Alliance College Of Engineering and Design

ALLIANCE UNIVERSITY
Private University established in Karnataka State by Act No.34 of year 2010
Recognized by the University Grants Commission (UGC), New Delhi

Celebrating 25 years SILVER JUBILEE of Alliance Education

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-INFORMATION TECHNOLOGY

Batch: 2019-2023

Detection of Fake and Clone accounts in Twitter using Classification and Distance Measure Algorithms

By

Name : Jallepalli Harsha Vardhan

Registration no : 19030141CSE047

Semester : VIth "A "Section

CS-603 MACHINE LEARNING PROJECT

ALLIANCE UNIVERSITY

Private University established in Karnataka State by Act No.34 of year 2010
Recognized by the University Grants Commission (UGC), New Delhi

Celebrating 25 years SILVER JUBILEE of Alliance Education

CERTIFICATE

This is to certify that **Jallepalli Harsha Vardhan**, Regn-No: **19030141CSE097** Of Batch 2019-2023 has completed the report titled “Detection of Fake and Clone accounts in Twitter using Classification and Distance Measure Algorithms”, for the partial fulfilment of the Course: Machine Learning in Semester VI of the Bachelor of Technology in Computer Science.

INTERNAL GUIDE: **HEAD OF THE DEPARTMENT:**

Dr. Shekhar. R **Dr. Abraham George**

ALLIANCE UNIVERSITY

Private University established in Karnataka State by Act No.34 of year 2010
Recognized by the University Grants Commission (UGC), New Delhi

Celebrating 25 years SILVER JUBILEE of Alliance Education

DECLARATION

This is to declare that the report titled “Machine Learning with Iris classification analysis with flowers” has **been** made for the partial fulfilment of the Course Bachelor of Technology in Computer Science and Engineering, under the guidance of **Dr. Shekhar R.**

I confirm that this report truly represents my work undertaken as a part of my project work. This work is not a replication of work done previously by any other person. I also confirm that the contents of the report and the views contained therein have been discussed and deliberated with the faculty guide.

NAME	REG NO	SIGNATURE
Jallepalli Harsha Vardhan	19030141CSE047	

TABLE OF CONTENTS

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER-1

INTRODUCTION:

Online Social Networks (OSN) such as Facebook, Twitter, LinkedIn, and Instagram are utilised by billions of people worldwide to develop network connections. The simplicity and accessibility of social networks have ushered in a new era of networking. OSN members post a variety of information on the network, including images, videos, school and college names, phone numbers, email addresses, home addresses, family relationships, bank information, and professional information. If this information falls into the hands of attackers, the consequences will be serious.

If the victims are minors, the hazards are multiplied. The profile information of current users is taken in a Profile Cloning attack to generate duplicate profiles, which are then exploited to compromise the identity of the original profile owners. Profile cloning is classified into two types: same site profile cloning and cross site profile cloning.

When user credentials from one network are used to produce a clone profile on the same network, this is referred to as Same Site profile cloning. Cross Site profile cloning occurs when an attacker uses user information from one network to generate a duplicate profile in another network where the user does not have an account.

In order to attract more users, the registration procedure in social networks has become quite easy and Fake profiles are likewise being created at an alarming pace. An attacker establishes a bogus profile in order to connect with a victim and carry out harmful operations. Furthermore, it is used to disseminate bogus news and spam communications.

OBJECTIVE:

In recent years, major social networks such as Facebook and Twitter have acknowledged that fraudulent and duplicate accounts exist on their platforms. Their authors can use these identities to spread false information, promote or criticise a concept, a product, or an election candidate, persuading physical network users to make a choice. They leverage the implicit belief ties between users to fulfil their malicious goals, such as creating hateful links within posts/tweets.

We employ various new characteristics to recognise Twitter accounts that are more effective and resilient than previously used features (e.g. amount of Users/followings/followers, etc.).

We assessed the suggested set of characteristics using two widely used machine learning classification techniques, Support Vector Machine (SVM) and Neural Networks (NN). Their adulation has resulted in a variety of issues, including the creation of phoney accounts and the propagation of false information, as well as the fabrication of harmful content. Such scenarios have the potential to harm real-world occurrences that are directly connected to individuals, business entities, learning sectors, and so on. In this study, we show our technique for detecting bogus users of the Twitter social network.

COLLECTING THE DATASET:

We ran our experiment on a collection of Twitter accounts compiled by "the Fake project." We contacted the authors of "the Fake project" who established it. The authors stated in their technical report that the dataset was gathered from many sources, the first of which was the #elezioni2013 dataset, which contained 1481 validated human accounts. A further 469 verified human accounts were gathered by the "Fake Project" team. The phoney accounts were obtained from three different sources; they purchased 1000 false accounts from http://fastfollowerz.com, 1000 from http://intertwitter.com, and 1000 from http://twittertechnology.com for $19, $14, and $13, respectively. In summary, the dataset that is used in this paper consists of 1481 human accounts and 3000 fake accounts.

STATE OF ART:

The proposed method is applied on Twitter account data sets and classifies the dataset into four classes. In this case, the network could select the good features and extract a small but adequate set of rules for the classification task. For Class one data set we obtained zero misclassification on test sets and for all other data sets the results obtained are comparable to the results reported in the literature.

It is observed from the literature survey that the existing algorithms face several difficulties like the computational power is increases when run Deep Learning on latest computation, requires a large amount of data, is extremely computationally expensive to train, they do not have explanatory power that is they may extract the best signals to accurately classify and cluster data, but cannot get how they reached a certain conclusion. Neural Networks cannot be retrained that is it is impossible to add data later. To address these problems the current work is taken up to develop a new

WORK DONE:

Various studies have been presented to detect fake accounts using various methodologies. The feature-based detection strategy will be used in this study. This method is based on tracking the user's activity, such as the amount of tweets, retweets, friends, and so on. This approach is founded on the belief that humans behave differently than fakes, and so identifying this behaviour will lead to the discovery of the fake accounts. In this part, we will show some of the works that have been shown in this region.

