deprecated. In future, it will be treated as `np.float64 == np.d
type(float).type`.

Warning (from warnings module):
  File "C:\Python27\lib\site-packages\matplotlib\collections.py", line 590
    if self._edgecolors == str('face'):
FutureWarning: elementwise comparison failed; returning scalar instead, but in t
he future will perform elementwise comparison
```



RF Algorithm

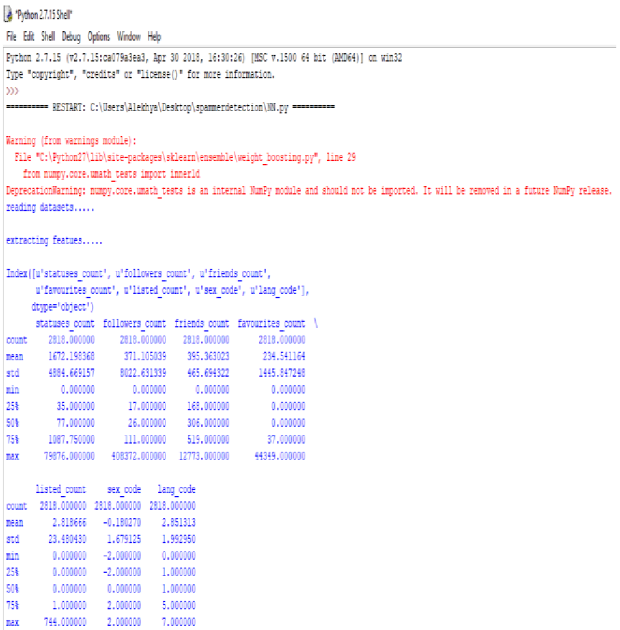
Output:

```
*Python 2.7.15 Shell*
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Alekhya\Desktop\spammerdetection\RF.py ==========

Warning (from warnings module):
  File "C:\Python27\lib\site-packages\sklearn\ensemble\weight_boosting.py", line
 29
    from numpy.core.umath_tests import inner1d
DeprecationWarning: numpy.core.umath_tests is an internal NumPy module and shoul
d not be imported. It will be removed in a future NumPy release.
reading datasets.....

extracting features.....

Index([u'statuses_count', u'followers_count', u'friends_count',
       u'favourites_count', u'listed_count', u'sex_code', u'lang_code'],
      dtype='object')
       statuses_count  followers_count  friends_count  favourites_count  \
count     2818.000000      2818.000000    2818.000000       2818.000000
mean      1672.198368       371.105039     395.363023        234.541164
std       4884.669157      8022.631339     465.694322       1445.847248
min          0.000000         0.000000       0.000000          0.000000
25%         35.000000        17.000000     168.000000          0.000000
50%         77.000000        26.000000     306.000000          0.000000
75%       1087.750000       111.000000     515.000000         37.000000
max      79876.000000    408372.000000   12773.000000      44349.000000

       listed_count     sex_code    lang_code
count   2818.000000  2818.000000  2818.000000
mean       2.818666    -0.180270     2.851313
std       23.480430     1.679125     1.992950
min        0.000000    -2.000000     0.000000
25%        0.000000    -2.000000     1.000000
50%        0.000000     0.000000     1.000000
75%        1.000000     2.000000     5.000000
max      744.000000     2.000000     7.000000
spliting datasets in train and test dataset...

training datasets.......

('The best classifier is: ', RandomForestClassifier(bootstrap=True, class_weight
=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=40, n_jobs=1,
            oob_score=True, random_state=None, verbose=0, warm_start=False))
[0.93569845 0.94235033 0.94235033 0.95343681 0.94222222]
Estimated score: 0.94321 (+/- 0.00286)

Warning (from warnings module):
  File "C:\Python27\lib\site-packages\sklearn\learning_curve.py", line 191
    if np.issubdtype(train_sizes_abs.dtype, np.float):
FutureWarning: Conversion of the second argument of issubdtype from `float` to `
np.floating` is deprecated. In future, it will be treated as `np.float64 == np.d
type(float).type`.

Warning (from warnings module):
  File "C:\Python27\lib\site-packages\matplotlib\collections.py", line 590
    if self._edgecolors == str('face'):
FutureWarning: elementwise comparison failed; returning scalar instead, but in t
he future will perform elementwise comparison
```

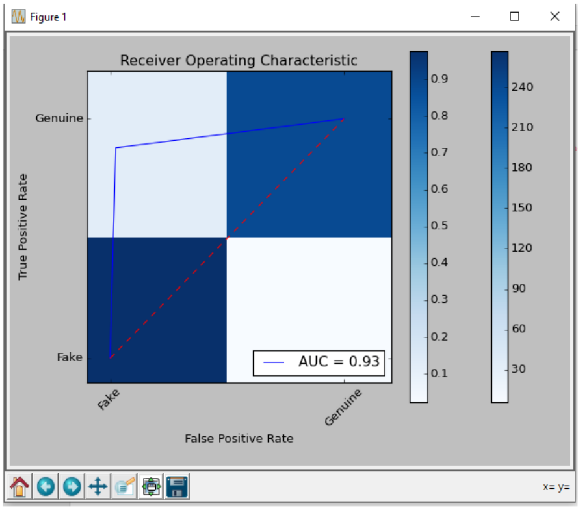


NN Algorithm

Output:



```
*Python 2.7.15 Shell*
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Alekhya\Desktop\spammerdetection\NN.py ==========

Warning (from warnings module):
  File "C:\Python27\lib\site-packages\sklearn\ensemble\weight_boosting.py", line 29
    from numpy.core.umath_tests import inner1d
DeprecationWarning: numpy.core.umath_tests is an internal NumPy module and should not be imported. It will be removed in a future NumPy release.
reading datasets.....

extracting features.....

