

FAKE SOCIAL PROFILE DETECTION USING MACHINE LEARNING

TALARI SIVALAKSHMI, K BALAJI SUNIL CHANDRA, KUMMARA RANGA SWAMY

Assistant Professor^{1,2,3},

shivalakshmidinesh@gmail.com, rangaswamy.kumara@gmail.com, hod.cse@svitatp.ac.in

department of CSE, Sri Venkateswara Institute of Technology,
N.H 44, Hampapuram, Raphadu, Anantapuramu, Andhra Pradesh 515722

Keywords:

Machine Learning,
Support Vector
Machines, Neural
Networks, Random
Forest, and Fake Social
Profile Detection

ABSTRACT

One of the most popular and widely utilised platforms for digital marketing, social media allows firms to keep tabs on public trends and preferences, and it also provides valuable insights into consumer behaviour. The number of false social media accounts that disseminate misinformation is on the rise. In order to address the issues surrounding the identification of false social media profiles, this study examines several machine learning techniques. Jupyter Notebook makes use of Python and a number of machine learning and data analytics libraries, including Numpy, Pandas, Sklearn, and others. Using AUC Score, Confusion Matrix, and total number of Fake and Genuine Users discovered, this article compares three machine learning algorithms: Support Vector Machines (SVM), Random Forest, and Neural Networks. For easier examination and comparison across all methods, results are shown as graphs. The dataset that was used for this project can be found in the following link: At this URL: <https://github.com/1205T1/SOCIALPROFILE.git>



This work is licensed under a Creative Commons Attribution Non-Commercial 4.0 International License.

Introduction

The term "social network" describes the web of interconnected online communities that many of us have been used to seeing in our daily lives. Facebook, Twitter, Instagram, Whatsapp, and many more are just a few of the thousands of social media platforms accessible today. In fact, many people now conduct their whole jobs on social media. There are a lot of individuals on the social network that we may accidentally or purposefully connect to. Like any other tool, social networks have their benefits and drawbacks. Every one of us relies on social media sites on a regular basis, whether for business or play. Without a doubt, Facebook is among the most widely utilised and well-known social networking sites. It may have been the first of its kind to cross the 1 billion user milestone. Facebook is great for a lot of reasons, like connecting with friends and family, making new acquaintances, and advertising your company or items via various apps. However, a big problem has emerged recently: the question of how trustworthy the millions of Facebook members really are. The topic of trust is addressed in this work. The goal here is to make an educated guess as to whether or not the account creator is trustworthy; this is an area that has been the subject of several surveys, and recent studies have shown that trust problems persist. The reason being that Facebook has access to an enormous quantity of data. Finding out whether the Facebook account is real or not is therefore not a simple task. We used machine learning to try to forecast which Facebook profiles might be fraudulent.

Per Facebook's security and privacy rules. Users may be certain that their Facebook profiles are authentic when they use this feature. Users should be forthright and honest with Facebook about their rights and duties. When they finished making their profiles. Many phoney accounts are also formed as a result of these rules. Fake Facebook accounts are a quick subject that we touch on. This article presents a smart forecasting system that uses data mining's prediction and classification techniques to solve the challenge of anticipating bogus accounts. The emphasis is on this particular area of research. The works that are exhibited inspire us to work. We want to analyse data better, improve our predictions on fraudulent profiles, and build better techniques to accurately anticipate Facebook accounts that are not real. To that end, we have developed a smart detection system called FB Checker.

Secondly, machine learning Nowadays, machine learning is a popular tool for a variety of tasks and methods. Supervised, unsupervised, and reinforcement learning are three of the many approaches to machine learning. employing these many methods from machine learning, you can zero in on a plethora of issues plaguing Facebook and find solutions. Machine learning is a popular and useful tool for a wide range of activities across many industries. Machine learning encompasses a wide range of methods that researchers use to accomplish certain tasks.

