# Performance Examination and Feature Selection on Sybil User Data using Recursive Feature Elimination

**Dheeraj Sonkhla, Manu Sood**

***Abstract*: *Machine Learning (ML) research greatly helps in predicting model-based outcomes with high levels of accuracy based upon the training and testing of the models through the datasets. The social networks constitute one of the domains where ML can be used effectively to ensure the authenticity and security of the valid users. With the increase in usage of Online Social Networks (OSNs), the cases of spam and malicious activities can be found in abundance and Sybil nodes pose one such kind of safety and security hazard. Sybil account detection is not an easy task since they mimic the actual behavior of human accounts up to a great extent. In this paper, we look at one such scenario of Sybil accounts on the OSN, Twitter where machine leaning models have been used to train the machine with the existing datasets so as to be able to detect these malicious users before they can bring harm to the normal communication of the genuine users. Since the datasets used are so vast, the process of feature selection has been carried on the datasets as part of pre-processing before the actual classification as it assists in enhancing the model performance. Support Vector Machine–Recursive Feature Elimination (SVM-RFE) and Logistic Regression–Recursive Feature Elimination (LR-RFE) techniques have been used in this study for the selection of significant features. The classification model is trained on the selected features using Random Forest (RF) and K-Nearest Neighbor (KNN) algorithms. We also analyzed the biasing effects of fake accounts on the human accounts datasets during the process of features selection and classification. It has been shown that the RF algorithm outperformed KNN on the feature sets selected through SVM-RFE and LR-RFE.***

***Index Terms*: *feature selection, k-nearest neighbor classifier, logistic regression-recursive feature elimination, machine learning.***

## I. INTRODUCTION

The data is being generated today at a massive rate and the sources contributing to this data explosion exist everywhere, whether it is online shopping, social networking, emailing, data storage or other things that we do online. The traditional methods of processing such data are being proven grossly inadequate to dig out useful information out of this huge data. Machine Learning (ML) provides us one of the easiest ways to dig out some sense out of these large amounts of data in simple and efficient ways, specifically from the prediction point of view. ML helps in extracting the most significant data from the piles of data [1,2].

**Revised Manuscript Received on July 10, 2019.**

**Dheeraj Sonkhla**, Department of Computer Science, Himachal Pradesh University, Shimla, India.

**Manu Sood**, Department of Computer Science, Himachal Pradesh University, Shimla, India.

Machine learning is a subdomain of Artificial Intelligence (AI) which mainly focuses on creating machines capable of learning from data these use. There are mainly four machine learning approaches to make these machines learn: supervised learning, unsupervised learning, reinforcement learning, and semi-supervised learning. The techniques involved in supervised learning are classification and regression. Clustering is a popular technique of unsupervised learning. To predict the discrete responses, classification is used. In the case of prediction of continuous responses, regression is used. Important abbreviations used in this paper are listed in Table 1.

TABLE 1. LIST OF ABBREVIATIONS

| Abbreviation | Explanation |
|---|---|
| AI | Artificial Intelligence |
| AUC | Area Under the Curve |
| KNN | K-Nearest Neighbor |
| LR | Logistic Regression |
| LR-RFE | Logistic Regression-Recursive Feature Elimination |
| ML | Machine Learning |
| OSN | Online Social Network |
| PTR-MS | Proton Transfer Reaction-Mass Spectrometry |
| RF | Random Forest |
| RFE | Recursive Feature Elimination |
| ROC | Receiver Operating Characteristics |
| SVM-RFE | Support Vector Machine-Recursive Feature Elimination |

There are various steps involved in machine learning depending upon the model being developed for prediction. The first step in this process is data gathering which is a very important step. The quality and quantity of data collected during this step decide how good your final predictive model will be. The second step in ML is to prepare the data, often known as pre-processing of data. The further sub-steps involved in this pre-processing are cleansing, transformation, normalizing, error-correction, and scaling of data [3]. There are different techniques available to perform the pre-processing of data. Subsequently, the next sub step undertaken for building the learning model is feature selection (FS), which is a process of selecting/extracting the meaningful features (contributing most to the accurate prediction). It can also be described as a process which helps in eliminating the unnecessary features from the total features set of the data.

Next comes the selection of the ML model or technique. There are many models which have been developed over the years for solving various kinds of problems. In our specific case, the problem is related to the classification since we have a labeled dataset.

To build the prediction model based upon this dataset, we have used the Random Forest (RF) and K-Nearest Neighbor (KNN) algorithms in this study. After selecting the ML technique/model, one needs to train the proposed prediction model and test the trained model. To train our proposed model, we split our pre-processed data into 80/20 or 70/33 ratio of training/testing.

Once the model has been trained, one can use it to evaluate the model using test data to check whether it works as per the requirement related to performance. In case, this evaluation during testing does not go so well, the model requires the parameter tuning. Then again evaluation is carried out for the performance of the proposed model. Once the desired performance is achieved, the model is ready to make predictions.

Different varieties of datasets are available in specific domains for the purpose of training the specific models. One of the areas where datasets are readily available is that of networking especially the online/offline social networking. In networking, a specific type of attack named Sybil attack with fake/fabricated identities poses a major threat to the wireless sensor networks or other categories of ad-hoc networks [4]. It is a malicious device/user which uses multiple identities to act as the legitimate nodes in the network. The multiple fake identities are known as the Sybil nodes. This attack was first explained by John R. Douceur [5]. Much work has been carried out to develop the techniques of defense against Sybil attacks in ad-hoc networks [6,7,8,9]. The concept of Sybil accounts in Online Social Networks (OSNs) has emerged lately in different ways. This study has been carried out to identify the Sybil accounts in OSNs.

One of the biggest contributors to the exponential growth of data is Online Social Networks (OSNs). With the technological advancement, the ease of usability for these OSNs is increasing which in turn results in an increase in a number of users. There are billions of active users monthly on these OSNs [1,10]. Instagram, Facebook, Twitter, WhatsApp, and YouTube are among the most famous online social networks which are actively being used by billions of people [11,12].

