# CSE3021- Social and Information Networks

## J Component –Project Report

# *FAKE PROFILE DETECTION SYSTEM*

*By*


Reg. No: *20BCE1560* Name: *Kunal Gupta*
Reg. No: *20BCE1650* Name: *Amritansh Anand*

B.Tech CSE-CORE


*Submitted to*


**Dr.A.Bhuvaneswari,**
Assistant Professor Senior,
SCOPE, VIT, Chennai

**School of Computer Science and Engineering**



VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

*August 2022*

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

*BONAFIDE CERTIFICATE*

Certified that this project report entitled “Fake Profile Detection System” is a bonafide work of Kunal Gupta-20BCE1560 and Amritansh Anand-20BCE1650, who carried out the J-component under my supervision and guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

**Dr.A.Bhuvaneswari,**

Assistant Professor Senior,

SCOPE, VIT, Chennai

# ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. A. Bhuvaneswari Assistant** Professor, School of Computer Science Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We express our thanks to our HOD **Dr. P. Nithyanandam** for her support throughout the course of this project.

We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Kunal Gupta 20BCE1560
Amritansh Anand 20BCE1650

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

# School of Computing Science and Engineering

VIT Chennai

Vandalur - Kelambakkam Road, Chennai - 600 127

FALL SEM 22-23

# Worklet details

| | | |
|---|---|---|
| *Programme* | B.Tech with CSE-CORE | |
| *Course Name / Code* | Social and Information Networks – CSE3201 | |
| *Slot* | B1+TB1 | |
| *Faculty Name* | Dr.A.Bhuvaneswari | |
| *Component* | J – Component | |
| *J Component Title* | Fake Profile Detection System | |
| *Team Members Name \| Reg. No* | Kunal Gupta | 20BCE1560 |
| | Amritansh Anand | 20BCE1650 |

**Team Members(s) Contributions:**

| ***Worklet Tasks*** | ***Contributor's Names*** |
|---|---|
| Dataset Implementation | Amritansh Anand & Kunal Gupta |
| Preprocessing | Amritansh Anand & Kunal Gupta |
| Model building | Amritansh Anand & Kunal Gupta |
| Visualization | Amritansh Anand & Kunal Gupta |
| Technical Report writing | Amritansh Anand & Kunal Gupta |
| Presentation preparation | Amritansh Anand & Kunal Gupta |

# ABSTRACT

In the present generation, the social life of everyone has become associated with 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 solutions exist to control these problems. In this report, we came up with a framework with which the automatic identification of fake profiles is possible and is efficient. This framework uses classification techniques like Random Forest Classifier 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 that have millions of profiles whose profiles cannot be examined manually. Also, we will be comparing the result with another method of implementation that is already existing, i.e., SVM (Support Vector Machine).

# 1. Introduction

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) use web2.0 technology, which allows users to interact with each other. Social networking sites are growing rapidly and changing the way people keep in contact with each other. The online communities bring people with the same interests together which makes users easier to make new friends.

In today's online social networks there have been a lot of problems like fake profiles, online impersonation, etc. To date, no one has come up with a feasible solution to these problems. In this project, I 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 can't be done manually.

# 2. Literature Survey

| Sl. no | Title | Author / Journal name / Year | Technique | Result (Accuracy) |
|---|---|---|---|---|
| 1 | Identifying Fake Profiles in LinkedIn | Shalinda Adikari & Kaushik Dutta<br><br>arXiv preprint arXiv: 2006.01381<br><br>(2020) | Number of languages spoken, education, skills, recommendations, interests, awards, etc. are used as features to train neural networks, SVMs, and principal component analysis. | 84% TP, 2.44% FN |