Reference has reached an accuracy of 84.5% to detect spammers by identifying 23 attributes, most of these attributes(17 attributes) are demonstrated in Table I which was mentioned in his research. However, in our research, we have reached more accuracy with smaller set of attributes as will be discussed in Section III. In [9], the set of attributes has been minimized by identifying ten attributes for detection, the attributes are presented in Table I. However, as mentioned in this research, the result was not promising for identifying fake accounts with more optimistic perspective that it is able to identify fake tweets with higher accuracy by the support of graph techniques. Although [10] has presented a minimized set of attributes which contained six attributes, however, it is mentioned that it could only detects determined types of spammers, they are bagger, and poster spammers. In our approach, we propose minimized set of attributes for detecting all types of spammers. In addition, one of these attributes requires text analysis procedure for finding the similarities among messages which is not required for our proposed approach.

TRAINING:

We use the dataset to train our model to predict output accurately. We focus on classifying the fake accounts and safe accounts of data from this dataset. The data provided is processed in a way of analysing every parameter. In the machine learning process, data pre-processing is required so that the data gets converted to such a form that the machine can now easily understand it. We can say that the data features can now be easily interpreted by the algorithm. In our project the data gets converted to binary format. Now, the algorithm does the necessary computation. The output is converted from binary to hexadecimal format so that we can understand it. The task is to classify the fake accounts from all accounts.

TESTING:

We use the Support vector classifier in our code to classify the testing data. After using the Support vector machine, we have found out that its accuracy is very high. We use the SUPPORT VECTOR MACHINE and NEURAL NETWORKS to get the fake accounts for best accuracy.

CHAPTER - 2

ANALYSIS:

The working plan that has been performed to detect the required features' set are described in steps, steps from 1 to 19 present the working plan in details

1. A dataset has been prepared for our experiment. Section III discusses the source and the steps of preparing of the used dataset

2. We have performed a survey that defined different sets of features. We have collected all the features that is proposed by these researchers

3. 22 attributes have been collected as a result of our extensive research. Section III discusses how these attributes have been collected

4. We have performed experiments based on different perspectives, in the first set of experiments, five of the most successful classification algorithms have been applied on the dataset using these 22 attributes. Section III discusses these five algorithms in brief

5. A 5-fold cross validation experiments for the five classification algorithms using the 22 attributes, and the results have been compared

6. We have selected 19 of these attributes according to our point of view and applied the classification algorithms in the second set of experiment

7. A 5-fold cross validation experiments for the five classification algorithms using the 19 attributes, and the results have been compared

8. For the target to get the minimized set of attributes with best classification results, we have applied the GAIN Measure to find a weight for all the 22 attributes. Section

III discusses how the GAIN measure has been applied to the 22 attributes and the weights for the attributes.

9. Using the calculated weighting for the attributes, in the third set of experiments, the five classification algorithms have been applied again on the dataset using the 22 attributes.

10. Using the calculated weighting for the 19 attributes, in the sixth set of experiments, the five classification algorithms have been applied again on the dataset using the attributes that had weight of equal or above 50%. The set of attributes has been minimized to six attributes.

17. A 5-fold cross validation experiments for the five classification algorithms, and the results have been compared

18. According to the previously applied experiments, we have reached the minimum set of attributes with maximum classification results, with selecting the best classification algorithm for the twitter account.

19. This final result and the final set of attributes that we have reached according to the performed experiments. This set of results is the minimum set of attributes that is able to discover the fake accounts from Twitter with maximum accuracy which reaches more than 99%.

CHAPTER - 3

DESIGN AND IMPLEMENTATION:

This module is used to detect fake Twitter profiles. Here fake profiles are detected based on rules that effectively distinguish fake profiles from genuine ones. Some of the rules that are used to detect fake profiles are - usually fake profiles do not have profile name or image. They do not include any description about the account. The geo-enabled field will be false as they do not want to expose their location in tweets.

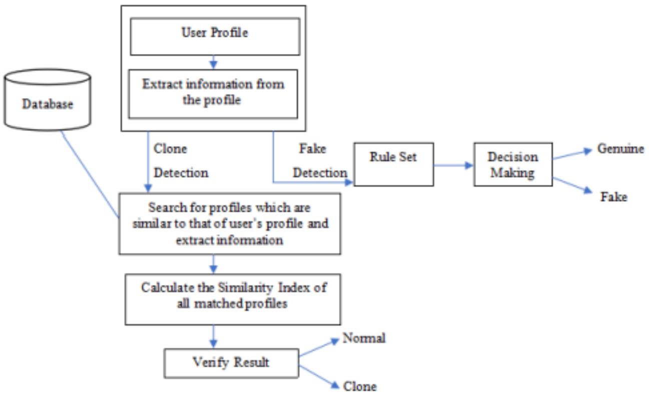

User Profile
Extract information from the profile
Database
Clone
Detection
Fake
Detection
Rule Set
Decision Making
Genuine
Fake
Search for profiles which are similar to that of user's profile and extract information
Calculate the Similarity Index of all matched profiles
Verify Result
Normal
Clone

Fig. 1. Architecture of proposed system.

They usually make large number of tweets or sometimes the profiles would not have made any tweets etc.

SETUP VIRTUAL ENVIRONMENT OR START JUPYTER NOTEBOOK:

Virtual environment helps to keep dependencies between different projects. Its main purpose is to create an isolated environment for python projects. To setup virtual environment we have to follow some steps that are:

1. Open the terminal

2. Setup the pip package manager

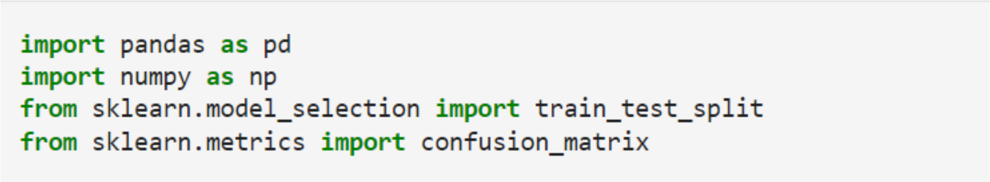
3. Create the virtual environment

4. Activate the virtual environment

5. After that install the appropriate libraries (NumPy, pandas, etc.) using pip

IMPORTING LIBRARIES AND DOWNLOAD THE DATA

The following libraries are required for this project:



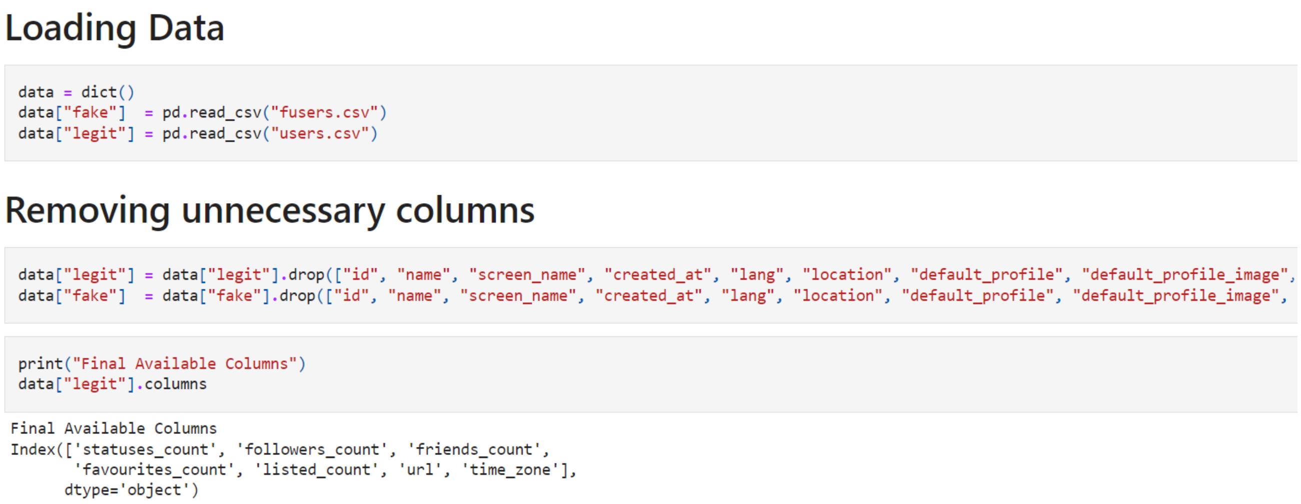
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

LIBRARIES IMPORTED:

Here, we are importing NumPy, pandas and sklearn libraries. Where, NumPy is an array processing package which is used in scientific computing with arrays. Pandas is built on the NumPy package and its key data structure is called the Data Frame. Data Frames allow us to store and manipulate tabular data in rows of observations and columns of variables. The Twitter accounts dataset can be downloaded from the UCI Machine Learning Repository. Characteristics of data set is multivariate. The dataset downloaded from the UCI Machine Learning Repository is in the form of CSV (Comma Separated Values) file and the file name is ‘Twitter data’ and save the file in the same directory as our project contains.

Now we are going to move into data exploration as well as analysis using the iris data. Let’s import our data set using 'pandas' library, which will convert our data into the tabular format from the CSV format. The beauty of using pandas library is just that we can read the csv files. For converted our data into the understandable format we have to add column to the imported dataset which contain the attributes (following, followers, likes, retweets), it gives heading for the imported data.

Dataset using pandas library:



Loading Data

```
data = dict()
data["fake"]  = pd.read_csv("fusers.csv")
data["legit"] = pd.read_csv("users.csv")
```

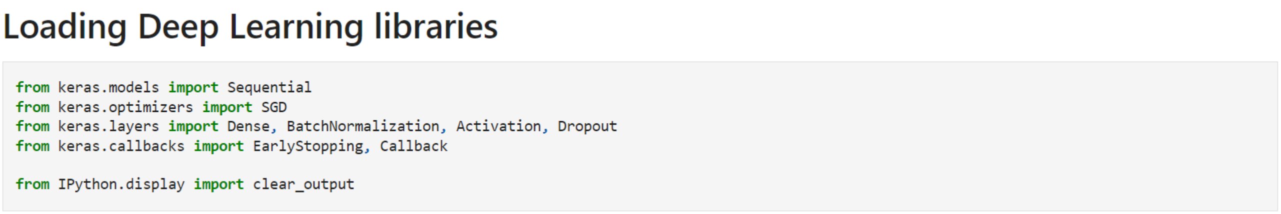
Removing unnecessary columns

```
data["legit"] = data["legit"].drop(["id", "name", "screen_name", "created_at", "lang", "location", "default_profile", "default_profile_image",
data["fake"]  = data["fake"].drop(["id", "name", "screen_name", "created_at", "lang", "location", "default_profile", "default_profile_image",
```

```
print("Final Available Columns")
data["legit"].columns
```

```
Final Available Columns
Index(['statuses_count', 'followers_count', 'friends_count',
       'favourites_count', 'listed_count', 'url', 'time_zone'],
      dtype='object')
```

Dataset using Keras library:



Loading Deep Learning libraries

```
from keras.models import Sequential
from keras.optimizers import SGD
from keras.layers import Dense, BatchNormalization, Activation, Dropout
from keras.callbacks import EarlyStopping, Callback

from IPython.display import clear_output
```

IMPLEMENTATION OF ALGORITHMS:

Neural Network

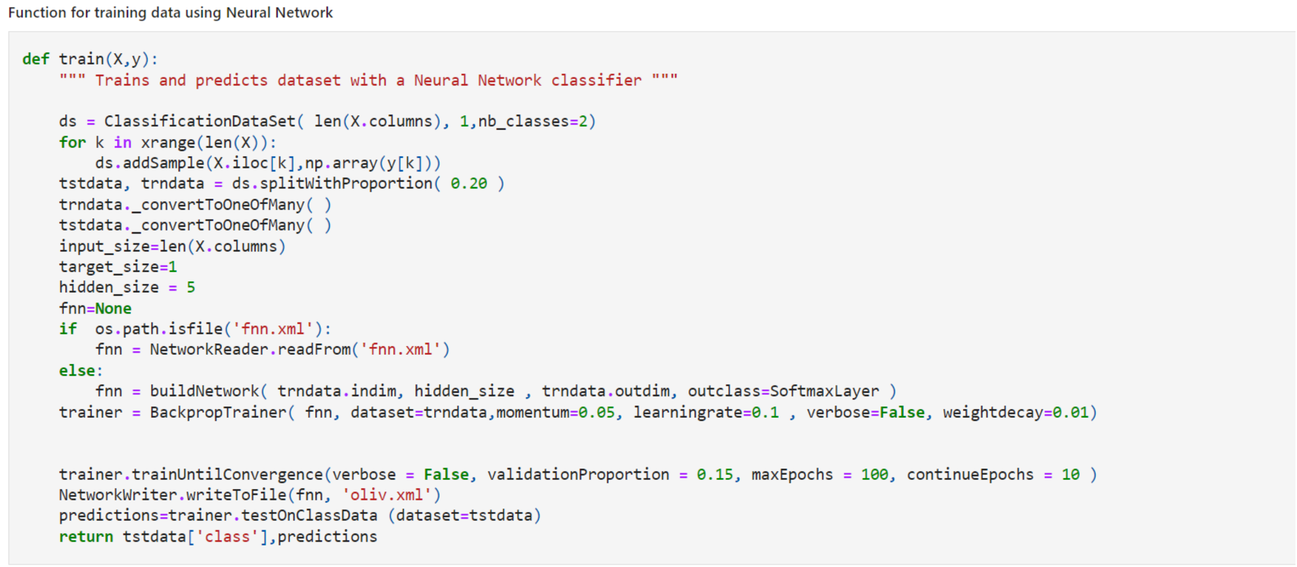
An artificial neural network learning algorithm, or neural network, or just neural net, is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output, usually in another form. The concept of the artificial neural network was inspired by human biology and the way neurons of the human brain function together to understand inputs from human senses. Neural networks are just one of many tools and approaches used in machine learning algorithms. The neural network itself may be used as a piece in many different machine learning algorithms to process complex data inputs into a space that computers can understand. Neural networks are being applied to many real-life problems today, including speech and image recognition, spam email filtering, finance, and medical diagnosis, to name a few. Machine learning algorithms that use neural networks generally do not need to be programmed with specific rules that define what to expect from the input. The neural net learning algorithm instead learns from processing many labelled examples (i.e. data with with "answers") that are supplied during training and using this answer key to learn what characteristics of the input are needed to construct the correct output. Once a sufficient number of examples have been processed, the neural network can begin to process new, unseen inputs and successfully return accurate results. The more examples and variety of inputs the program sees, the more accurate the results typically become because the program learns with experience.

This concept can best be understood with an example. Imagine the "simple" problem of trying to determine whether or not an image contains a cat. While this is rather easy for a human to figure out, it is much more difficult to train a computer to identify a cat in an image using classical methods. Considering the diverse possibilities of how a cat may look in a picture, writing code to account for every scenario is almost impossible. But using machine learning, and more specifically neural networks, the program can use a generalized approach to understanding the content in an image. Using several layers of functions to decompose the image into data points and information that a computer can use, the neural network can start to identify trends that exist across the many, many examples that it processes and classify images by their similarities.

Neural networks can be applied to a broad range of problems and can assess many different types of input, including images, videos, files, databases, and more. They also do not require explicit programming to interpret the content of those inputs.

Because of the generalized approach to problem solving that neural network offer, there is virtually no limit to the areas that this technique can be applied.

TRAINING MODEL USING NEURAL NETWORKS:



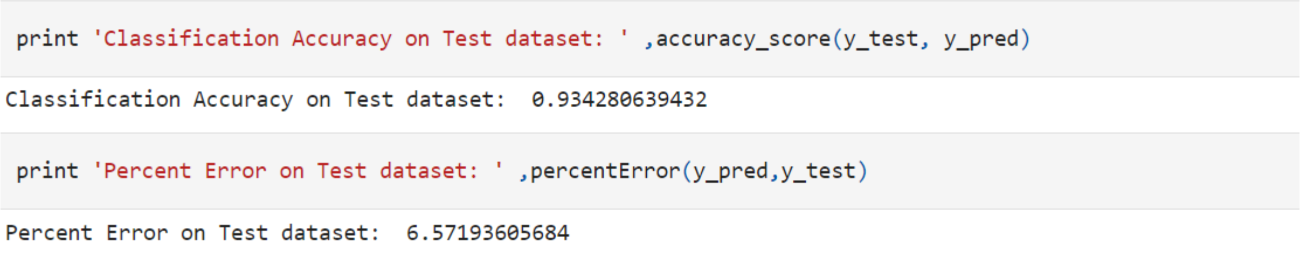
Function for training data using Neural Network

```
def train(X,y):
    """ Trains and predicts dataset with a Neural Network classifier """

    ds = ClassificationDataSet( len(X.columns), 1,nb_classes=2)
    for k in xrange(len(X)):
        ds.addSample(X.iloc[k],np.array(y[k]))
    tstdata, trndata = ds.splitWithProportion( 0.20 )
    trndata._convertToOneOfMany( )
    tstdata._convertToOneOfMany( )
    input_size=len(X.columns)
    target_size=1
    hidden_size = 5
    fnn=None
    if  os.path.isfile('fnn.xml'):
        fnn = NetworkReader.readFrom('fnn.xml')
    else:
        fnn = buildNetwork( trndata.indim, hidden_size , trndata.outdim, outclass=SoftmaxLayer )
    trainer = BackpropTrainer( fnn, dataset=trndata,momentum=0.05, learningrate=0.1 , verbose=False, weightdecay=0.01)


    trainer.trainUntilConvergence(verbose = False, validationProportion = 0.15, maxEpochs = 100, continueEpochs = 10 )
    NetworkWriter.writeToFile(fnn, 'oliv.xml')
    predictions=trainer.testOnClassData (dataset=tstdata)
    return tstdata['class'],predictions
```

CLASSIFICATION ACCURACY ON TEST DATASET:



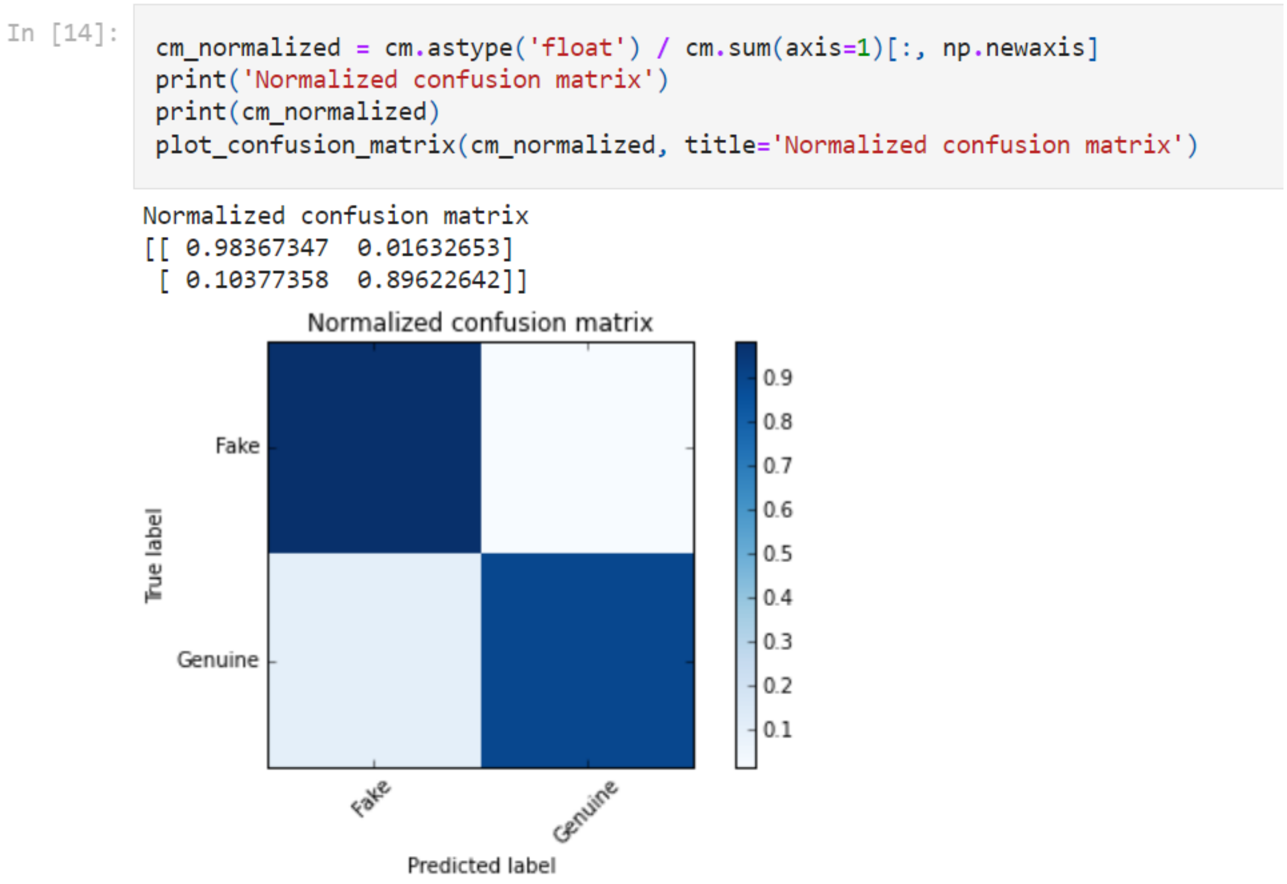
```
print 'Classification Accuracy on Test dataset: ' ,accuracy_score(y_test, y_pred)
```

```
Classification Accuracy on Test dataset:  0.934280639432
```

```
print 'Percent Error on Test dataset: ' ,percentError(y_pred,y_test)
```

```
Percent Error on Test dataset:  6.57193605684
```

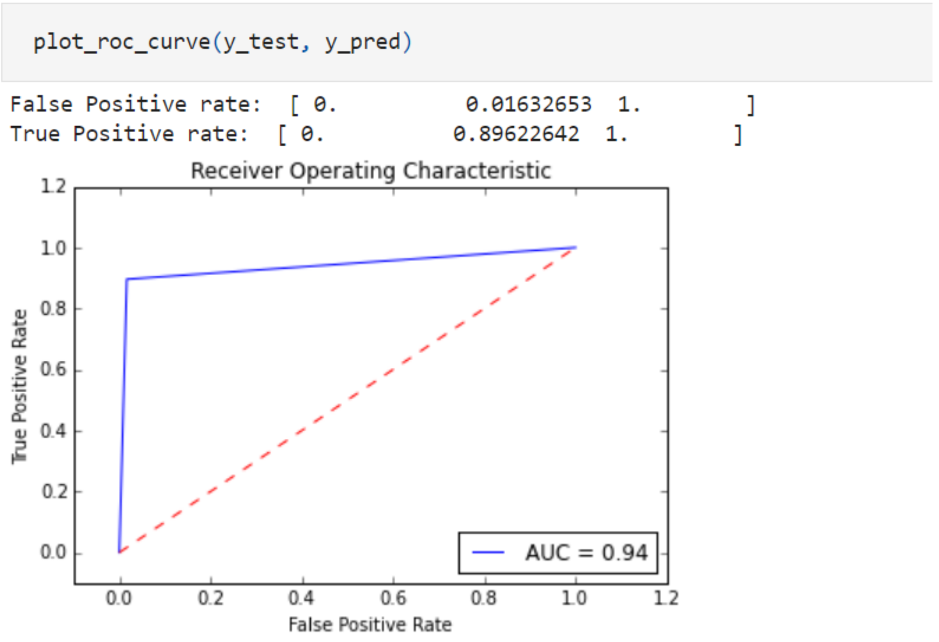
NORMALISED CONFUSION MATRIX:



```
In [14]:
cm_normalized = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
print('Normalized confusion matrix')
print(cm_normalized)
plot_confusion_matrix(cm_normalized, title='Normalized confusion matrix')
```

```
Normalized confusion matrix
[[ 0.98367347  0.01632653]
 [ 0.10377358  0.89622642]]
```

FALSE POSITIVE RATE AND TRUE POSITIVE RATE CURVE:



```
plot_roc_curve(y_test, y_pred)
```