Twitter originally started as a microblogging site but with the increase in its usage, it became a kind of information publishing platform almost for free. With such widespread access and easy-to-use interfaces, it became a suitable domain for Sybil accounts. Sybil accounts are generally fake accounts [13]. These accounts are primarily created for increasing the followers of a targeted account. The fake follower accounts can be created in large numbers with human intervention, or automatically by Intelligent Agents called Bots or by both, after which these accounts are sold to the users desirous of getting their number of followers increased. There are plenty of inappropriate uses of these kinds of Sybil accounts. Some people may use Sybil accounts to gain the trust of other users or to gain popularity. Others may use these fake followers to influence the judgment of people by a large number of fake reviews through these accounts. Some companies use these accounts for the purpose of marketing. The URL to their products can be added to the tweets generated through these accounts. The number of followers for a particular account can easily influence the opinions of others. By increasing the followers, it can show the targeted account(s) as trustworthy and attract other genuine followers too [14].

In this paper, the authors have explored the possibility of accurately identifying any twitter account whether it is a fake or genuine human account through classification with the help of ML. If one can know in advance whether an account is real or fake, it can help in avoiding the opening of unnecessary, spam, harmful, inappropriate or such tweets [15]. The fake accounts normally act as human accounts by posting tweets or behaving in a manner similar to that of a genuine one. Normally, these fake accounts retweet certain kind of content of targeted account to publicize it or market various products from other social sites by posting URLs in between the tweets. This work mainly focuses on finding a certain set(s) of features which can be used to identify whether an account is fake or not based on its profile data.

### A. Objectives

The main objectives of this research work are as follows:

1) To analyze the importance of feature selection process in reference to the datasets under consideration.
2) To select the best features set by using Support Vector Machine–Recursive Feature Elimination (SVM-RFE) and Logistic Regression–Recursive Feature Elimination (LR-RFE).
3) To analyze the effect of biasing of fake accounts on human accounts in the feature selection process.
4) To analyze the results of classification using Random Forest and K-Nearest Neighbor classifiers with the selected features and to study the effects of biasing in classification.

### B. Road Map

This paper is partitioned into five sections. Section II gives a description of the datasets used for this study. The types of data included in the datasets and other necessary details have been provided. Section III discusses the methodology used for achieving the overall objectives of this work. Section IV provides the results of experiments and their analysis. The work has been concluded in section V while providing a pointer towards future work. In addition, three research questions have also been answered with some explanation in Sections III and IV.

## II. DATASET

In this section, a brief description of the datasets used for experiments in this research work is given. Table 2 shows the details of the datasets we have used in this study. While searching for the dataset containing Sybil accounts, we came across a research study [14],

which did some considerable amount of work on these datasets. We are very grateful to the authors of [14] who have permitted us to conduct our research work on these datasets which we have acquired from them.

TABLE 2. DETAILS OF DATASET

| Nature of Accounts | Sr. No. | Dataset | Number of Accounts |
|---|---|---|---|
| Human | 1 | E13 (elezioni2013) | 1481 |
| | 2 | TFP (TheFakeProject) | 469 |
| Fake | A | FSF (fastfollowerz) | 1169 |
| | B | INT (intertwitter) | 1337 |
| | C | TWT (twittertechnology) | 845 |

The datasets contain Twitter data for user accounts. There are total of five datasets, two of which contain the data for human accounts while the other three datasets contain the fake accounts data. It is pertinent to note here that the number and labels of features in the datasets in both of these cases are exactly the same.

TABLE 3. COMPLETE FEATURES SET IN PROFILE DATA OF USER

| | | |
|---|---|---|
| Id | Location | Profile background Tile |
| Name | Default profile | Profile sidebar fill color |
| Screen name | Default profile image | Profile background image url |
| Status count | Geo-enabled | Profile background color |
| Followers count | Profile image url | Profile link color |
| Friends count | Profile banner url | Utc offset |
| Favorites count | Profile use background image | Protected |
| Listed count | Profile background image url https | Verified |
| Created at | Profile text color | Description |
| url | Profile image url https | Updated |
| Language | Profile sidebar border color | Dataset |
| Time zone | | |

The authors [14], themselves have collected the human datasets personally for their own exploration purpose and the authors of this paper have also used the same datasets for our experiments in this paper with due permission from these authors. They have verified every human account. The dataset named E13 was collected during the elections and dataset TFP (TheFakeProject) was a project initiated to collect the data for human accounts as a part of academic study. The three datasets containing fake account details were bought online. The folder pertaining to each of these datasets further consisted of datasets belonging to profile data of the user, tweet data, and followers details. The authors of this paper have used the dataset containing profile data only which consisted of 34 features in total. All these features are listed in Table 3.

TABLE 4. DATASETS USED IN STUDY

| Dataset | | Cases | Number of Samples |
|---|---|---|---|
| Dataset 1 | E13-FSF | C-1A | 2650 |
| Dataset 2 | E13-INT | C-1B | 2818 |
| Dataset 3 | E13-TWT | C-1C | 2326 |
| Dataset 4 | TFP-FSF | C-2A | 1638 |
| Dataset 5 | TFP-INT | C-2B | 1806 |
| Dataset 6 | TFP-TWT | C-2C | 1314 |

To conduct this study we have made a total of six sets of profile data. Each set contains the data for human and fake accounts. The detail of these six sets is given in Table 4 with the number of samples (accounts) in each set. Later we prepared the four cases of each dataset to study the biasing effect. The cases for C-1A are detailed in the Table 5. Similar cases were designed for C-1B, C-1C, C-2A, C-2B, and C-2C. The biasing effects were checked using these cases during the process of feature selection and classification.