| | | | | |
|---|---|---|---|---|
| 2 | Source Based Fake News Classification using Machine Learning | Avinash Bharadwaj, Brinda Ashar<br><br>IJIRSET, 2020 | I. Text classification via Bayesian classifier (Orthogonal Sparse Bigram);<br>2. Regularity of tweets;<br>3. Frequency and types of URLs; the use of APIs. | 100% |
| 3 | Identifying Fake Profile in Online Social Network: An Overview and Survey. | Shruti Joshi<br><br>International Conference on Machine Learning, Image Processing, Network Security and Data Sciences. Springer, Singapore, 2020. | In a general approach for the identification of fake accounts in large scale online social networks following steps are used:<br>• Data collection.<br>• Feature selection.<br>• Feature extraction.<br>• Data classification/techniques used. | 79% |
| 4. | The social honeypot project: protecting online communities from spammers | Kyumin Lee & Steve Webb<br><br>Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, 2010 | Over 60 classifiers available in Weka are tried.<br>Features include:<br>i) demographics,<br>ii) content and<br>iii) frequency of content generation,<br>iv) number and type of connections. The Decorate meta-classifier provided the best results. | 99,21% (MySpace). 88.98% (Twitter) |
| 5. | Detecting spammers on social networks | Gianluca Stringhini<br><br>Twenty-Sixth Annual | Random forest was constructed based on the following features: ratio of accepted friend requests, URL. ratio, message similarity, regularity in | 2% FP. 1% FN (Facebook): 2.5% FP. 3.0% |

| | | | | |
|---|---|---|---|---|
| | | Computer Security Applications Conference, ACSAC 2010, Austin, Texas, USA, 2010 | the choice of friends, messages sent, and number of friends. | EN (Twitter) |
| 6. | Twitter spam detection: Survey of new approaches and comparative study | Yang Xiang & Tingmin Wu November 2017 | Graph based features (local clustering coefficient, betweenness centrality, and bi-directional links ratio), neighbor-based features (e.g. average neighbors' followers), automation-based features (API ratio, API URL ratio and API Tweet similarity), and timing-based features were used to construct different classifiers. | 86% TP. 0,5% EP |
| 7. | Aiding the Detection of Fake Accounts in Large Scale Social Online Services | Xiaowei Yang<br>USENIX conference, 2012 | Invitation frequency, rate of accepted outgoing and incoming requests, and clustering coefficient were used as features for an SVM classifier. | 99% |
| 8. | Towards Detecting Fake User Accounts in Facebook | Aditi Gupta & Rishabh Kaushal 2017 ISEA Asia Security and Privacy (ISEASP) January 2017 | Machine learning classification techniques on a basic dataset comprising of our own node and our friends in our social neighborhood and also a set of manually identified spam accounts. | accuracy of 79%. ( 79% |
| 9. | Facebook immune system | Stein, Tao, Erdong Chen, and Karan | To protect the graph the Immune System runs classifiers to block and respond and anomaly detection | 82% |

| | | | | |
|---|---|---|---|---|
| | | Mangla<br><br>Proceedings of the 4th workshop on social network systems, 2011 | to detect new and mutated at- tacks. Developing, deploying, and operating these classifiers has a number of challenges. Attacks mutate across different channels within a large user-interface surface area. The system must defend against these attacks while meeting severe scalability and latency requirements. This section discusses several of the important sys- tem requirements. | |
| 10. | Detecting fake accounts on social media. | 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018 | In this paper, a new algorithm, SVM-NN, is proposed to provide efficient detection for fake Twitter accounts and bots, feature selection and dimension reduction techniques were applied. Machine learning classification algorithms were used to decide the target accounts identity real or fake, those algorithms were support vector machine (SVM), neural Network (NN), and our newly developed algorithm, SVM-NN. | 98% |

# 3. Dataset and Tool to be used

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

used to prepare a training dataset and 20% of both profiles are used to prepare a testing dataset.

LINK TO DATASET:
https://www.kaggle.com/datasets/joewilson02/social-media-fake-account

# 4. Algorithms / Techniques description

The classifier that we have implemented for classifying the profiles is Random Forest.