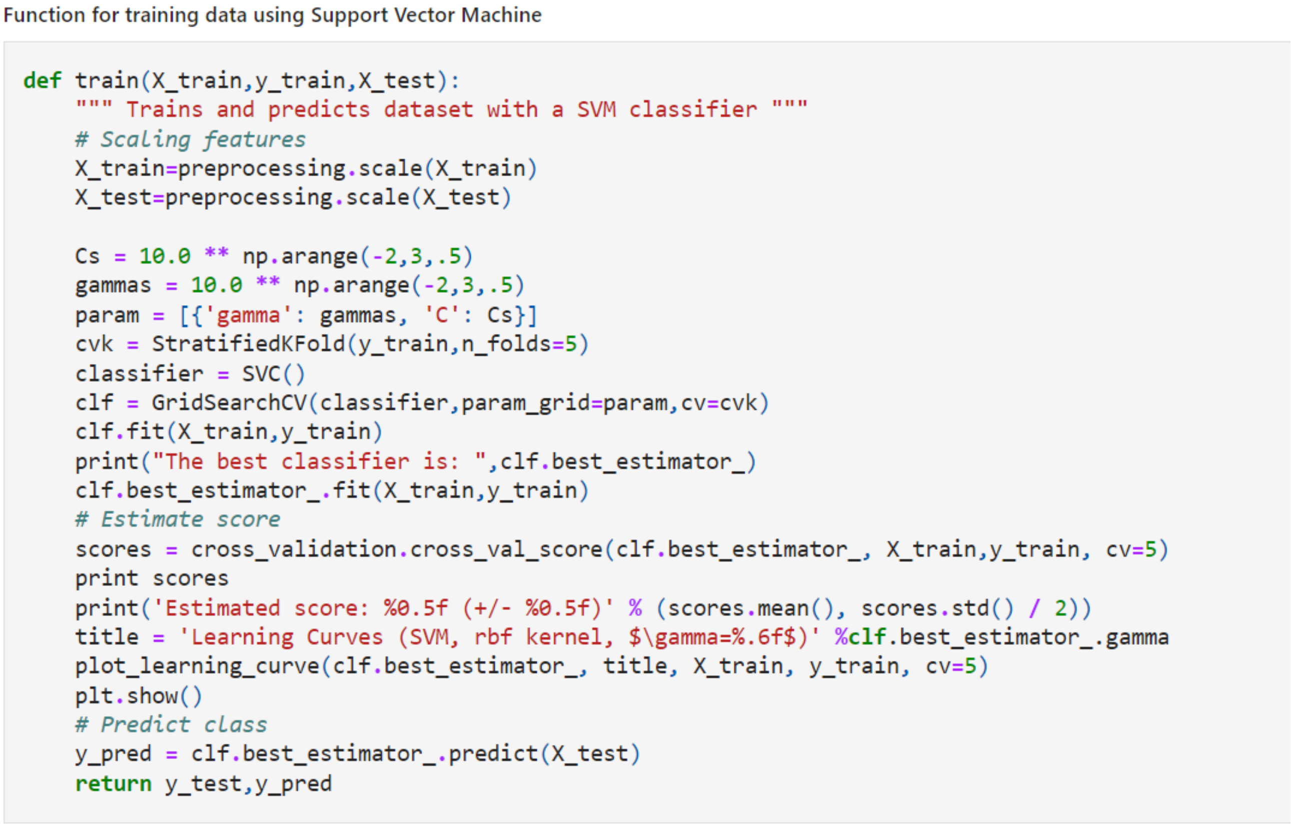
```
False Positive rate:  [ 0.          0.01632653  1.        ]
True Positive rate:  [ 0.          0.89622642  1.        ]
```

SUPPORT VECTOR MACHINE(SVM):

A support vector machine abbreviated as SVM was first introduced in the 1960's and late an improvised in the 1990's. SVM is supervised learning machine learning classification algorithm that has become extremely popular nowadays owing to its extremely efficient results so SVM is implemented in a slightly differently than other machine learning algorithms it is capable of performing classification and regression and outlier detection as well. Support vector machine is a discriminative classifier that

is formally designed by a separative hyper plane. It is a representation of examples as points in space that are mapped so that the points of different categories are separated by a gap as wide as possible. SVM does not directly provide probability estimates these are calculated using five-fold cross validation. Five-fold cross validation means the dataset will divided randomly into 5 subsets and then take a subset for use as a test set and use remaining subgroups as a training set, then fit a model on the training set and evaluate the score, this will happen until each subset one-by-one considered as a test-set. The main objective of support vector machine is to separate the given data of different classes by a line (hyper plane) in the best possible way. The nearest points of classes from the hyper plane is known by support vectors. There can be many hyper planes that will separate the classes, so to choose the appropriate hyper plane SVM algorithm find the nearest points to the hyper plane of both the classes and check the distance between the hyper plane and support vectors. Here, this distance is known by margin. SVM algorithm selects the hyper plane which gives the maximum margin. 27 In the below figure in Graph-A we can see that there are two lines blue and green. It is clearly seen that blue is placed in the space in which the support vectors give maximum margin from the blue line there for blue line will selected as a hyper plane.

TRAINING MODEL USING SUPPORT VECTOR MACHINE:

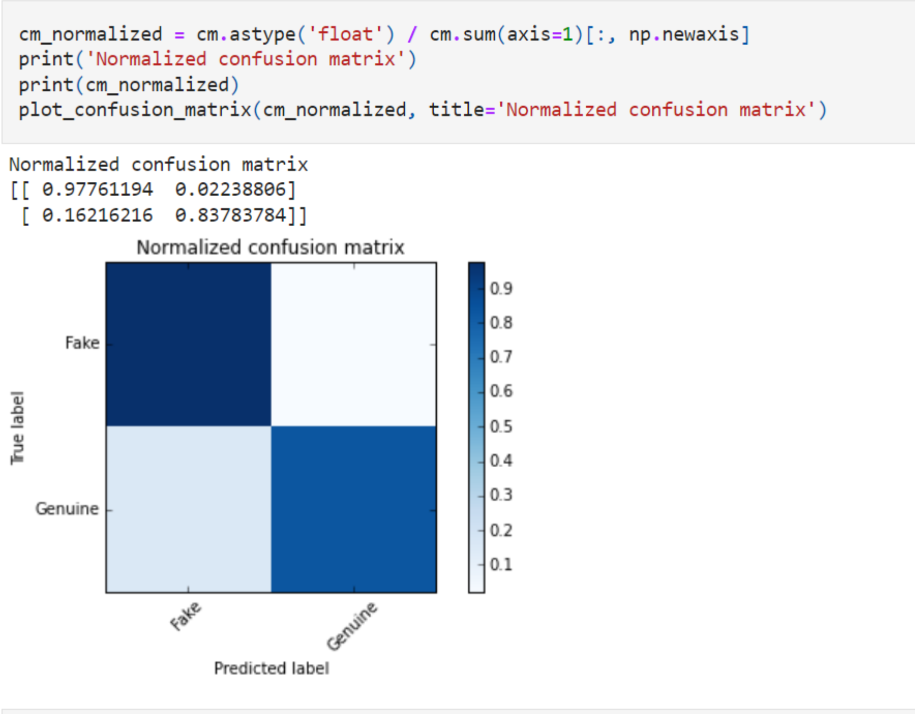


Function for training data using Support Vector Machine

```
def train(X_train,y_train,X_test):
    """ Trains and predicts dataset with a SVM classifier """
    # Scaling features
    X_train=preprocessing.scale(X_train)
    X_test=preprocessing.scale(X_test)

    Cs = 10.0 ** np.arange(-2,3,.5)
    gammas = 10.0 ** np.arange(-2,3,.5)
    param = [{'gamma': gammas, 'C': Cs}]
    cvk = StratifiedKFold(y_train,n_folds=5)
    classifier = SVC()
    clf = GridSearchCV(classifier,param_grid=param,cv=cvk)
    clf.fit(X_train,y_train)
    print("The best classifier is: ",clf.best_estimator_)
    clf.best_estimator_.fit(X_train,y_train)
    # Estimate score
    scores = cross_validation.cross_val_score(clf.best_estimator_, X_train,y_train, cv=5)
    print scores
    print('Estimated score: %0.5f (+/- %0.5f)' % (scores.mean(), scores.std() / 2))
    title = 'Learning Curves (SVM, rbf kernel, $\gamma=%.6f$)' %clf.best_estimator_.gamma
    plot_learning_curve(clf.best_estimator_, title, X_train, y_train, cv=5)
    plt.show()
    # Predict class
    y_pred = clf.best_estimator_.predict(X_test)
    return y_test,y_pred
```

CONFUSION MATRIX WITH NORMALIZATION:



```
cm_normalized = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
print('Normalized confusion matrix')
print(cm_normalized)
plot_confusion_matrix(cm_normalized, title='Normalized confusion matrix')
```

```
Normalized confusion matrix
[[ 0.97761194  0.02238806]
 [ 0.16216216  0.83783784]]
```

CHAPTER-4

In the previous section what we gone through is the implementation of all the data where we did some preliminary analysis of the data and get a few of it, but to progress further and to dive into the data a little bit more we are going to do some visualization. Visualization is a great way to develop a better understanding of your data and python and has a lot of great tools for specifically that purpose

RESULTS AND DISCUSSION:

A. Dataset Used

The datasets used in the experiment are collected from MIB projects. It consists of Genuine and Fake Twitter datasets. The Genuine accounts dataset contains accounts of people who came forward to be part of academic study for detecting fake accounts on Twitter and it is mostly a mixture of accounts of researchers, social experts and journalists from Italy, US and other European countries[4]. The fake accounts were purchased from three different Twitter online markets namely fastfollowerz.com, intertwitter.com and twittertechnology.com [4].

B. Evaluation Metrics

In order to evaluate the performance of the system, various evaluation metrics are used based on following four standard indicators

• True Positive (TP): True positives are records that are correctly detected with expected vectors.

• True Negative (TN): True negatives are records correctly detected expected as Neutral. • False Positive (FP): False positives are records that were detected by the system as expected but actually are listed in the other vectors.

• False Negative (FN): False negatives are records not detected by the system.

The evaluation metrics considered are

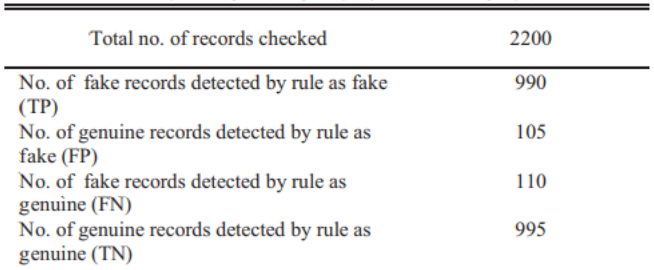
1. Accuracy which gives the ratio of number of correct results to the total number of inputs

2. Precision which gives the proportion of positive detection that was actually correct

3. Recall which gives the proportion of actual positives that was detected correctly

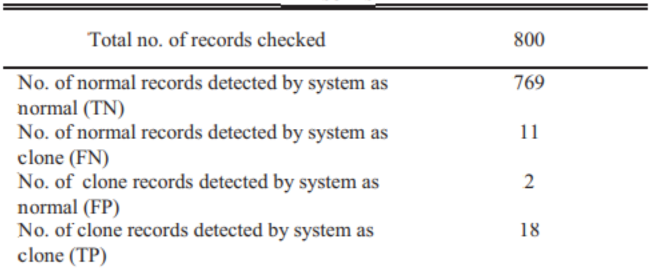
4. F1 Score which takes into account both precision and recall to compute the score. F1-score is given by harmonic mean of precision and recall. If F1-score is 1, then it is best value and worst is 0.

TABLE I
PERFORMANCE EVALUATION OF FAKE DETECTION



Total no. of records checked	2200
No. of fake records detected by rule as fake (TP)	990
No. of genuine records detected by rule as fake (FP)	105
No. of fake records detected by rule as genuine (FN)	110
No. of genuine records detected by rule as genuine (TN)	995

TABLE II
PERFORMANCE EVALUATION OF CLONE DETECTION USING SIMILARITY MEASURES



Total no. of records checked	800
No. of normal records detected by system as normal (TN)	769
No. of normal records detected by system as clone (FN)	11
No. of clone records detected by system as normal (FP)	2
No. of clone records detected by system as clone (TP)	18

For detection of fake profiles, a total of 2200 accounts were fed into the system in which 1100 were genuine and 1100 were fake. The rule set worked fine and was able to classify genuine and fake accounts with an accuracy of 90.2% shown in Fig. 2. Table I gives the performance evaluation of fake detection module. For detection of clone profiles, 780 normal profiles along with 20 artificially generated clone profiles were fed to the modules to check how accurately it detects clone profiles from the given set. The modules worked fine and was able to detect clones with good accuracy. Table II and Table III gives the performance evaluation of Clone Detection using Similarity Measures and using C4.5 respectively.