## III. EXPERIMENT SETUP

### A. Data Pre-processing

Data pre-processing mainly includes the cleaning, scaling and transformation of data. The cases in which problem of NaN (Not-a-Number), missing values, inconsistent values, infinity values are dealt, known as data cleaning [3]. Data scaling normally includes the standardization of feature ranges (on their domains) available in data, also known as normalization of data.

In the present dataset, the problem of missing values was present. Since the dataset used in this study included the values in the form of binary and the missing values just represent the absence of data, so these missing values were replaced with the zero. There was also the case where whole feature values were missing and some redundant features were also available. The "verified" and "protected" features contained no values and thus we have simply dropped those features. Two more features contained the redundant values which were later dropped in the pre-processing process. After the process of pre-processing, we got the subset of 28 features from the primary dataset of 34 features. The subset thus obtained was further processed by using the feature selection techniques as explained in the next section. The steps of the methodology we followed during this study are shown in Fig. 1.

For the purpose of implementation of various algorithms in this study, we have used the Python language. An open-source web-based application known as Jupyter Notebook is used to run the python codes [16].

### B. Feature Selection

In this section, the following research question is answered.

**RQ1.** *What is feature selection and its importance for the classification process?*

The main reason for the large size of datasets is huge number of features these contain. These features are mainly used for storing the information about the target variables. Common thinking suggests that more the number of features, better the performance of the predictive model. But this is not the case always. There are features in the dataset which contribute way less to the predictive modeling than the vital ones. They may have even no contribution at all. They are generally irrelevant, noisy, and redundant features [17]. The resulting effects of these features are that they decrease the efficiency of the final predictive model. Common troubles caused by these features are:

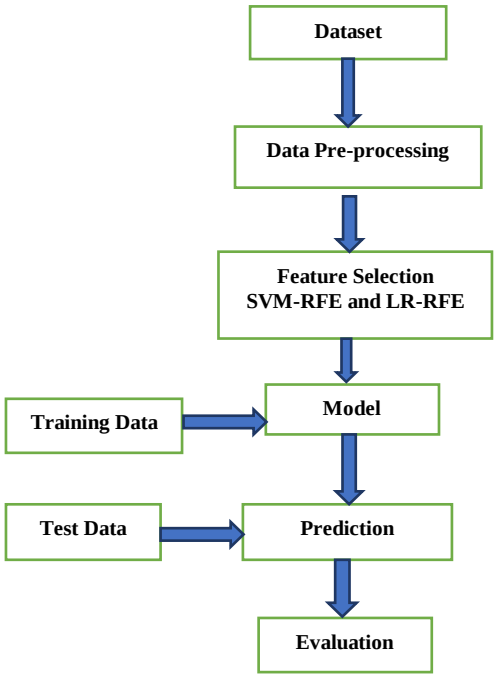


Fig. 1. The methodology followed

1) They mostly act as noise which in turn cause the ML model to perform poorly.
2) The addition of such features results in more training time.
3) Needless allocation of resources for irrelevant features is also required.

The solution to all these problems is feature selection (FS). The process of feature selection helps in removing these noisy, irrelevant and redundant features [18]. It selects the most significant features which contribute to the target variable most and helps in making the predictive model more efficient. FS also helps in reducing the overfitting. According to [19], the way FS algorithms interacts with the classifiers helps in dividing the algorithms loosely into three categories: filters, wrappers and embedded.

The bases on which filter methods perform the selection of features is correlation criteria, correlation of feature(s) with the output variable [19]. They use some mathematical functions like chi-square to do the evaluation. They generate the ranking of features using this evaluation. One of the advantages of filter methods is that they work faster than wrapper methods. The Univariate and Correlation Matrix with Heatmap are examples of filter methods. The univariate method uses the chi-square test to produce a ranking list of features. Correlation defines the relation of features with each other or target variable. Wrapper methods, on the other hand, prepare the feature subset with the maximum accuracy provided by the classifier used. For evaluating the subset, a performance measure like the accuracy of the classifier is used. Recursive feature elimination is an example of a wrapper method. The performance of wrapper methods mainly depends on the classifier. So generally, the computation speed of filter methods is better than the wrapper methods. The accuracy provided by wrapper methods is better. In case of a large number of features, filter methods are better in performing the selection of features. But when we have fewer features like near to hundred only, wrapper methods work better. This is the reason why we choose RFE for our work. Embedded methods judge the importance of features such as those which contribute more to the accuracy of the model [17].

We checked the accuracy of Logistic Regression (LR) model for the features before and after FS. The result shows us that FS provides a very big help in increasing the effectiveness of the model. The accuracy before the FS was 0.498 whereas after the FS, it increased to 0.850, where 0 is minimum and 1 is the maximum value.

**Recursive Feature Elimination method (RFE).** This method recursively considers the smaller and smaller number of features so as to select them [20]. The predictor is first trained on the primary set of features to get the importance of each one. After this, the features with the least importance are removed from the subset and process continues recursively until the desired number of features in the subset is reached. The pseudo code for the RFE algorithm is given in Fig. 2. The RFE method of FS is better than filter methods as it gives the ranking to important features by eliminating the less important features recursively. We can use a classifier during the process of FS which will help us in selecting the best features. The results of ranking include the numeric 1 for selected features and while the ranking of other (not selected) features is given using numeric values ranging from 2 to n, where n is equal to the number of all features inputted to the RFE algorithm.

The study in [17] was conducted to analyze the correlation in gas sensor data. It included the study of RFE using SVM-RFE. A similar kind of study was conducted by [8]. The study was conducted to analyze the Proton Transfer Reaction-Mass Spectrometry (PTR-MS) using RFE with Random Forest (RF-RFE). A comparison was performed between the RF-RFE (Random Forest-Recursive Feature Elimination) and SVM-RFE.

Some of the advantages of using RFE are as follows:

1) It can be implemented on different datasets easily.
2) Different classifiers can be used to perform the FS as it supports various classifiers.
3) It can be applied to a large set of features.