Random forest is a supervised learning algorithm that is used for both classifications as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees mean more robust forests. Similarly, the random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method that is better than a single decision tree because it reduces the over-fitting by averaging the result.

We can understand the working of the Random Forest algorithm with the help of following steps:
Step 1 − First, start with the selection of random samples from a given dataset.
Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
Step 3 − In this step, voting will be performed for every predicted result.
Step 4 − At last, select the most voted prediction result as the final prediction result.

# 5. Implementation Details

## 5.1 Proposed Framework

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

1. The detection process starts with the selection of the profile that needs to be tested.
2. After the 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 whether the profile is fake or genuine.
5. The classifier may not be 100% accurate in classifying the profile so; the feedback of the result is given back to the classifier.
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.

## 5.2 Attributes Considered

Table below shows the Attributes considered for fake profile identification and the description for each of the attributes is provided.

| S. No | Attribute | Description |
|---|---|---|
| 1 | Profile ID | The Profile ID of account holder |
| 2 | Profile Name | The name of the account bolder |
| 3 | Status Count | The number of tweets made by the account |
| 4 | Followers Count | The number of followers for the account |
| 5 | Friends Count | The number of friends for the account |
| 6 | Location | The location of the account holder |
| 7 | Created Date | The date the account was created |
| 8 | Share count | The number of shares done by account holder |
| 9 | Gender | The Gender of the account holder |
| 10 | Language Code | The language of account bolder |

## 5.3 Evaluation Parameters

Efficiency/Accuracy = Number of predictions/Total

Number of Predictions Percent Error = (1-Accuracy)*100

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

TPR- True Positive Rate TPR=TP/(TP+FN)

FPR- False Positive Rate FPR=FP/(FP+TN)

TNR- True Negative Rate TNR=TN/(FP+TN)

FNR- False Negative Rate FNR=1-TPR

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

Recall = TP / (TP+FN)

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

Precision = TP / (TP + FP)

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

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

# 6. Results And Discussion

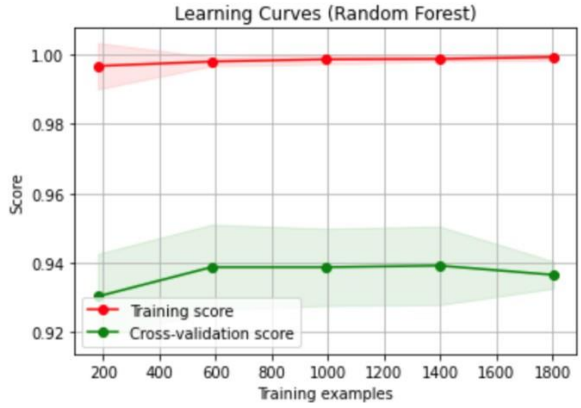
The efficiency of the Random Forest Classifier in classifying data is 95% and SVM is 88%. We have taken 80% of the data for the training dataset and 20% for the testing dataset.

**Random Forest-**

```
print("training datasets.......\n")
y_test,y_pred = train(X_train,y_train,X_test)
```

```
training datasets.......

The best classifier is:  RandomForestClassifier(n_estimators=40, oob_score=True)
[0.92239468 0.94013304 0.94456763 0.94900222 0.93555556]
Estimated score: 0.93833 (+/- 0.00457)
```