SVM – SUPPORT VECTORMACHINES

A Support-Vector Machine is a machine learning algorithm comes under supervised category. After providing the support vector machine model set of labeled trained data for every category, new text can be categorized. SVM is a fast and reliable algorithm which works perfectly for small amount of data to analyze. The main idea with SVM is pretty simple and it applies to natural language classification that need

<https://doi.org/10.5281/zenodo.12707506>

not to have the complex stuff. A simple example depicting Support Vector Machines is provided where we have two color tags i.e. red and blue, with our data is having two features i.e. x and y. A classifier is needed with a pair of coordinates that displays if it is red or blue. Data which is already labeled is plotted as below:

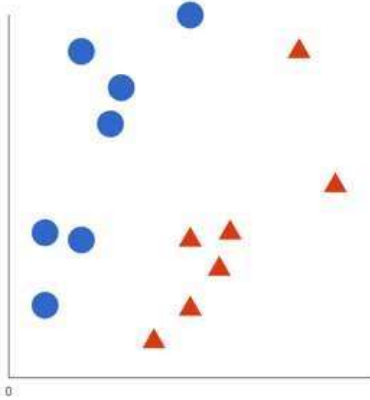


Figure 1.1 – Labeled Data with SVM

Support Vector Machine get the data points and displays the hyperplane that in two-dimensions is in actual a simple line separating the red and blue tags. The line here acts as the decision boundary which means that anything that is on one side is blue and anything that falls on the other side is red. A figure below depicting the classification:

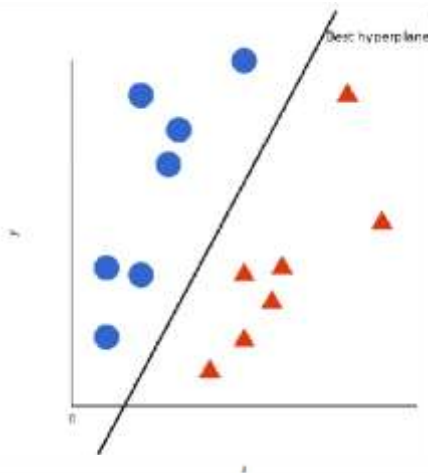


Figure 1.2 – Hyperplane is line in simple 2D With Hyperplane in SVM, nearest the distance between elements larger the tag is as shown below:

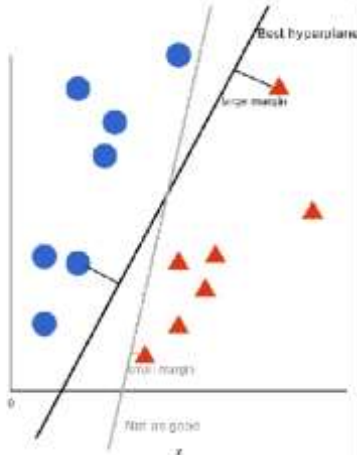


Figure 1.3 – Not all hyperplanes are equal

1.1. Neural Networks

Neural Networks are subset of deep learning which is an subset of machine learning where a structure like human brain stimulate algorithms. NN get the input data, train themselves to recognize patterns found in data, and then prediction is made for the output for similar set of data. It's the functional part of deep learning, which imitates the way humans solve the problems using their brains.

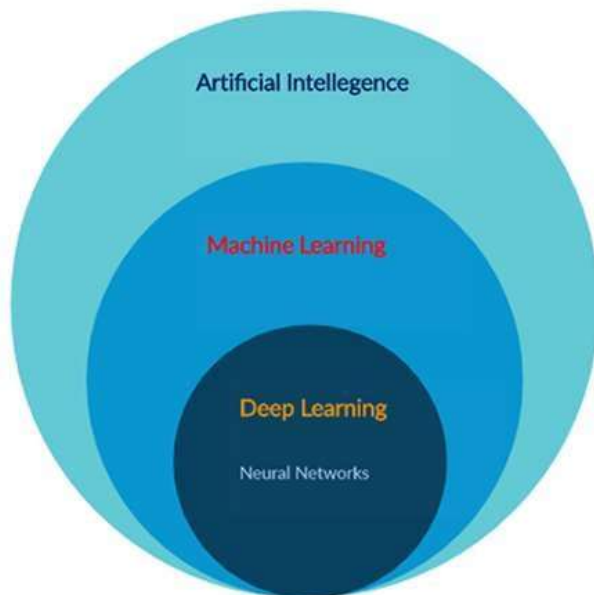


Figure 1.4 – Neural Networks Structure

1.2. Random Forest

Random Forest as the name depicts, consists of huge number of individual decision trees that act as an ensemble. Every individual tree in the random forest brings a class prediction and any class that