The choice for the feature selection technique for online datasets needs to be exercised very carefully since the computation time required in RFE is quite high as it uses the classifiers to rank the features.

```
Inputs:
    Training set T
            Set of n features F = {f 1. ..., f k}
            Ranking method M (T, F)
Outputs:
        Ranking R
Code:
        Repeat for i in {1 : n}
            Rank set F using M(T, F)
            f * ← feature with the last ranking in F
            R (p – i + 1) ←f *
            F ←F – f *
```

Fig. 2. Pseudo code for recursive feature elimination (RFE) [20]

**RQ 2.** *Does the biasing of human accounts with Sybil accounts affect the FS process?*

Support Vector Machine–Recursive Feature Elimination (SVM-RFE) and Logistic Regression–Recursive Feature Elimination (LR-RFE) have been used for FS in this paper. The FS has also been performed when we biased the datasets. After analysis of the result, it can be seen that for LR-RFE, the output changes when the number of fake followers was considerably fewer. The results for LR-RFE are given in the Table 5. SVM-RFE is another popular FS algorithm, widely used for the feature selection which we have also used in addition to LR-RFE in this paper.

In both of these algorithms, 15 features out of the original 28 features were selected. But the selected features for C-1A$_{25}$ differs from the ones selected in other cases.

In this case, the number of fake followers in the dataset were only 25% of their total number of fake followers. The value "1" for a feature in Table 5 means that the feature is selected. Other values show the ranking of features and the smaller values represent a higher ranking of the features under consideration.

The cases were designed to analyze the possible effects of biasing in the FS process.

The biasing is carried out by varying the number of fake accounts in the dataset and keeping the number of human accounts constant for all cases. The four cases that we have used for the combination of datasets C-1A are shown in Table 6. C-1A$_{25}$ means that the number of fake followers used for the experiment were the 25 percentage of total number of fake followers. Same logic applies to the cases C-1A$_{50,}$ C-1A$_{75}$ and C-1A$_{100}$.

TABLE 6. CASE DETAILS

| Case | Total Accounts | Fake | Human |
|---|---|---|---|
| **C-1A$_{25}$** | 1375 | 275 | 1100 |
| **C-1A$_{50}$** | 1650 | 550 | 1100 |
| **C-1A$_{75}$** | 1925 | 825 | 1100 |
| **C-1A$_{100}$** | 2199 | 1099 | 1100 |

So from Table 5, it is clear that when the number of fake followers was considerably lesser then the human accounts (Case C-1A$_{25}$), the FS results vary for LR-RFE, where this was not the case for SVM-RFE. For all the four cases C-1A$_{25,}$ C-1A$_{50,}$ C-1A$_{7,}$ and C-1A$_{100,}$ the selected features we got from SVM-RFE as the output of this technique were the same.

### C. Models

Random forest (RF) classifier is commonly used for the classification and regression problems. There are mainly two steps in RF algorithm, first is to create a forest of random trees, and the second step is to make the prediction from the votes of trees created in the first step to determine the output [3]. The same Random Forest algorithm can be used for the problems of classification and regression. In many cases, the accuracy for RF is usually higher than other classifiers. Even from the results of [14], it can be seen that the prediction performance of RF is better than the other classifiers they have used. So we have used RF as one of the two classifiers for our prediction model. The other classifier that we have used is K-Nearest Neighbor (KNN). The principle on which KNN is based is that the objects within a dataset will generally exist in close proximity to the other objects with similar properties [21]. The objects categorized by KNN are based on the classes of their nearest neighbors in the dataset. This algorithm is faster to train and can also be used for the multi-class classification problems. Due to its simple implementation and less training time we have selected KNN for our other classifier.

While training and testing the models the ratio of 70/30 were used, 70 for training and 30 for testing.

### D. Prediction and Evaluation Criteria

While considering the two classes a) fake and b) human, the experiment was conducted for each case mentioned above. To evaluate the final outcomes of these experiments some metrics were considered based on standard indicators, namely [13]:

**True Positive (TP).** Number of fake followers identified as fake.

**True Negative (TN).** Number of human followers identified as human.

**False Positive (FP).** Number of human followers identified as fake.

**False Negative (FN).** Number of fake followers identified as human.

The matrix called a confusion matrix, shown in Table 7 easily brings out the meaning of each indicator. The instances of the predicted class are shown by the columns while the actual class instances are depicted by the rows.

**Evaluation Metrics.** To evaluate the final results we considered the five standard evaluation metrics. These metrics are [14]:

*Accuracy.* Ratio of predicted true results in the population, that is , *(TP+ TN)/(TP+TN+FP+FN)*

*Precision.* Ratio of identified positive cases which were indeed positive, that is, *TP/(TP+FP)*

*Recall.* Ratio of real positive cases that are indeed identified positive, that is, *TP/(TP+FN)*

*F-Measure.* it is harmonic mean of recall and precision, that is, (2•precision•recall)/(precision+recall)

*Matthew Correlation Coefficient (MCC).* Estimates the correlation between the identified class and the real class of samples, given as

$$(TP\cdot TN-FP\cdot FN)/\sqrt{((TP+FN)(TP+FP)(TN+FP)(TN+FN))}$$

TABLE 7. CONFUSION MATRIX

| Actual Class | Predicted Class | |
|---|---|---|
| | Human | Fake |
| Human | TN | FP |
| Fake | FN | TP |

**Overfitting** of a model occurs when one trains a model too well on their training set. This results in a good accuracy during training but when we run the same model for testing, the accuracy we get is not up to the mark.

When we need to analyze and visualize the classification performance of a model we use the AUC (Area Under the Curve)-ROC (Receiver Operating Characteristics) curve.