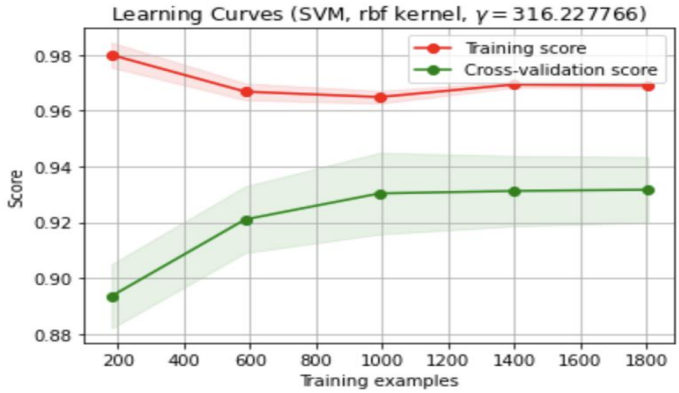



**SVM-**

```
[ ]  print("training datasets.......\n")
     y_test,y_pred = train(X_train,y_train,X_test)
```

```
training datasets.......

The best classifier is:  SVC(gamma=316.22776601683796)
[0.92017738 0.91796009 0.93126386 0.94900222 0.94       ]
Estimated score: 0.93168 (+/- 0.00587)
```



## Random Forest-

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

```
Classification Accuracy on Test dataset:  0.9485815602836879
```

## SVM-

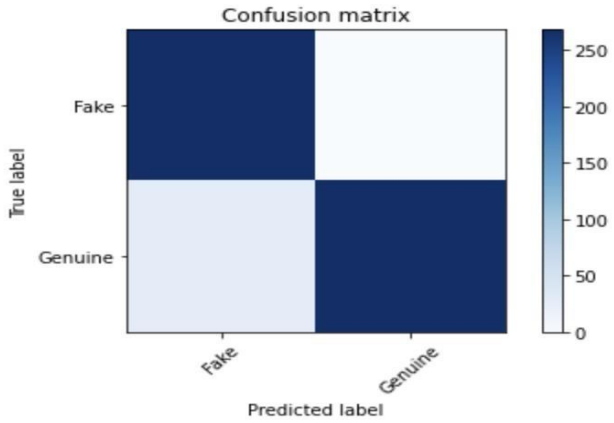
```
print('Classification Accuracy on Test dataset: ' ,accuracy_score(y_test, y_pred))
```

```
Classification Accuracy on Test dataset:  0.875886524822695
```

## Random Forest-

```
cm=confusion_matrix(y_test, y_pred)
print('Confusion matrix, without normalization')
print(cm)
plot_confusion_matrix(cm)
```

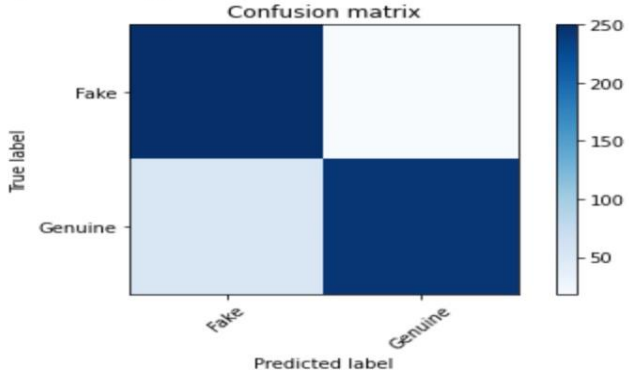
```
Confusion matrix, without normalization
[[268   0]
 [ 29 267]]
```




## SVM-

```
cm=confusion_matrix(y_test, y_pred)
print('Confusion matrix, without normalization')
print(cm)
plot_confusion_matrix(cm)
```

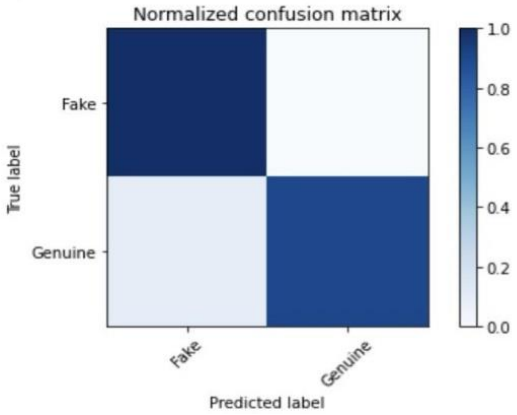
```
Confusion matrix, without normalization
[[250  18]
 [ 52 244]]
```



