

## Fake Twitter(X) Profile Detection

Mr. Yogeshchandra L Puranik<sup>1</sup>, Dr. Ravikant S. Zirmite<sup>2</sup>, Mr. Manish Bhalerao<sup>3</sup>, Mrs. Manasi Shirurkar<sup>4</sup>

<sup>1</sup>Assistant Professor, MCA Dept., PES Modern College of Engg. Pune, S.P. Pune University, India.

<sup>2</sup>Professor, MCA Dept., MES IMCC Pune, S.P. Pune University, India.

<sup>3</sup>Research Scholar, MES IMCC Pune, S.P. Pune University, India.

<sup>4</sup>Assistant Professor, MCA Department, MES IMCC Pune, S.P. Pune University, India.

Email: 1 [yogeshchandra.puranik@gmail.com](mailto:yogeshchandra.puranik@gmail.com), 2 [rsz@mespune.in](mailto:rsz@mespune.in)

Cite this paper as: Mr. Yogeshchandra L Puranik, Dr. Ravikant S. Zirmite, Mr. Manish Bhalerao, Mrs. Manasi Shirurkar, (2025). Fake Twitter(X) Profile Detection. *Journal of Neonatal Surgery*, 14 (21s), 942-947.

### ABSTRACT

#### ABSTRACT

Individuals engage with social networking platforms for communication and interaction daily. People regularly create profiles on social networking websites where users actively connect and interact. can connect and communicate with one another at any time and from anywhere. This has been a boon as well as a bane. People of all ages spend much of their time on social networking websites. This leads to the creation and sharing of massive volumes of data on social networks all around the world. These reasons have contributed to the rise of fake users who prey on other social network users. A fake user is the one who creates a profile under a pseudonym and false information. These profiles are created for satirical reasons, to deceive and spread fake news and misinformation. The user profiles on social networks need to be categorised into fake and real. The categorisation may enable us to access the actual user profiles. This study provides a classification technique for identifying false Twitter accounts based on various user features like account follower and following count, posted tweets, gender, etc. using different algorithms like, Random Forest, SVM etc. and measure its accuracy.

**Keywords:** Twitter, Fake Profile, Trolling, Machine Learning, Social Media.

### INTRODUCTION

Twitter is a social networking platform that enables users to communicate with their followers by sharing their ideas, opinions, and thoughts. Unfortunately, people who make false profiles and participate in trolling frequently abuse this platform. A false profile is an accounts that has been created and established. with the purpose of misleading other users. A profile that is intended to provoke and upset people is known as a trolling profile. These profiles Could lead to negative consequences, including harm to a person's or organization's reputation and financial loss.

So, it is essential for preserving Twitter's credibility to identify bogus and trolling profiles. Because these profiles might mirror actual users, it may be challenging to recognise them. Furthermore, many of These profiles are created utilizing automated technologies, making it harder to detect them. As a result, there is a need for efficient methods to identify fake and trolling Twitter profiles.

### LITERATURE SURVEY

[1] Patxi Galan Garc, in his paper suggested that behind every trolling profile is a real profile. He selected 19 profiles for the study and gathered 1900 tweets, 100 tweets corresponding each user by employing various machine learning algorithms on WEKA toolkit, they were able to detect trolling profiles. They also implemented their research work on a real-life case of cyberbullying in school and were able to catch the person behind the fake profile.

[2] Kaubiyal employed a feature-based method to find fake Twitter profiles. He made use of 24 features in all, including the numbers off friends and followers, the numbers of retweets each tweet, tweet similarity, etc. They made use From publicly accessible datasets that included information from actual Twitter user profiles and timelines that was gathered through the Twitter API. It classified accounts as 'bots' and 'humans' with 80.8% accuracy for SVM and 95.3% for Logistic Regression.