<https://doi.org/10.5281/zenodo.12707506>

has most number of votes in its favor becomes the prediction of the model. A figure depicting random forest behavior is shown below:

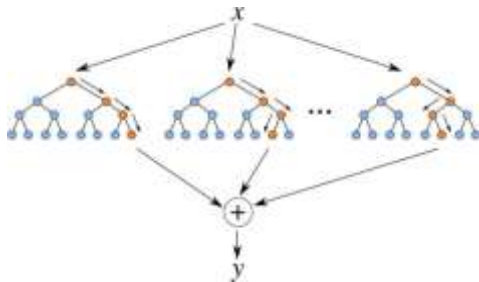


Figure 1.5 – Random Forest

The working of random forest algorithm is divided into two phases, where in first stage, „n” random trees are built and they further create a random forest. In the second stage, the outcome of all the decision trees is integrated. Final prediction is fetched by computing the results of every decision tree.

2. RESULTS AND DISCUSSIONS

Detection of fake profiles in online social networks using Random Forest algorithm

Step 1: The detection of fake profiles on online social network sites begins with importing libraries including sys, csv, datetime, numpy, pandas, matplotlib, and sklearn.

Step 2: The next step is to read the datasets from the csv file while describing the fake and genuine users.

Step 3: Defining a function that basically finds or predicts the gender by providing the name of the user. The first name gets segregated from the string for computation. **Step 4:** After this extract_feature function needs to be defined that describes various features as well.

Step 5: For plotting the curve, function is defined. It will describe the cross-validation score of the dataset.

Step 6: Furthermore, for plotting the confusion matrix, function has been defined that will work on the basis of fake and genuine users.

Step 7: Function for describing the ROC plot needs to be implemented, which is meant for defining the characteristics of False Positive and True Positive values.

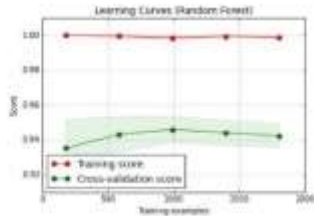
Step 8: To train the model using random forest algorithm is being implemented. This will provide the estimated score of the classifier while discussing the curve.

Step 9: The output of the trained data is shown ahead. It describes the status count, follower count, and favorite color and friends count of each user.

Step 10: Next step is to split the values of datasets according to train and test the dataset. The learning curve has been published after processing the datasets.

```
In [67]: print 'training datasets.....\n'
y_test, y_pred = train(X_train, y_train, X_test)
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)
)
[ 0.93781574 0.93781574 0.94678492 0.9578714 0.95777778]
Estimated score: 0.94365 (+/- 0.00385)
```

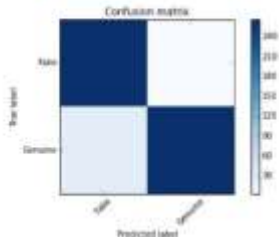


Step 11: The classification accuracy of the datasets of test model has been displaying the outcome value which is promising after computation.

```
In [68]: print 'Classification Accuracy on Test dataset: ', accuracy_score(y_test,
y_pred)
Classification Accuracy on Test dataset: 0.941489361782
```

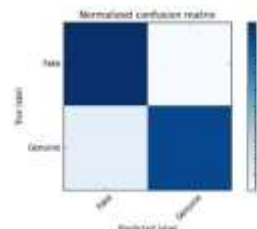
Step 12: The confusion matrix before normalization is as follows:

```
In [76]: cm=confusion_matrix(y_test, y_pred)
print('Confusion matrix, without normalization')
print(cm)
plot_confusion_matrix(cm)
Confusion matrix, without normalization
[[268  3]
 [ 30 266]]
```



Step 13: After normalizing the dataset, the confusion matrix has been presented.

```
In [71]: 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
[[ 0.98885597  0.01114403]
 [ 0.0233125  0.9766875]]
```