**TABLE 5. FS RESULTS FOR LR-RFE**

| CASES | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10 | F11 | F12 | F13 | F14 | F15 | F16 | F17 | F18 | F19 | F20 | F21 | F22 | F23 | F24 | F25 | F26 | F27 | F28 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C-1A$_{25}$ | 14 | 1 | 1 | 5 | 1 | 6 | 1 | 9 | 1 | 12 | 1 | 10 | 1 | 1 | 13 | 8 | 7 | 1 | 1 | 4 | 1 | 1 | 11 | 2 | 3 | 1 | 1 | 1 |
| C-1A$_{50}$ | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 10 | 1 | 1 | 12 | 13 | 14 | 9 | 1 | 7 | 1 | 6 | 1 | 8 | 11 | 3 | 5 | 4 | 1 | 1 |
| C-1A$_{75}$ | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 10 | 1 | 1 | 12 | 13 | 14 | 8 | 1 | 7 | 1 | 5 | 1 | 9 | 11 | 3 | 6 | 4 | 1 | 1 |
| C-1A$_{100}$ | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 10 | 1 | 1 | 12 | 13 | 14 | 5 | 1 | 8 | 1 | 6 | 1 | 9 | 11 | 3 | 7 | 4 | 1 | 1 |

It is an important metric used to analyze the model performance. It is also written as AUROC (Area Under the Receiver Operating Characteristics). ROC is a probability curve and used to show the performance of the classification model at various probability thresholds. AUC is used to represent the measure or degree of separability, to which extent model is able to differentiate between the classes. ROC curve is plotted for the values of True Positive Rate (TPR) and False Positive Rate (FPR) where TPR is represented on y-axis and FPR on the x-axis. The range of threshold values is 0.0 to 1.0

*True Positive Rate.* It is equivalent to Recall. It measures the proportion of accurately identified actual positives. So, it is defined as: TP/(TP+FN)

*False Positive Rate.* It measures the proportion of accurately identified actual negatives. It is defined as: FP/(FP+TN).

Higher the curve for AUC better will be our prediction model at identifying the fake twitter followers as fake and human accounts as human. For an excellent model, the AUC value is near '1', which means the degree of separability is quite good whereas for a poor model this value will be near the '0'. The meaning of '0' AUC is that it has almost the worst measure of separability. In any case, the probability of getting zero is quite low as it means that it identifies every fake account as human and every human as a fake account. And in case we have AUC approximately or equal to '0.5' then that means that our model cannot distinguish between fake and human.

## IV. RESULTS AND ANALYSIS

**RQ 3.** *Does the biasing of Sybil accounts affect the process of classification?*

The datasets for experiments were prepared on the basis of selected features that we get after FS techniques. Prediction model was then trained on these datasets with Random Forest and KNN algorithms. The model was tested on total 6 datasets with 24 cases with varying number of fake follower accounts. We prepared these cases by varying the number of fake Twitter followers to test the biasing on the performance of the prediction model. The results of experiments are given in Table 8 and Table 9. It includes the values of evaluation metrics for all cases. Both models were executed for every case and results were collected.

From the results of Table 8 and 9, we can easily compare the performance to both predictive models. We clearly see that the prediction results for both classifiers are excellent. In Table 8, the experimental results for Random Forest are given. The highest value for every metric is shown in bold. The values of precision and recall reached to the maximum value i.e. '1'. For four datasets 2, 3, 4, and 6, we get the accuracy near 0.99. The maximum accuracy which we got was 0.9986 for case C-1B$_{75}$. The overall values of MCC were around 0.99 except for the datasets 1 and 5.

The results of KNN are summarized in Table 9. The maximum accuracy which we get from the KNN was 0.9959, which was as good as RF. The best results we get with KNN are for the dataset 4, case C-2A$_{25}$.

For the effect of biasing in the datasets, we can say that both the models work better when the number of fake accounts is nearly equal to real accounts. The detection accuracy of the model varies with the number of sybil accounts. To check the overfitting of our model we have split our datasets in different ratios for training and testing. In our first tests, we have used the ratio of 70/30 for training and testing respectively. For overfitting testing, we have analyzed the model by using the ratio of 80/20, 60/40 and even 50/50. The results show that both the models work fine for all cases.

After observing the results, we can say that both classifiers distinguish the fake twitter followers from the human almost correctly except for a few cases.

To visualize the performance for our models we have plotted the ROC and AUC curves. They help us in understanding the results better. Fig. 3 and Fig. 4 shows the ROC curve for Random Forest and KNN for the values of TPR and FPR. By observing the values of probability thresholds for both (RF and KNN), it is clear that the value of true positive is greater than the false positive. Fig. 5 provides the AUC-ROC curve for both classifiers, RF and KNN. From this figure, we can see that the value of AUC for RF is much near to '1' as compared to the KNN. This means that the performance of our model using RF is better than the KNN for the set of selected features of our datasets.