Index([u'statuses_count', u'followers_count', u'friends_count',
       u'favourites_count', u'listed_count', u'sex_code', u'lang_code'],
      dtype='object')
       statuses_count  followers_count  friends_count  favourites_count  \
count     2818.000000      2818.000000    2818.000000       2818.000000
mean      1672.198368       371.105039     395.363023        234.541164
std       4884.669157      8022.631339     465.694322       1445.847248
min          0.000000         0.000000       0.000000          0.000000
25%         35.000000        17.000000     168.000000          0.000000
50%         77.000000        26.000000     306.000000          0.000000
75%       1087.750000       111.000000     515.000000         37.000000
max      79876.000000    408372.000000   12773.000000      44349.000000

       listed_count     sex_code    lang_code
count   2818.000000  2818.000000  2818.000000
mean       2.818666    -0.180270     2.851313
std       23.480430     1.679125     1.992950
min        0.000000    -2.000000     0.000000
25%        0.000000    -2.000000     1.000000
50%        0.000000     0.000000     1.000000
75%        1.000000     2.000000     5.000000
max      744.000000     2.000000     7.000000
training datasets.......

Classification Accuracy on Test dataset:  0.9296219658259326
Percent Error on Test dataset:  7.037655741741
Confusion matrix, without normalization
[[253   6]
 [ 97 267]]
Normalized confusion matrix
[[0.97633396 0.02316602]
 [0.12171053 0.87828947]]
              precision    recall  f1-score   support

       Fake       0.87      0.98      0.92       259
    Genuine       0.98      0.88      0.93       304

avg / total       0.93      0.92      0.92       563

roc_auc_score :  0.9275617252590938
False Positive rate:  [0.         0.02316602 1.        ]
True Positive rate:  [0.         0.87828947 1.        ]

Warning (from warnings module):
  File "C:\Python27\lib\site-packages\matplotlib\collections.py", line 590
    if self._edgecolors == str('face'):
FutureWarning: elementwise comparison failed; returning scalar instead, but in the future will perform elementwise comparison
```



6. CONCLUSION

In this paper, we have suggested some user-based and content-based features that can be used to distinguish between spammers and legitimate users on Twitter, a popular online social networking site. These suggested features are influenced by Twitter spam policies and our observations of spammers' behaviors. Then, we use these features to help identify spammers.

We evaluate the usefulness of these features in spammer detection using traditional classifiers like Random Forest, Naïve Bayesian, Support Vector Machine, K-NN neighbor schemes using the Twitter dataset we have collected.

Our results show that the Random Forest classifier gives the best performance. Using this classifier, our suggested features can achieve precision and F-measure as we have mentioned in the images. Based on our dataset, our features provide slightly better classification results. Our next step is to evaluate our detection scheme using larger Twitter dataset as well as possibly wall-post datasets from other online networking sites like Facebook. We also hope to include the content similarity metric in our near future work.

REFERENCES

[1] How to; 5 Top methods & applications to reduce Twitter Spam http://blog.thoughtpick.com/2009/07/how-to-5-topmethods-applications-to- reduce-twitter-spam.html

[2] Rish. "An empirical study of the naïve bayes classifier". Proeedings of IJCAI workshop on Empirical Methods in Artificial Intelligence, 2005.

[3] L. Bilge et al, "All your contacts are belong to us: automated identifty theft attacks on social networks", Proceedings of ACM World Wide Web Conference, 2009.

[4] T.N. Jagatic et al, "Social Phishing", Communications of ACM, Vol 50(10):94-100, 2007.

[5] S. Yardi et al, "Detecting Spam in a Twitter Network", First Monday, Vol 15(1), 2010.

[6] G. Stringhini, C. Kruegel, G. Vigna, "Detecting Spammers on Social Networks", Proceedings of ACM ACSAS'10, Dec, 2010.

Authors' Profiles:

Dr. MOLLI SRINIVASA RAO received his Ph.D. award (2018) in Computer Science and Systems Engineering and M.Tech in Computer Science and Technology (2003) from Andhra University, Andhra Pradesh and B.Tech (2001) from CBIT affiliated to Osmaina University, Hyderabad.

He has more than 17 years' good teaching and research experience. He is working as Professor and HOD, Department of CSE in Viswanadha Institute of Technology and Management, Visakhapatnam, Andhra Pradesh. His research interest includes Mobile Ad-hoc Networks, Soft Computing Techniques; Cloud computing, Artificial Intelligence, Machine Learning and Block Chain.

He has Published 15 research papers in various reputed SCI/Scopus/ UGC approved International and national Journals and conferences and 7 international / national Workshops, 10 seminars and 4 FDP.