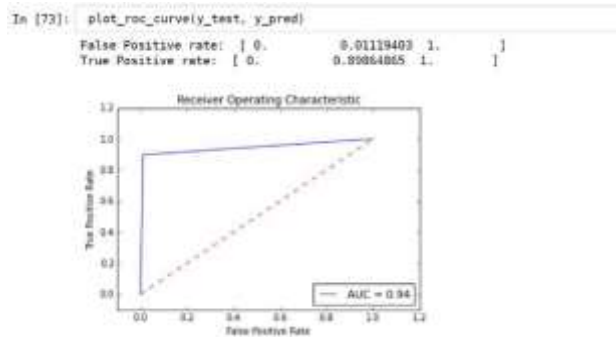
Step 14: The report of classification as per the matrix of Fake and Genuine accounts have to be noted.

```
In [72]: print(classification_report(y_test, y_pred, target_names=['Fake', 'Genuine']))
```

	precision	recall	f1-score	support
Fake	0.98	0.98	0.94	268
Genuine	0.99	0.90	0.94	296
avg / total	0.95	0.94	0.94	564

<https://doi.org/10.5281/zenodo.12707506>

Step 15: Finally, ROC curve has to be plotted along with True Positive and False Negative features.



3.2. Fake Profile detection with Neural Networks For the detection of fake profiles in online social networking site using Neural network methodology, following steps have been taken into consideration: Step 1: Firstly, we have imported libraries of sys, csv, os, datetime, math, numpy, panda, and matplotlib.

Step 2: After this, gender detection libraries are being loaded to compute the information about the gender. For validating the data and preprocessing, sklearn libraries have been integrated to plot the matrix. The evaluation metric provides the information about different variables of confusion matrix. For the evaluation of classifier, area under cover and accuracy have been used.

Step 3: Then, import the Pybrain library for training the datasets. It is freely available open-sourced library for machine learning algorithms. Different utility tools get implemented along with this library.

Step 4: Next is to read the dataset by defining a function name read_datasets(). The CSV or comma separated value files get used for this. We need to set the default for datasets to read. After combining the fake and authentic user, length of users needs to be found.

Step 5: Afterward, another function is defined to predict the gender of the person through the given name. First name of the person gets declassified into parts for computing the model. Further, different features associated with that will be integrated along with status counts, and followers counts and so on.

Step 6: Next, plotting of confusion matrix begins which integrate the plot as per the fake and genuine profile accounts.

Step 7: Function for defining the ROC curve has been implementing for further computation.

Step 8: Using the neural network, function is declared to train the dataset. For this, read_datasets() has been used.

Step 9: After reading the data, the output will look like this.

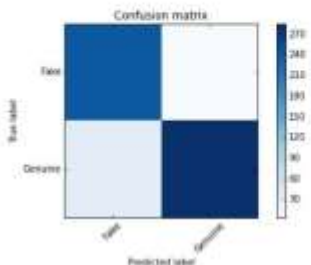
Step 10: Extracting the features of the training datasets.

Step 11: The graph of confusion matrix without normalization has been shown.

<https://doi.org/10.5281/zenodo.12707506>

```
In [13]: cm=confusion_matrix(y_test, y_pred)
print('Confusion matrix, without normalization')
print(cm)
plot_confusion_matrix(cm)

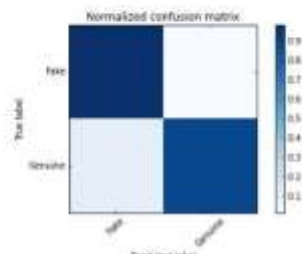
Confusion matrix, without normalization
[[241  4]
 [33 265]]
```



Step 12: After doing the normalization of the confusion matrix.

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

Normalized confusion matrix
[[ 0.9867347  0.0132653]
 [ 0.10377358  0.89622642]]
```



Step 13: The classification of reports has been presented to define the fake and genuine profile precision index, recall, f1-score and support vector.

```
In [15]: print(classification_report(y_test, y_pred, target_names=['Fake', 'Genuine']))
```

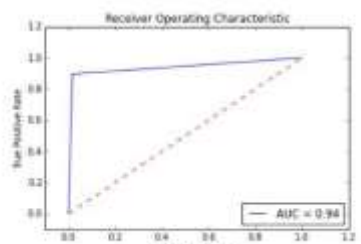
	precision	recall	f1-score	support
Fake	0.88	0.98	0.93	245
Genuine	0.99	0.90	0.94	318
avg / total	0.94	0.93	0.93	563

```
In [16]: v=roc_auc_score(y_test, y_pred)
print('roc_auc_score : ',v)
roc_auc_score : 0.939949942241
```

Step 14: Final outcome of the experiment is as following that describe the false positive and true positive values.