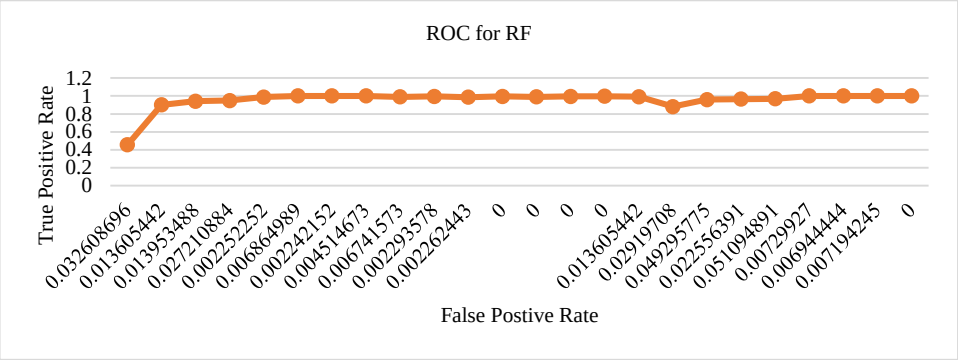


Fig. 3. ROC curve for Random Forest

TABLE 8. RESULTS OF RANDOM FOREST ALGORITHM FOR A TOTAL OF 24 CASES

| Datasets | Cases | Random Forest | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | TP | FN | FP | TN | Accuracy | Precision | Recall | F Measure | MCC |
| Dataset 1 | C-1A$_{25}$ | 50 | 60 | 15 | 445 | 0.8684 | 0.7692 | 0.4545 | 0.5714 | 0.5239 |
| | C-1A$_{50}$ | 118 | 13 | 6 | 435 | 0.9668 | 0.9516 | 0.9008 | 0.9255 | 0.9047 |
| | C-1A$_{75}$ | 193 | 12 | 6 | 424 | 0.9717 | 0.9698 | 0.9415 | 0.9554 | 0.9349 |
| | C-1A$_{100}$ | 244 | 13 | 12 | 429 | 0.9642 | 0.9531 | 0.9494 | 0.9513 | 0.9230 |
| Dataset 2 | C-1B$_{25}$ | 88 | 1 | 1 | 443 | 0.9962 | 0.9888 | 0.9888 | 0.9888 | 0.9865 |
| | C-1B$_{50}$ | 183 | 0 | 3 | 434 | 0.9952 | 0.9839 | **1.0000** | 0.9919 | 0.9885 |
| | C-1B$_{75}$ | 262 | 0 | 1 | 445 | **0.9986** | 0.9962 | **1.0000** | 0.9981 | **0.9970** |
| | C-1B$_{100}$ | 352 | 0 | 2 | 441 | 0.9975 | 0.9944 | **1.0000** | 0.9972 | 0.9949 |
| Dataset 3 | C-1C$_{25}$ | 99 | 1 | 3 | 442 | 0.9927 | 0.9706 | 0.9900 | 0.9802 | 0.9758 |
| | C-1C$_{50}$ | 208 | 1 | 1 | 435 | 0.9969 | 0.9952 | 0.9952 | 0.9952 | 0.9929 |
| | C-1C$_{75}$ | 300 | 4 | 1 | 441 | 0.9933 | 0.9967 | 0.9868 | 0.9917 | 0.9861 |
| | C-1C$_{100}$ | 392 | 2 | 0 | 452 | 0.9976 | **1.0000** | 0.9949 | 0.9975 | 0.9953 |
| Dataset 4 | C-2A$_{25}$ | 102 | 1 | 0 | 139 | 0.9959 | **1.0000** | 0.9903 | 0.9951 | 0.9916 |
| | C-2A$_{50}$ | 212 | 1 | 0 | 129 | 0.9971 | **1.0000** | 0.9953 | 0.9976 | 0.9938 |
| | C-2A$_{75}$ | 312 | 1 | 0 | 129 | 0.9977 | **1.0000** | 0.9968 | 0.9984 | 0.9946 |
| | C-2A$_{100}$ | 392 | 3 | 2 | 145 | 0.9908 | 0.9949 | 0.9924 | 0.9937 | 0.9767 |
| Dataset 5 | C-2B$_{25}$ | 59 | 8 | 4 | 133 | 0.9412 | 0.9365 | 0.8806 | 0.9077 | 0.8655 |
| | C-2B$_{50}$ | 120 | 5 | 7 | 135 | 0.9551 | 0.9449 | 0.9600 | 0.9524 | 0.9099 |
| | C-2B$_{75}$ | 191 | 7 | 3 | 130 | 0.9698 | 0.9845 | 0.9646 | 0.9745 | 0.9378 |
| | C-2B$_{100}$ | 250 | 8 | 7 | 130 | 0.9620 | 0.9728 | 0.9690 | 0.9709 | 0.9163 |
| Dataset 6 | C-2C$_{25}$ | 91 | 0 | 1 | 136 | 0.9956 | 0.9891 | **1.0000** | 0.9945 | 0.9909 |
| | C-2C$_{50}$ | 172 | 0 | 1 | 143 | 0.9968 | 0.9942 | **1.0000** | 0.9971 | 0.9936 |
| | C-2C$_{75}$ | 265 | 0 | 1 | 138 | 0.9975 | 0.9962 | **1.0000** | 0.9981 | 0.9945 |
| | C-2C$_{100}$ | 350 | 0 | 1 | 141 | 0.9980 | 0.9972 | **1.0000** | **0.9986** | 0.9951 |