## Random Forest-

```
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')
```

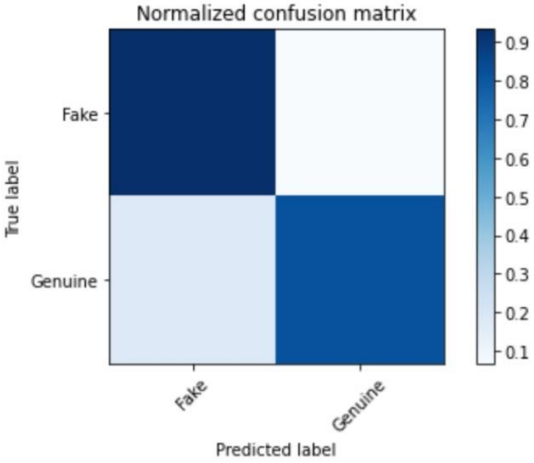
```
Normalized confusion matrix
[[1.         0.        ]
 [0.09797297 0.90202703]]
```




## SVM-

```
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')
```

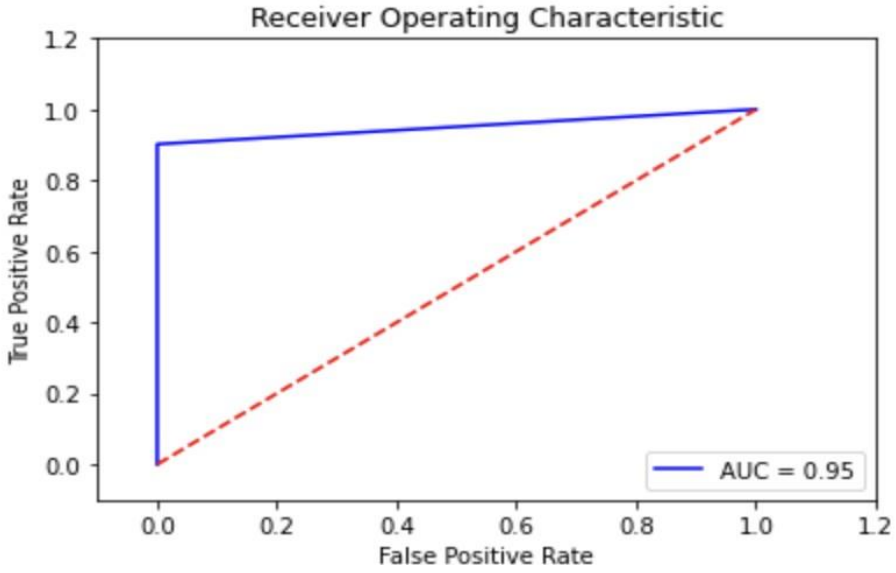
```
Normalized confusion matrix
[[0.93283582 0.06716418]
 [0.17567568 0.82432432]]
```



## Random Forest-

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

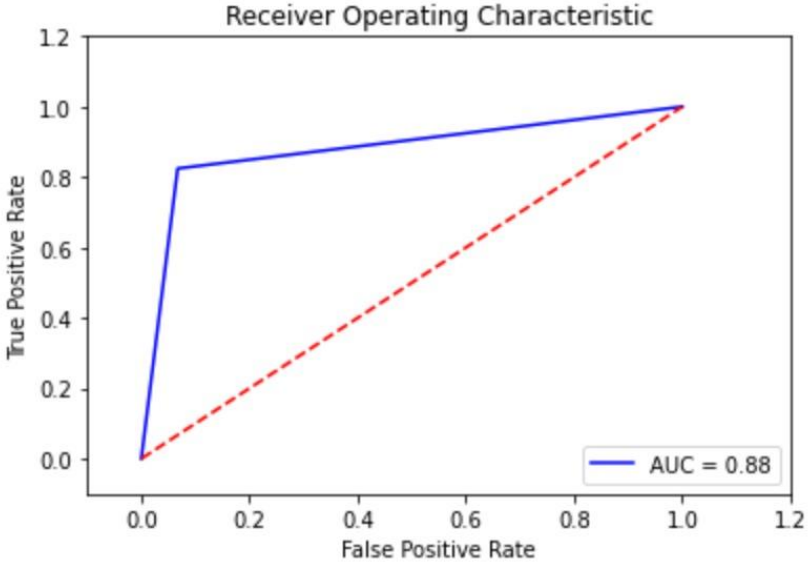
```
False Positive rate:  [0. 0. 1.]
True Positive rate:  [0.         0.90202703 1.        ]
```