```
In [17]: plot_roc_curve(y_test, y_pred)
```

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



3.3. Detect fake profiles in online social networks using Support Vector Machine

<https://doi.org/10.5281/zenodo.12707506>

Step 1: For the detection of fake profiles onlinesocial networks using SVM, several libraries needto be integrated including sys, csv, datetime,matplotlib and so on. These libraries are essential for reading the csv datasets and plotting the matrix.

Step 2: The second step contains, reading of datasets. In our case the name of the file is fusers.csv and users.csv to train and test the model. The genuine users are stored in user.csv file while fake or bogus users are in fusers.

Step 3: Define a function to get the informationabout the gender through name given in the dataset. Step

4: For feature extraction, we have declared thefunction „extract_features”.

Step 5: Further to this, we will draw the plot oflearning curve which will provide informationabout certain features used to process the vectors. Step 6: Next, there is confusion plot matrixassociated with Fake & Genuine user profiles.Also, we will set the color value for plotting thesame.

Step 7: For plotting the ROC or receiver operatingcharacteristic, function has been defined.

Step 8: For training the dataset with supportvector machine, function with the name of „train”has been declared along with the SVM classifier. Step 9: Next step is for reading and extracting thedatasets features.

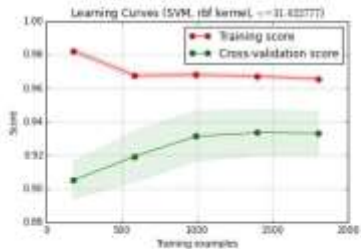
Step 10: Different values will be shown afterextracting the features, including count ofstatuses, follower counts, listed count, language code and so on for further processing.

Step 11: Splitting the datasets from the training and testing datasets.

Step 12: The training data of learning curve will be displayed in red color and the cross-validation in green color. The score data has been mentioned along with the training.

```
In [79]: print 'training datasets.....\n'
y_test,y_pred = train(X_train,y_train,X_test)
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))
[ 0.91796099  0.92904658  0.92463197  0.9556541  0.93777778]
Estimated score: 0.93301 (+/- 0.00651)
```

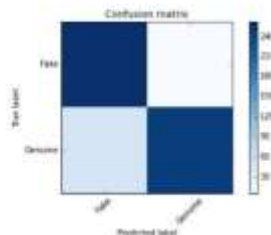


```
In [80]: print 'Classification Accuracy on Test dataset: ',accuracy_score(y_test,
y_pred)
Classification Accuracy on Test dataset: 0.904255319145
```

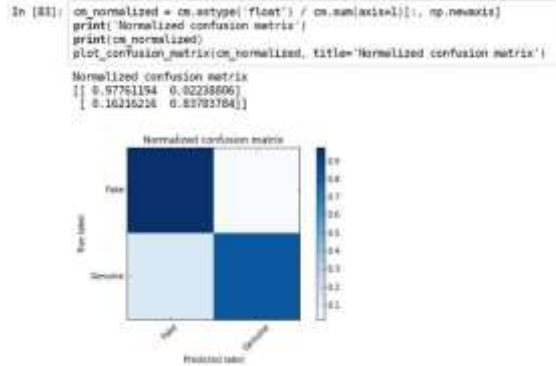
Step 13: Predictive labeling of confusion matrix is being performed before normalization.

```
In [82]: cm=confusion_matrix(y_test, y_pred)
print('Confusion matrix, without normalization')
print(cm)
plot_confusion_matrix(cm)