TABLE 9. RESULTS OF KNN ALGORITHM FOR A TOTAL OF 24 CASES

| Datasets | Cases | KNN | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | TP | FN | FP | TN | Accuracy | Precision | Recall | F Measure | MCC |
| Dataset 1 | C-1A$_{25}$ | 24 | 33 | 50 | 401 | 0.8366 | 0.3243 | 0.4211 | 0.3664 | 0.2775 |
| | C-1A$_{50}$ | 52 | 83 | 2 | 435 | 0.8514 | 0.9630 | 0.3852 | 0.5503 | 0.5527 |
| | C-1A$_{75}$ | 126 | 62 | 1 | 446 | 0.9008 | 0.9921 | 0.6702 | 0.8000 | 0.7624 |
| | C-1A$_{100}$ | 215 | 39 | 53 | 391 | 0.8682 | 0.8022 | 0.8465 | 0.8238 | 0.7193 |
| Dataset 2 | C-1B$_{25}$ | 85 | 0 | 31 | 417 | 0.9418 | 0.7328 | **1.0000** | 0.8458 | 0.8259 |
| | C-1B$_{50}$ | 156 | 0 | 48 | 416 | 0.9226 | 0.7647 | **1.0000** | 0.8667 | 0.8280 |
| | C-1B$_{75}$ | 287 | 0 | 28 | 393 | 0.9605 | 0.9111 | **1.0000** | 0.9535 | 0.9222 |
| | C-1B$_{100}$ | 340 | 0 | 31 | 424 | 0.9610 | 0.9164 | **1.0000** | 0.9564 | 0.9241 |
| Dataset 3 | C-1C$_{25}$ | 97 | 0 | 27 | 421 | 0.9505 | 0.7823 | **1.0000** | 0.8778 | 0.8574 |
| | C-1C$_{50}$ | 182 | 1 | 34 | 428 | 0.9457 | 0.8426 | 0.9945 | 0.9123 | 0.8797 |
| | C-1C$_{75}$ | 297 | 1 | 35 | 413 | 0.9517 | 0.8946 | 0.9966 | 0.9429 | 0.9052 |
| | C-1C$_{100}$ | 409 | 2 | 31 | 404 | 0.9610 | 0.9295 | 0.9951 | 0.9612 | 0.9242 |
| Dataset 4 | C-2A$_{25}$ | 101 | 0 | 1 | 140 | **0.9959** | 0.9902 | 1.0000 | **0.9951** | **0.9916** |
| | C-2A$_{50}$ | 193 | 2 | 7 | 140 | 0.9737 | 0.9650 | 0.9897 | 0.9772 | 0.9465 |
| | C-2A$_{75}$ | 293 | 2 | 2 | 145 | 0.9910 | **0.9932** | 0.9932 | 0.9932 | 0.9796 |
| | C-2A$_{100}$ | 393 | 1 | 18 | 130 | 0.9649 | 0.9562 | 0.9975 | 0.9764 | 0.9115 |
| Dataset | C-2B$_{25}$ | 46 | 21 | 18 | 119 | 0.8088 | 0.7188 | 0.6866 | 0.7023 | 0.5619 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *5* | **C-2B$_{50}$** | 114 | 14 | 62 | 77 | 0.7154 | 0.6477 | 0.8906 | 0.7500 | 0.4686 |
| | **C-2B$_{75}$** | 150 | 50 | 50 | 81 | 0.6979 | 0.7500 | 0.7500 | 0.7500 | 0.3683 |
| | **C-2B$_{100}$** | 201 | 49 | 39 | 106 | 0.7772 | 0.8375 | 0.8040 | 0.8204 | 0.5282 |
| *Dataset 6* | **C-2C$_{25}$** | 87 | 0 | 4 | 137 | 0.9825 | 0.9560 | **1.0000** | 0.9775 | 0.9638 |
| | **C-2C$_{50}$** | 169 | 0 | 4 | 143 | 0.9873 | 0.9769 | **1.0000** | 0.9883 | 0.9748 |
| | **C-2C$_{75}$** | 249 | 0 | 3 | 152 | 0.9926 | 0.9881 | **1.0000** | 0.9940 | 0.9844 |
| | **C-2C$_{100}$** | 358 | 0 | 10 | 124 | 0.9797 | 0.9728 | **1.0000** | 0.9862 | 0.9488 |

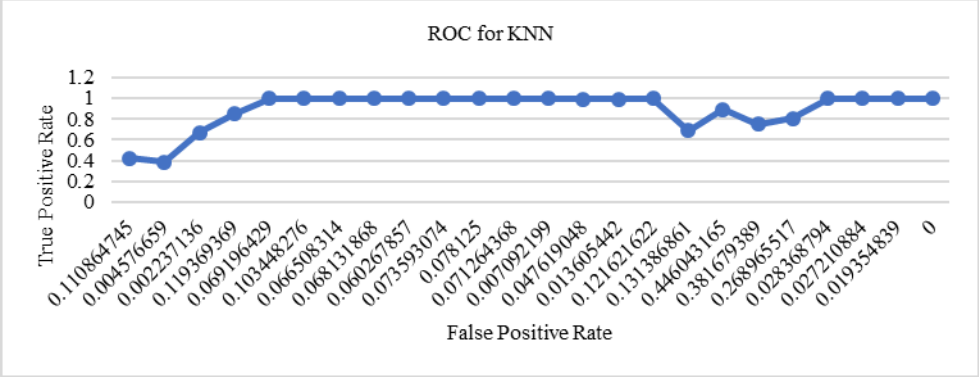


Fig. 4. ROC curve for KNN

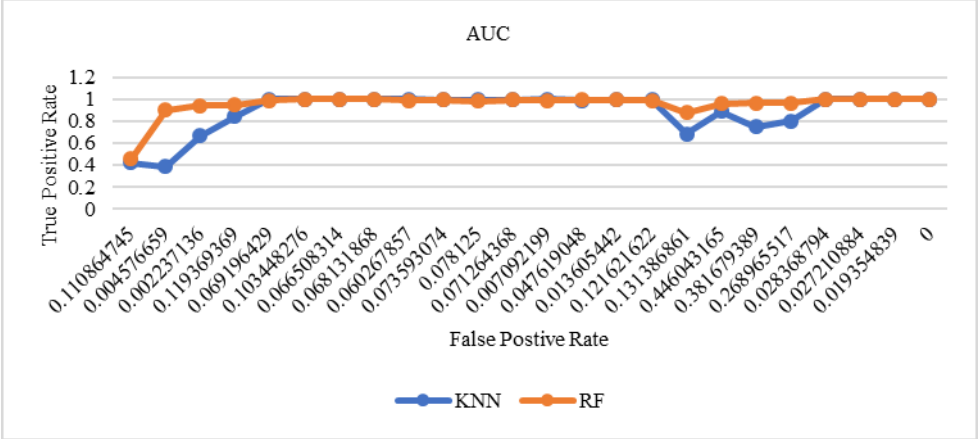


Fig. 5. AUC $_{curve}$ for RF and KNN

## V. CONCLUSION AND FUTURE WORK

In this study, we have implemented the data pre-processing and feature selection techniques on the datasets we got from the authors of one of the previous studies. After performing the FS with SVM-RFE and LR-RFE on biased data, we showed that in the case of Logistic Regression-RFE the outcome of FS varies in the case C-1A$_{25}$ where the number of fake accounts were 25% of the total number but it remains same for all other cases. 14 common features along with another significant feature highlighted by each of these techniques were selected making a set of total 16 features for further experimentation. We tested the predictive model with Random Forest and KNN algorithms for 6 different datasets having these 16 features with the varying degree of Sybil accounts to analyze the biasing effect. We concluded that the performance of Random Forest for our datasets based on selected set of features was better than KNN classifier.