## SVM

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

```
False Positive rate:  [0.         0.06716418 1.        ]
True Positive rate:  [0.         0.82432432 1.        ]
```



# 7. Model Evaluation-

**Random Forest-**

```
print(classification_report(y_test, y_pred, target_names=['Fake','Genuine']))
```

```
              precision    recall  f1-score   support

        Fake       0.90      1.00      0.95       268
     Genuine       1.00      0.90      0.95       296

    accuracy                           0.95       564
   macro avg       0.95      0.95      0.95       564
weighted avg       0.95      0.95      0.95       564
```

**SVM-**

```
print(classification_report(y_test, y_pred, target_names=['Fake','Genuine']))
```

```
              precision    recall  f1-score   support

        Fake       0.83      0.93      0.88       268
     Genuine       0.93      0.82      0.87       296

    accuracy                           0.88       564
   macro avg       0.88      0.88      0.88       564
weighted avg       0.88      0.88      0.88       564
```

# 8. GitHub Repository Link

https://github.com/amritansha28/amritanshkunal_fakeprofiledetection

# 9. Conclusion

We have given a framework using which we can identify fake profiles in any online social network by using Random Forest Classifier with a very high efficiency as high as around 95% and with a comparative analysis through SVM with a efficiency of around 88%. Fake profile Identification can be improved by applying NLP techniques and Neural Networks to process the posts and the profiles. In the future, we wish to classify profiles by taking profile pictures as one of the features.

# REFERENCES


[1] Mohanty, Sachi, et al. Recommender System with Machine Learning and Artificial Intelligence. Wiley-Scrivener, 2020.

[2] Balaanand, Muthu, et al. "An enhanced graph-based semi-supervised learning algorithm to detect fake users on Twitter." The Journal of Supercomputing 75.9 (2019): 6085-6105.

[3] Boshmaf, Yazan, et al. "Integro: Leveraging Victim Prediction for Robust Fake Account Detection in OSNs." NDSS. Vol. 15. 2015.

[4] Erşahin, Buket, et al. "Twitter fake account detection." 2017 International Conference on Computer Science and Engineering (UBMK). IEEE, 2017.

[5] Mateen, Malik, et al. "A hybrid approach for spam detection for Twitter." 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST). IEEE, 2017.

[6] Gupta, Arushi, and Rishabh Kaushal. "Improving spam detection in online social networks." 2015 International conference on cognitive computing and information processing (CCIP). IEEE, 2015.

[7] Nazir, Atif, Saqib Raza, Chen-Nee Chuah, Burkhard Schipper, and C. A. Davis. “Ghostbusting Facebook: Detecting and Characterizing Phantom Profiles in Online Social Gaming Applications.” In WOSN. 2010.

[9] Chu, Zi, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. “Who is tweeting on Twitter: human, bot, or cyborg?.” In Proceedings of the 26th annual computer security applications conference, pp. 21- 30. ACM, 2010.

[10] Stringhini, Gianluca, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y. Zhao. “Follow the green: growth and dynamics in twitter follower markets.” In Proceedings of the 2013 conference on Internet measurement conference, pp. 163-176. ACM, 2013.