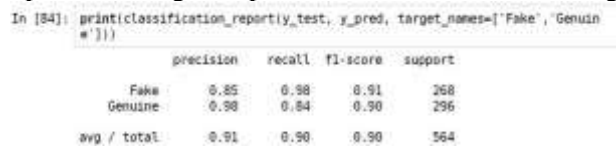
Confusion matrix, without normalization
[[252  6]
 [ 48 248]]
```



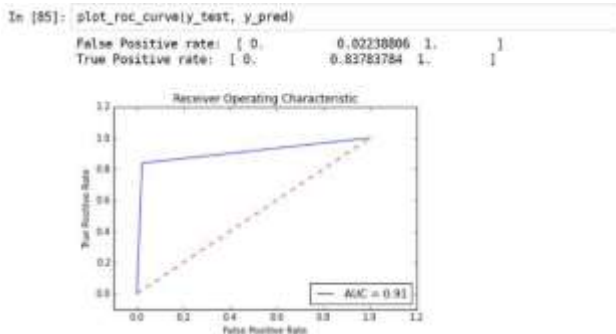
Step 14: Next Step is to normalize the given matrix.



Step 15: Printing of report of classification according to fake and genuine users.



Step 16: Lastly, plotting of ROC curve having True Positive and False Positive characteristics.



So, after using three different machine learning algorithms in our work, a comparison table is built on the basis of AUC score as given below

Table 3.1 – Comparison of RF, NN and SVM on the basis of AUC Score

Machine Learning Algorithm	Random Forest	Neural Networks	Support Vector Machine
AUC Score	0.94	0.94	0.91

CONCLUSION AND FUTURE SCOPE

Social Networks are booming in almost all the industries and they are becoming the main platform for companies to showcase their products or services to the end customers. Public Relation companies make millions with social networks by publicizing the content related with different entities which can be political parties, any celebrity, institutions etc. Fake News with the help of fake profiles is also increasing at rapid pace and people use fake identities on social networks to publicize fake news, they are also related with the fake reviews, comments etc. Social Network giants like Facebook, Twitter etc. continuously try to reduce the fake accounts by detecting them, but the problem is in actual keeps on rising with the rise of social network. We have used machine learning with Python in order to detect the fake social profiles. Three different algorithms i.e. Support Vector Machines (SVM), Neural Network (NN) and Random Forest (RF) are used and it is found that RF and NN bring higher AUC than SVM. Researchers are continuously working on reducing if not eliminating this big problem on social networks and with continuous

<https://doi.org/10.5281/zenodo.12707506>

improvements in Artificial Intelligence features, researchers in future expect to minimize this issue.

REFERENCES

- "Popularity-Based Detection of Malicious Content in Facebook Using Machine Learning Approach."
1. Sahoo, SomyaRanjan, and B. B. Gupta. The Very First Global Symposium on Eco-Friendly AI Computing Solutions. Press, Singapore: Springer, 2020.
 2. "An empirical study for detecting fake Facebook profiles using supervised mining techniques." Albayati, Mohammed Basil, and Ahmad Mousa Altamimi. The most recent version of Informatica is 43.1 from 2019.
 3. Ahmad Nazren Hakimi, et al.: "Identifying Fake Account in Facebook Using Machine Learning." A conference on visual informatics held annually. Cham, 2019 (Springer).
 4. "Towards detecting fake user accounts in Facebook." Gupta, Aditi, and RishabhKaushal, 2004.4. 2017 IEEE International Symposium on East Asian Security and Privacy (ISEASP).
 5. Durga Prasad, Samala Reddy. "Fake Profile Identification using Machine Learning." (2019).
 6. Yeshwant Singh and Subhasish Banerjee wrote the article. "Fake (Sybil) Account Detection Using Machine Learning." You may find it at SSRN 3462933 (2019).
 7. Michael Fire et al. "Ally or enemy?" Online social network fake profile detection. Mining and Analysis of Social Networks Article 4.1 (2014) page 194.
 8. "An automated framework for finding fake accounts on Facebook." Sami, Memoona, et al. Computer Science and Engineering: An International Journal of Advanced Studies 7.2 (2018): 8-16.
 10. "Identifying Fake Account in Facebook Using Machine Learning." Zainudin, NorulzahrahMohd, et al. The Proceedings of the 6th International Visual Informatics Conference, IVIC 2019, Bangi, Malaysia, November 19-21, 2019, Volume 6, Advances in Visual Informatics. In 2019, Springer Nature published the work. "Detection of fake profile in online social networks using machine learning."
 10. Singh, Naman, et al. The 18th Annual International Conference on Advancements in Computer and Communication Engineering (ICACCE). In 2018, the IEEE published. Zakaria Sahnoune, Esma Aïmeur, and the 1st place. "Privacy, Trust, and Manipulation in Online Relationships." Journal of Technology in Human Services (2019):1-35. "Survey on Automated System for Fake News Detection using NLP & Machine Learning Approach."
