University of Fort Hare
*Together in Excellence*

**Department of Computer Science**

Private Bag X1314

Alice, 5700

# Modelling the detection of social media imposter using machine learning techniques

Yonela Nuba

201300992

201300992@ufh.ac.za

(+27)  660 127 381

Prof. K. Sibanda

Ksibanda@ufh.ac.za

29 November 2019

BSc Honours Degree

***Abstract***: People use social media for different intensions, some people use social media to advice or build other people socially, physically and emotionally, but, some also use social media to defame other people due to personal, business and/or political hatreds. In social media, a user create a social media account with their profile (location, age, contact details, pictures and etc.). In the present generation, people's social lives have become more associated with social media. The rapid increasing growth of use of social media and the large amount of personal data of its subscribers have attracted attackers and imposters to steal other people's personal data, spread false new and spread malicious activities as if they are from the rightful owner. In social media, the number of people commanded by an organization or an individual is a critical measure for their popularity and this can have a negative impact on the economic and or political implications. The measure of popularity of users is complicated by the fake profiles on the social media. In this study I will be creating a model that will identify the impersonated social media accounts. This can be performed on a number of social media networks but my focus is twitter because of its ease of access to data. On the other hand, researchers have been investigating efficient techniques that would be able to detect impersonated accounts and activities that are abnormal, but they are relying on the account features and classification algorithms. But then, some features of the account can have a negative impact on the final result and also using a stand-alone classification algorithm cannot lead to desired result at all times. In this paper I will be using three machine learning classification algorithms to decide the target account is real or fake, the algorithms would be, Support Vector Machine (SVM), Neural Network (NN) and a combination of the two algorithms which would be SVM-NN which correctly classified 97% of the accounts for our training dataset.

**Key words**: SVM, NN, SVM-NN, Social Media, Fake, Imposter, Machine learning, Classification, Real

## 1 Introduction

Social media is an online platform that allow people from different areas to interact or communicate to one another or as a group. Social media is used as a tool to create and to enhance friendship, advertising, data mining and for business networking.

Social media in our days has turned to be the main source of data mining. Almost every young and middle age person and every business do utilize the social media platforms either to socialize or for business purposes. The vast amount of data is being created mostly on social networks, millions of new social network accounts are created every year by both individuals and businesses/organizations. Social media platforms such as Twitter, Facebook, LinkedIn, Instagram and Google+, have gained popularity and have been increasingly becoming popular in the past few years.

Social media is a bridge to the gap of distant geographical area because it keeps people that are geographically distant connected and aware of what is happening anywhere. Through social media, people keep in touch with each other, they share news, organize events and run their own e-businesses. Between 2014 and 2018 close 2.53 million U.S dollars was spent sponsoring political ads on Facebook by non-profits in U.S [1]. Rohit Raturi [2] says that, as per a survey, there are 500 million female accounts on Facebook only, surprisingly, the population of females in the world is roughly 300 million. This tells us that, there are nearly 200 million female fake accounts on Facebook only, you can imagine the amount of unwanted data that is being created by this.

One vital tool in machine learning is text classification, this will teach us the security levels that we need to maintain in social media and in our daily routines [2]. In this generation of social networks, we have turned to use social media platforms like Twitter, Facebook Instagram and others to express our feelings, ideas and or opinions. There are a number of fake accounts that have been created on social networks, because of those, fake information is being circulated on the portals through improper channels. We need to come up with a strategy to minimize or avoid fake and harmful social media accounts to make sure that we save space in the data centers and as well to stop the controversy caused by these in the society on the social and political spheres.

One of the biggest problem with or in social media is that people turn to create fake profiles just to scam others as they can use the impersonated accounts for different targets within the social media. You will find that one of these targets will spread rumors which will affect a determined business [3] or even the society at large [4]. One example, in 2013, after the event of Boston Marathon bombing, an impersonated twitter account took the advantage of the social media community by twitting an announcement of a donation of 1 U.S dollar for each retweet [4].

The social media can be regarded as one of the most powerful tool for communication and very important to the society. Therefore in this research paper we aim to detect the social media imposters from twitter social network platform as a step towards the detection of fabricated news. The rest of the document is structured this way. Section 2 of the paper contains the previous work done related to the subject. Section 3 describes the twitter dataset, section 4, we demonstrate how the collected data has been used to classify the twitter accounts into fake and real accounts. In section 5 we discuss the overall accuracy rates and compare with other used methods. Then in section 6 we present our conclusion and future work.

## 2 Related work

Researchers have a task of coming up with different approach to detect fake accounts on social media. In this article we will be following the feature based detection approach [18]. In this approach we are based on monitoring the user behavior such as their number of tweets, friends following, etc. This is based on the fact that the behavior of human is totally different from that of fake. Therefore revealing this kind of behavior will lead to detection of fake social media accounts [19]. We will be showing what other researchers have presented in this section.