[3] Using user-based features, content-based features, and each user's sentiment score, Monica attempted to identify bots from real-time data. Utilising the Twitter API, information on up to 200 tweets for each user was collected. She made use of the VADER (Valence Aware Dictionary and Sentiment Reasoner) lexical library. Scores greater than 0.05 were regarded as favourable, those less than 0.05 as unfavourable, and those between -0.05 and 0.05 as neutral. [4] Supraja et al. obtained almost 62 million user profiles on Twitter through social web crawling. They considered 33 different attributes of the profile and examined combination patterns among them to recognise primary set of fake profiles. According to the study, fraudulent profiles were almost usually made in groups and took less than 40 seconds to build. They also did profile picture and tweets analysis. They did a overall study of profile characteristics like creation and updation times, friends and followers and found that fake users tend to gather friends faster.

[5] Naman et al. theorized that real profiles have at least more than 30 followers. It is mostly based on psychological analysis of fake profiles and their creators. They concluded that fake accounts almost always set their age wrong, lie about their gender and even post stock image as profile picture. They suggest checking the email ids linked to the profile as well as their location to verify the profiles.

[6] In order to identify false Twitter profiles, Buket et al. devised a method that demonstrated how discretization affects Nave Bayes. They based their study on various profile attributes like follower and friend count, verified or not, profile image, average hashtags etc. It concluded that discretization increases the accuracy by 4.86%.

[7] By utilizing machine learning techniques such as Logistic Regressions and the Random Forest Algorithm, Ananya Dey et al. evaluated a method to distinguish between false and legitimate profiles on Instagram. They prepared the Kaggle dataset for training and testing by cleaning and analysing it. (Random Forest earned a precision score of 93.2% compared to 87.6% for Logistic Regression.) It considered features like followers, follows, posts, profile picture, nums/length username, etc. It concluded with higher accuracies for both the algorithms with Random Forest taking the lead by 1.7%.

[8] Dr. Suchita studied different techniques For identifying fake accounts on social media platforms media like Twitter, Instagram, etc. using SVM – Support Vector Machines, ANN – Decision Tree, KNN – K Nearest Neighbour, Naïve Bayes, Artificial Neural Network She states that Random Forest is mostly used for detection of fake and fraudulent accounts.

[9] This study suggests a method for identifying spam tweets based on their emotional content. Spammers are divided into four groups: those who create phoney content, those who employ URL-based spam detection, those who find spam in hot themes, and those who utilise false user identification. Different algorithms were used to study the detection, including Naive Bayesian, Support Vector Machines, Random Forest and K Nearest Neighbour, with the conclusion that Random Forest had the edge.

[10] Sowmya and Madhumita generated two module system namely i) Fake Profile detection – to separate fake and genuine users based on pre-defined rules. ii) Detection of Clone Profiles using Similarity Measure – uses an attribute and similarity measure to find clones. It uses C4.5 algorithms to detect a clone profile. The datasets were collected from MIB Projects. Similarity Measures performed more effectively.

## PROPOSED METHODOLOGY

The task of distinguishing fake profiles from the genuine ones is not a simple task. Various factors need to be considered. These include the twitter users' profile features.

A Machine Learning Model has to be created for the detection of fake profiles. The data for training and testing was downloaded from Kaggle. There were two files namely fake users and genuine users.

Since we already have the classes for the profile to be classified into, supervised machines learnings algorithms were employed used. The first task after data collection is making the data fit for use.

After data pre-processing, a model is created using Random forest and Support Vector Machine algorithm. Their performance is then evaluated and compared.

### *Pre- processing and feature selection:*

The data though good enough for use must be cleansed first. The data is checked for nan values, whitespaces, etc.