 12. Gurav, Subhadra, et al. 18. this year (2019).
 13. The authors are Sowmya and J. Shiva Shankar. "A Survey on Detection of Fake News in Social Media." International Journal of Research 9.4 (2019): 469–474. In the 2017 IEEE 15th Student Conference on Research and Development (SCOREd), ShlokaGilda presented her work on evaluating machine learning algorithms for fake news detection.
 15. "Fake News Detection Using Naive Bayes Classifier" at the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) by Mykhailo Granik and Volodymyr Mesyura.
 16. "Automatically Identifying Fake News in PopularTwitter Threads" presented at the 2017 IEEE International Conference on Smart Cloud by Cody Buntain and Jennifer Golbeck. "Automatic Online Fake News Detection Combining Content and Social Signals," published in 2017 by Marco L. Della Vedova, Eugenio Tacchini, Stefano Moret, Gabriele Ballarin, Massimo DiPierro, and Luca de Alfaro, with the ISSN number 2305-7254. This sentence is copyrighted by IEEE in 2015 and is titled "Improving Spam Detection in Online Social Networks" and was written by Arushi Gupta and Rishabh Kaushal. "Identifying Tweets witj Fake News," 2018 IEEE International Conference on Information Reuse and Integration for Data Science, SaranyaKrishanan and Min Chen, 19. 20. "Improving Spam Detection in Online Social Networks" (Arushi Gupta and Rishabh Kaushal, 2015), 978-1-4799-7171-8/15/\$31.00 copyright 2015 by the IEEE. The authors of the article are Conroy, Rubin, and Chen (2015). "Methods for finding fake news: Automatic deception detection." Proceedings of the Association for Information Science and Technology, 52(1), pp.1-4. 22. "How fake news goes viral: A case study" by S. Maheshwari,

<https://doi.org/10.5281/zenodo.12707506>

published in November 2016. [On the web]. This article may be found at: <https://www.nytimes.com/2016/11/20/business/media/how-false-news-spreads.html>. had a look at on 11/08/2017.

23. "Comparative Study of Text Summarization Methods" by Nikita Munot and Sharvari S. Govilkar, published in the September 2014 issue of the International Journal of Computer Applications (0975 - 8887), volume 102, issue 12. An automated methodology for discovering phoney accounts on Facebook was developed by Sami, Memon, Baloch, and Bhatti (24). Journal of Advanced Computer Science and Engineering, Volume 7, Issue 2, Pages 8-16 (2018)

25. Sohrabi, M.K., A feature selection method for spam detection in the Arab Journal of Science and Engineering, 43(2), 949-958 (2018). Facebook social network. 6. Campos, G.F., Tavares, G.M., Igawa, R.A., Guido, R.C.: Using wavelets to distinguish between humans, genuine bots, and malicious bots in online social networks. Applied Computing in Medicine and Healthcare (ACMM) 14(1), 26 (2018)

27. Gurumurthy et al.: (2019) Intelligent system design and implementation for harmful post detection on Facebook using support vector machine (SVM). Articles 17–24 from the book Soft Computing and Medical Bioinformatics. Singapore: Springer, 2019

Article 28: "Analysing and detecting money-laundering accounts in online social networks" (Zhou, Y., Wang, X., Zhang, J., Zhang, P., Liu, L., Jin, H., Jin, H.). (2018), 32(3), 115-121, IEEE Network. The paper "Abbasniff: automatic detection and defences against abusive Facebookfriends" was published in 2018 by Talukder and Carbunar.

30. Post-print at arXiv:1804.10159. Research by Wang, X., Lai, C.M., Hong, Y., Hsieh, C. J., and Wu, S.F. (2018) identifies several Facebook accounts by utilising semi-supervised learning on graphs. preprint on arXiv:1801.09838 page 31. "Facebook Inspector" (FbI) is an initiative by Dewan and Kumaraguru that aims to automatically identify harmful information on Facebook in real-time. The analysis of social networks. Chapter 7, Section 1, Page 15 (2017)