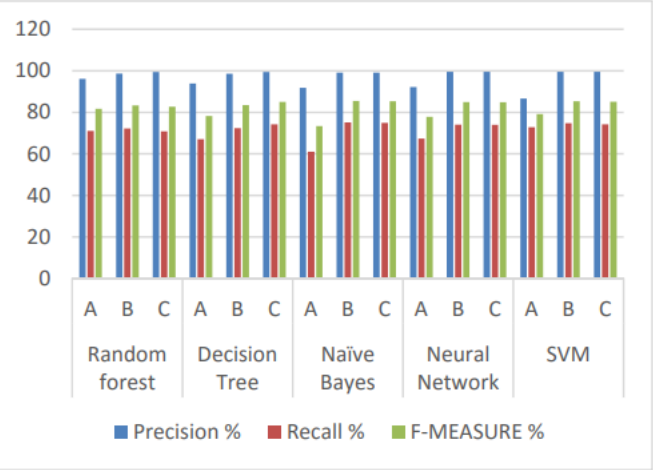


Fig. 1 Comparison of all the experiments results

CHAPTER-5

CONCLUSION:

In this research, we proposed an approach for detecting fake accounts on Twitter social network, the proposed approach was based on determining the effective features for the detection process. The attributes have been collected from different research, they have been filtered by extensive analysis as a first stage, and then the features have been weighted. Different experiments have been conducted to reach the minimum set of attributes with perceiving the best accuracy results. From more than 22 attributes, the proposed approach has reached only seven effective attributes for fake accounts detection. Although we claim that these attributes can succeed in discovering the fake accounts in other social networks such as Facebook with minor changes according to the unique nature of each social network, however, we need to prepare a dataset to prove our claim. Moreover, providing an analysis to the tweets content of the user can provide more accurate results in the detection process. we have maintained the highest accuracy in detecting fake accounts by different classifying algorithms. The results shows the increase of the accuracy results of two of the classification algorithms after using the suggested attributes with their corresponding heaviness. The classification algorithms are proposed to improve detecting fake accounts on social networks, where the SVM trained model decision values were used to train a NN model, and SVM testing decision values were used to test the NN model.

SCOPE FOR FUTURE WORK:

Fake accounts are being continuously evolving in online social media. Therefore, it is very essential to invent new methods to detect Fake profiles in online social media. So the real time Twitter dataset were required to detect the fake accounts and vulgar images in Twitter. For the detection of Fake accounts, the user timeline information namely post-count, comment-count, etc. were used and for the vulgar image detection the images from the user time line and the display picture of the users were taken out. The performance was evaluated using the supervised machine learning algorithms and the highest 80%accuracy were obtained and the maximum percentage of skin exposed were calculated from the images collected from the fake accounts. For the future scope, a more complex algorithm for the skin detection can be implemented. The natural language processing techniques can be implemented to detect fake accounts more accurately. The new features will be certainly introduced by the Twitter, and these features can also be included while analysing the fake accounts.

REFERENCES:

[1] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, 2012.

[2] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete, "Information credibility on twitter," in Proceedings of the 20th international conference on Worldwide web, 2011.

[3] Manish Gupta, Peixiang Zhao, and Jiawei Han, "Evaluating Event Credibility on Twitter," Siam, 2012.

[4] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting spam on social web sites: A survey of approaches and future challenges," IEEE Internet Computing, 11, 2007.

[5] Aditi Gupta, Hemank Lamba, and Ponnurangam Kumaraguru, "$1.00 per RT #BostonMarathon #PrayForBoston: Analyzing Fake Content on Twitter," Eigth IEEE APWG eCrime Research Summit (eCRS), 12, 2013.

[6] Yazan Boshmaf et al., "Íntegro: Leveraging Victim Prediction for Robust Fake Account Detection in OSNs," in NDSS ’15, 8-11 , San Diego, CA, USA, February 2015.

[7] Vladislav Kontsevoi, Naim Lujan, and Adrian Orozco, "Detecting Subversion of Twitter," May 14, 2014.

[8] Fabr´ıcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virg´ılio Almeida, "Detecting spammers on twitter," Collaboration, electronic messaging, anti-abuse and spam conference (CEAS). Vol. 6, 2010.

[9] Supraja Gurajala, Joshua S. White, Brian Hudson, and Jeanna N. Matthews, "Fake Twitter accounts: Profile characteristics obtained using an activity-based pattern detection approach," in SMSociety '15, July 27 - 29, Toronto, ON, Canada, 2015

[10] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in Proceedings of the 26th Annual Computer Security Applications Conference, 2010, pp. 1–9.

[11] L. Breiman, "Random forests," Machine Learning, 2001.

[12] Zhi Yang et al., "Uncovering Social Network Sybils in the Wild," in Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, November 02-04, 2011, Berlin, Germany, 2011.

[13] T. Joachims, Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms. Boston: Kluwer Academic Publishers, 2002.

[14] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, Introduction to Information Retrieval. New York: Cambridge University, 2008.

[15] SocialBakers. (Online) http://www.socialbakers.com/products/ analytics?ref=fakefollowers-top-bar, last retrieved on 30-10-2015

[16] M. Camisani-Calzolari. (2012, August) Analysis of Twitter followers of the US Presidential Election candidates: Barack Obama and Mitt Romney. (Online). http://digitalevaluations.com/

[17] The Fake project. (Online). http://wafi.iit.cnr.it/theFakeProject/ (last retrieved on 30-10-2015).

[18] Asha Gowda Karegowda, A. S. Manjunath, and M.A. Jayaram, "Comparative Study of Attribute Selection Using Gain Ratio," International Journal of Information Technology and Knowledge Management, vol. 2, no. 2, pp. 271-277, July-December 2010.

[19] Tatsunori Mori, Miwa Kikuchi, and Kazufumi Yoshida, "ermWeighting Method based on Information Gain Ratio for Summarizing Documents retrieved by IR systems," Journal of Natural Language Processing, vol. 9, no. 4, pp. 3--32, 2002.

[20] S. Cresci, M. Petrocchi, and R. Di Pietro, "A criticism to Society (As seen by Twitter analystics)," in IEEE 34th international conference on distributes computing systems workshops, 2014.