We can further extend this study to implement our model on real-time data. Further an online application to identify the Sybil accounts can be developed to enhance the security of OSNs. Mitigation of biasing effects and testing the model against the spambot accounts will also be undertaken in future.

## REFERENCES

1. M. Al-Qurishi, M. Al-Rakhami, M. Alrubaian, A. Alarifi, S. M. M. Rahman, and A. Alamri (2015), "Selecting the best open source tools for collecting and visualizing social media content," In 2015 2nd world symposium on web applications and networking (WSWAN), IEEE, pp. 1-6.
2. A. A. Chinchore (2016), "Data mining of classification for sybil user detection" (Doctoral dissertation).
3. N. Bindra and M. Sood (2018), "Data pre-processing techniques for boosting performance in network traffic classification", First International Conference on Computational Intelligence and Data Analytics, ICCIDA-2018 26-27 October 2018, Springer CCIS Series, Gandhi Institute For Technology (GIFT), Bhubaneshwar, Odhisha, India.
4. J. Newsome, E. Shi, D. Song, and A. Perrig (2004), "The sybil attack in sensor networks: analysis & defences," Third international symposium on information processing in sensor networks, 2004, IPSN 2004, IEEE.
5. J. R. Douceur (2002), "The sybil attack," In First International Workshop on Peer-to-Peer Systems (IPTPS '02).
6. A. Vasudeva, and M. Sood (2018), "Survey on sybil attack defense mechanisms in wireless ad hoc networks," Journal of Network and Computer Applications.

7. A. Vasudeva and M. Sood (2016), "A vampire act of sybil attack on the highest node degree clustering in mobile ad hoc networks," Indian Journal of Science and Technology, vol 9.
8. M. Sood and A. Vasudeva (2013), "Perspectives of Sybil Attack in Routing Protocols of Mobile Ad Hoc Network," Computer Networks & Communications (NetCom). Springer, New York, NY, pp. 3-13.
9. A. Vasudeva and M. Sood (2012), "Sybil attack on lowest id clustering algorithm in the mobile ad hoc network," International Journal of Network Security & Its Applications.
10. M. Al-Qurishi, M. Al-Rakhami, M. Alrubaian, A. Alarifi, S. M. M. Rahman and M. S. Hossain (2017), "Sybil defense techniques in online social networks: A survey," IEEE Access, 5, pp. 1200-1219.
11. A. H. Wang (2010), "Don't follow me: Spam detection in twitter," In 2010 international conference on security and cryptography (SECRYPT),IEEE, pp. 1-10.
12. P. Galán-García, J. G. D. L. Puerta, C. L. Gómez, I. Santos, and P. G. Bringas (2016), "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, 24(1), pp. 42-53.
13. M. Alsaleh, A. Alarifi, A. M. Al-Salman, M. Alfayez, and A. Almuhaysin (2014), "Tsd: Detecting sybil accounts in twitter," In 2014 13th International Conference on Machine Learning and Applications, IEEE, pp. 463-469.
14. S. Cresci, R. D. Pietro, R. Petrocchi, A. Spognardi, and M. Tesconi (2015), "Fame for sale: Efficient detection of fake Twitter followers," Decision Support Systems, 80, pp. 56-71.
15. S. Cresci, R. D. Pietro, R. Petrocchi, A. Spognardi, and M. Tesconi (2017), "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," In Proceedings of the 26th International Conference on World Wide Web Companion International World Wide Web Conferences Steering Committee, pp. 963-972.
16. Project Jupyter. Available: https://jupyter.org/. Last Accessed on 29 April 2019.
17. K. Yan, and D. Zhang (2015), "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," Sensors and Actuators B: Chemical, 212, pp. 353-363.
18. H. Nkiama, S. Z. M. Said, and M. Saidu (2016), "A subset feature elimination mechanism for intrusion detection system," International Journal of Advanced Computer Science and Applications, 7(4), pp. 148-157.
19. I. Guyon, and A. Elisseeff (2003), "An introduction to variable and feature selection," Journal of Machine Learning Research, 3(Mar), pp. 1157-1182.
20. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi (2006), "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products," Chemometrics and Intelligent Laboratory Systems, 83(2), pp. 83-90.
21. S. B. Kotsiantis, I. Zaharakis, and P. Pintelas (2007), "Supervised machine learning: A review of classification techniques," Emerging artificial intelligence applications in computer engineering, 160, pp. 3-24.

## AUTHORS PROFILE

**Dheeraj Sonkhla** holds an engineering degree in Information Technology from University Institute of Information Technology, Himachal Pradesh University, Shimla, India. Currently he is pursuing M.Tech. in Computer Science from the Department of Computer Science, Himachal Pradesh University, Shimla, India.

**Dr. Manu Sood** is a Professor in the Department of Computer Science, HPU, India. He had held the additional charge of the Director, University Institute of Information Technology, Himachal Pradesh University, Shimla, India for four and a half years. He had also remained the Chairman of Department of Computer Science, Himachal Pradesh University, Shimla, India for two terms of two years each. He is an Engineering graduate, has an M.Tech. degree in Information Systems from University of Delhi (DU) with Gold Medal. He also holds the degree of Ph.D, from the Faculty of Technology, DU, Delhi, India. He possesses around 5 years of Industry experience and more than 25 years of academics and diverse administrative experience. His areas of interest in research are Software Engineering, e-Learning, security in WANETs and SDNs.