The data used was in two separate files namely fake users and real users. Both the files are merged. A new column is introduced in both files as a target column displaying whether the profile is fake or real. It contains 1 for fake users and 0 for real users.

```
def read_user_datasets():
    # Reads users profile from csv files
    real_users = pd.read_csv("users.csv")
    fake_users = pd.read_csv("fusers.csv")
    x = pd.concat([real_users, fake_users])
    x['geo_enabled'] = x['geo_enabled'].replace(np.nan, 0)
    y = len(fake_users)*[1] + len(real_users)*[0]
    return x,y
```

The data had a lang column specifying the language of the user. The language is translated into code number called lang\_code column. The gender of the user is predicted using the gender\_guesser library in python. It predicts the gender of the user based on firstname. A new column for sex\_code is created. Both columns have numerical values.

```
def predict_user_sex(name):
    d = gender.Detector(case_sensitive=False)
    first_name= str(name).split(' ')[0]
    sex = d.get_gender(u"{}".format(first_name))
```

```
gender_code_dict = {'female': -2, 'mostly_female': -1, 'unknown': 0, 'mostly_male': 1, 'male': 2}
code = gender_code_dict[sex]

return code
```

The feature extraction comes next.

### Description of features

The features that play a role in determining the profile classes are selected.

[6] user followers\_count, statuses\_count, listed\_count, favorites\_count, friends\_count

[2] used geo\_enabled along with the above stated features.

Table 1: Description of Features

Features	Description
statuses_count	Number statuses this user has updated
followers_count	Current number of followers for this account
friends_count	Number of Users who are engaged with this account
favorites_count	The total number of tweets liked by this user over the account's history
listed_count	The quantity from public lists this user belongs to
sex_code	User's gender
lang_code	Language belonging to the user
geo_enabled	Whether the user has enabled geolocation for his/her postings is indicated.

```
feature = ['statuses_count', 'friends_count', 'followers_count', 'favourites_count', 'listed_count', 'geo_enabled', 'sex_code', 'lang_code']
x = x.loc[:, features]
```

### Implementation

```
# splitting train and test data
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=45)
```

Popular supervised machines learning techniques include SVM and Random Forest.

### Support Vector Machines Classifier:

SVM effectively categorizes data points even in situations where they cannot be separated linearly by mappings the data into a high dimensional features space. The hyperplane, a simple line separates the data. It is called a decision boundary.

To remove the data sparsity which means biased or skewed data we use preprocessing.scale () from the sklearn library. It helps to standardize the data points.

### Random Forest Classifier:

Decision trees from a variety of subsets of the supplied dataset are included in a random forest. By averaging all of the decision tree forecasts, it helps to increase accuracy.

```
rf_classifier = RandomForestClassifier(n_estimators=300, max_depth=5, random_state=1)
XX_train, XX_test, yy_train, yy_test = train_test_split(x, y, test_size=0.20, random_state=44)
rf_classifier.fit(XX_train, yy_train)
train_predictions = rf_classifier.predict(XX_train)
prediction = rf_classifier.predict(XX_test)
```

When building a model on the available dataset it works as expected but when tested out on real data it may not perform as well. This is an overfitting problem. So, to understand the actual performance of our model we use validation dataset that is the test data which is taken from the whole data available but is yet unseen to the model. This is cross validation

We employ Stratified K-fold cross validation to solve the overfitting and under fitting issues. This K-fold cross validation variant has been improved. Data is divided into a specified number of folds using K-fold.

It trains on few of splits and tests on the remaining as given in the ratio during model creation. But Stratified K-fold ensures that these splits contain the same The volumes of trainings and testing data as specified in the ratio. Both rotate training and evaluation data among them.

Hyper parameters are defined as parameters that the user has deliberately established to control how the model learns.

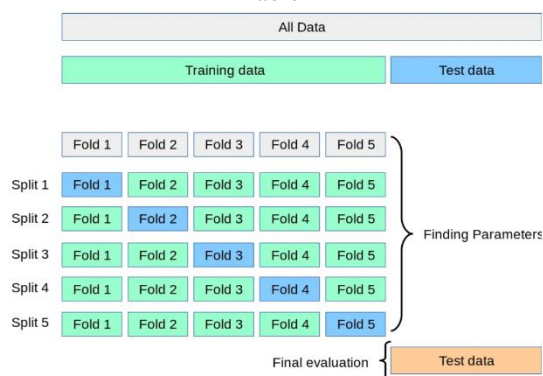
#### The hyper parameters used for SVM are:

1) For each incorrectly classified data point, the C parameter increases a penalty. The penalty is negligible for small values of the C hyper parameter, hence decision boundaries with greater separations from the data points are chosen at the expense of greater misclassifications. For large values, Support Vector Machine (SVM) tries to minimize the numbers of misclassified samples, and thus decision boundary has small margin.

2) The gamma parameter controls the influence distance of training points. With a low gamma value, more points are clustered together because there exists a greater radius of resemblance. Points within a specific class are very similar to each other when the gamma value is high. This causes overfitting of model.

The GridSearchCV () function is used for determined the ideal hyper parameter values. That is, it performs hyper parameter optimization using the provided C and gamma values. Once the best estimator is found, models is fit.

Table 2



#### Evaluation and Results:

This model is testing against the predicted and target values of the is fake column.

#### I) Confusion Matrix

Table 3: Confusion Matrix

SVM	Random Forest
[[243 53] [2 266]]	[[266 30] [1 267]]
<p>Fig. 1.1</p> <p>The confusion matrix shows that the model classifies 243 fake users correctly and 2 fake users incorrectly. It 266 real users correctly and 53 users incorrectly.</p>	<p>Fig. 2.1</p> <p>The confusion matrix displays that 266 fake users are correctly classified and 1 user incorrectly, 267 real users are correctly classified and 300 users incorrectly.</p>

#### II) AUC – ROC Curve

The classification model's performance is measured using the AUC-ROC curve at various threshold levels. Receiver Operating Curve represents a probability graph. It is plotted between 2 parameter True Positive Rate(TPR) and False Positive Rate(FPR)

$$\text{TPR} = \text{TP} / (\text{TP} + \text{FN}) \quad \text{Specificity} = \text{TN} / (\text{TN} + \text{FP})$$

$$\text{FPR} = \text{FP} / (\text{TN} + \text{FP}) = 1 - \text{Specificity}$$

where,

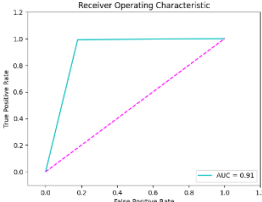
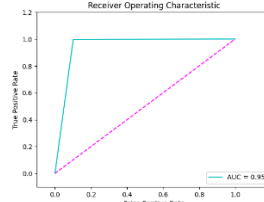
TP – True Positive, TN – True Negative, FP – False Positive, FN – False Negative

Area Within the ROC The two-dimensional area under the ROC is calculated by Curve and ranges from (0,0) to (1,1).

AUC calculates the binary classifier's performance across various thresholds and offers an aggregate metric. Its value is between 0 and 1. Values that are nearer 1 indicate that it can classify more fairly.

**AUC Value of the SVM is 0.91 and value of Random Forest is 0.95**

Table 4: AUC ROC Curve

SVM	Random Forest
FPR: [0. 0.17905405 1. ] TPR: [0. 0.99253731 1. ]	FPR: [0. 0.10135135 1. ] TPR: [0. 0.99626866 1. ] Train Accuracy: 94.85359361135758 Test Accuracy: 94.50354609929079
 <p>Fig. 1.2 Receiver Operating Characteristic</p>	 <p>Fig. 2.2 Receiver Operating Characteristic</p>

### 1) Report on SVM Classification

	Decision	all	score	report
ke	9	2	0	5
nuine	3	9	1	8
uracy			0	4
cro average	1	1	0	4
ighted average	2	0	0	4

### Learning Curve for SVM

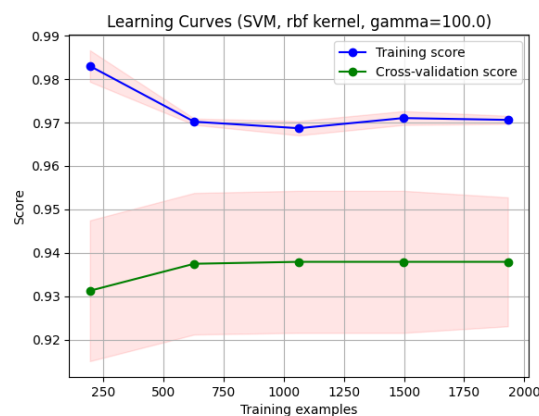


Fig. 1.3

### FUTURE SCOPE

Identifying fake Twitter profiles is a serious issue that requires attention. Current methods come with certain constraints. due to insufficiently labelled data for training the models.

We have achieved some per precision using the Random Forest Classifier and some per performance with the Support Vector

#### Machine Algorithm.

Using a hybrid approach like network analysis, content-based techniques like sentiment analysis on user tweets and Natural Language Processing - NLP or Artificial Neural Network – ANN methodologies would help in improving the correctness of fake profile detection.

The scope This pertains to not only limited to Twitter, as the techniques and approaches developed/used can Can be applied to other social media Platforms(SMPs) as well. (Moreover, the topic has applications across different domains, such as computer science, language processing, network analysis and sociology.

#### CONCLUSION

In this study, a feature-based technique is utilized to detect fake profiles on Twitter. It incorporates the elements of the profile that contribute to its authenticity. Random Forest outperformed with respect to detecting fake profiles than SVM. The model could be tested in real world scenarios.

#### REFERENCES

- [1] Patxi Galan-Garc, Jos Gaviria de la Puerta, Carlos Laorden Gmez, Igor Santos, Pablo Garca Bringas; Supervised machine learning for the detection of troll profiles in twitter social network: application to a real case of cyberbullying, Logic Journal of the IGPL, Volume 24, Issue 1, 1 February 2016, Pages 4253
- [2] Jyoti Kaubiyal, Ankit Kumar Jain; A Feature Based Approach to Detect Fake Profiles in Twitter, 2019, Australia, Association for Computing Machinery
- [3] C Monica, N Nagarathna; Detections of Fake Tweets Using Sentiment Analysis, SN Computer Science, 18 March 2020
- [4] Supraja Gurajala, Joshua S White, Brian Hudson, Brian R Voter and Jeanna N Matthews; Profile characteristics of fake Twitter accounts, Big Data & Society, 2016
- [5] Naman Singh, Tushar Sharma, Abha Thakral, Tanupriya Choudhary; Detection of Fake Profiles in Online Social Networks Using Machine Learning, 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE-2018) Paris, France 22-23 June 2018
- [6] Buket Ersahin, Ozelm Aktas, Denis Kilmc, Ceyhun Akyol; 2<sup>nd</sup> International Conference on Computer Science and Engineering, 2017, IEEE
- [7] Ananya Dey, Hamsashree Reddy, Manjistha Dey, Niharika Sinha; Detection of Fake Accounts in Instagram Using Machine Learning, International Journal of Computer Science & Information Technology(IJCSIT) Vol 11, No 5, October 2019
- [8] Dr. Suchita Amey Bhovar; A Study of Different Methodologies to Detect Fake Account on Social Media using Machine Learning; International Journal of Science and Research(IJSR), 2022
- [9] Chekuri. Sri Divya Naga Durga, S. Vennela; Fake Profile Detection in Using Machine Learning; Journal of Engineering Sciences, Vol 13 Issue 2022
- [10] Sowmya P, Madhumita Chatterjee; Detection of Fake and Clone accounts in Twitter using Classification and Distance Measure Algorithms; International Conference on Communication and Signal Processing, July 2022