In [20], researchers correctly classified 84.5% of data to detect spammers by identifying 23 attributes. In this research, we have reached more accuracy with lower number of attributes, and that will be discussed in the next section. In [21], the authors have reduced the number of attributes by identifying only 10 attributes for detection. As mentioned, results won't be of satisfaction for identifying fake accounts with more positive perspective that it is able to identify fake tweets with higher accuracy by the support of graph techniques [9]. Although reference [22] presented the detection spammers with a minimized set of attributes that contained only six attributes, they mentioned that it could only detect certain types of spammers which are bagger and poster spammers [22]. In this paper we suggest an approach that will determine all types of fake accounts. To add on, one of the attribute requires text analysis procedure for finding similarities among the texts.

Researchers have started to investigate for an efficient way to detect fraudulent social media accounts mechanism. Most of these detection mechanisms classify and predict user accounts as real or fake by analyzing how active the user is or graph-level structures [10]. There are various number of data mining methods [11] and approaches that can yield to detecting fake accounts, they are explained in the following subsections.

### 2.1 Feature based Detection

This approach relies on user-level activities and its account details (user login’s and profiles). Unique features are extracted from recent user activities (e.g. frequency of friend requests, fraction of accepted requests), then those features are applied to a classifier that has been trained using machine learning techniques [5], [6], [7], [8]. In [6], authors have used a click-stream dataset which is provided by the social network that is used in China called RenRen [12] to cluster user accounts into groups according to similarities, corresponding to real or fake. Using the algorithm METIS clustering, with both session and clicks features such as:

- Average clicks per session
- Average session length
- Visit photos
- The percentage of clicks used to send friend requests
- Share contents

Authors were able to classify the dataset with 1% false negative rate and 3% false positive rate.
In [8], they used ground-truth also provided by RenRen to train a Support Vector Machine classifier to detect impersonated accounts, using fewer features such as:

- Fraction of accepted requests
- Frequency of friend requests

The authors managed to train the classifier with 0.7% false positive rate and 99% true positive rate. In [13], the authors also used a ground-truth but provided by twitter, the dataset was processed using two approaches:

- Feature sets proposed in the literature for detecting spammers
- Single classification rules

Features such as Stateofsearch.com rule set [14] and Social bakers rule set [15] have been used in the previous work and the authors were able to correctly classify the original training dataset with a percentage more than 95%.

## 2.2 Feature reduction

Memory usage and high computational costs could have a serious negative impact on high dimensional data for many classification algorithms. Thus, reducing dimensional space would lead to correctly classifying a model and simplifies the visualization technique [16] and would remove noisy and redundant features. Feature reduction can be split into two:

Dimension reduction: whereby, data in high dimensional space is turned to fewer dimensional space.

Feature subset selection: whereby only fewer features are extracted from the original features in order to build simpler and faster models, this also increases the model performance and gain better understanding of the dataset. Feature subset selection also has its own types (filtering methods and wrapping methods).

### 2.2.1 Principal Component Analysis (PCA)

This is the technique which is used to identify features within the dataset that best describes the predominant normal user behavior [8], [17]. PCA projects high dimensional data is transformed into a low dimensional subspace which is called normal subspace of the top-N principal components that accounts for as much inconsistency in the dataset as possible.

In [8], the researchers used the real dataset from three social media networks to show that normal user behavior is low-dimensional. The researchers used 14K Facebook users, 92K Yelp users and 100K Twitter users. The anomaly detection was evaluated using the ground-truth dataset of anomalous behavior which is revealed by compromised, fake and colliding users. The approach achieved a detection rate of over 66% covering more than 94% of misbehavior with less than 0.3% false positive rate [10].

### 2.3 Neural Network (NN) and Support Vector Machine (SVM)

In reference [23], extracted user profile data features using PCA, then applied NN and SVM to determine how legit the profiles are. A mathematical way of deriving PCA results which is Variance maximization was selected in order to produce such results as the following:

- Profile summary
- Number of languages
- Number of skills
- Number of connections
- Number of LinkedIn groups
- Number of publications

Researchers have used Neural Network with resilient back propagation and Support Vector Machine with C-support vector classification and polynomial kernel as kernel function. Our findings shows that, using PCA to select features can yield to better results than using all the dataset features without any selection. Feature based detection is regarded as the largest in social media but it is still relatively easy to avoid. Attackers would change their behavior to circumvent spam detection techniques by adversely changing the activity pattern and the contents of their fakes [5]. One can’t guarantee any formal security from feature based detection and it often leads to high false positive rate in practice [24].

## 3 Proposed Methodology

In this section, we aim to accurately classify the detection of impersonated accounts on twitter, with possible minimum set of attributes. Thus we will be using Twitter data mined from the public tweets for this study. The data was mined in the following manner:

The data is extracted from twitter using an API key and an access token to get the public tweets.

- A python library called tweepy was used to connect to tweet and collect tweets and profile data about the user that posted that tweet.
- API search was used to get certain information, only tweets that contained a certain word that were collected. The process was repeated several to get enough data.

Figure. 1 shows all the steps implemented to build a model that classified the twitter accounts.

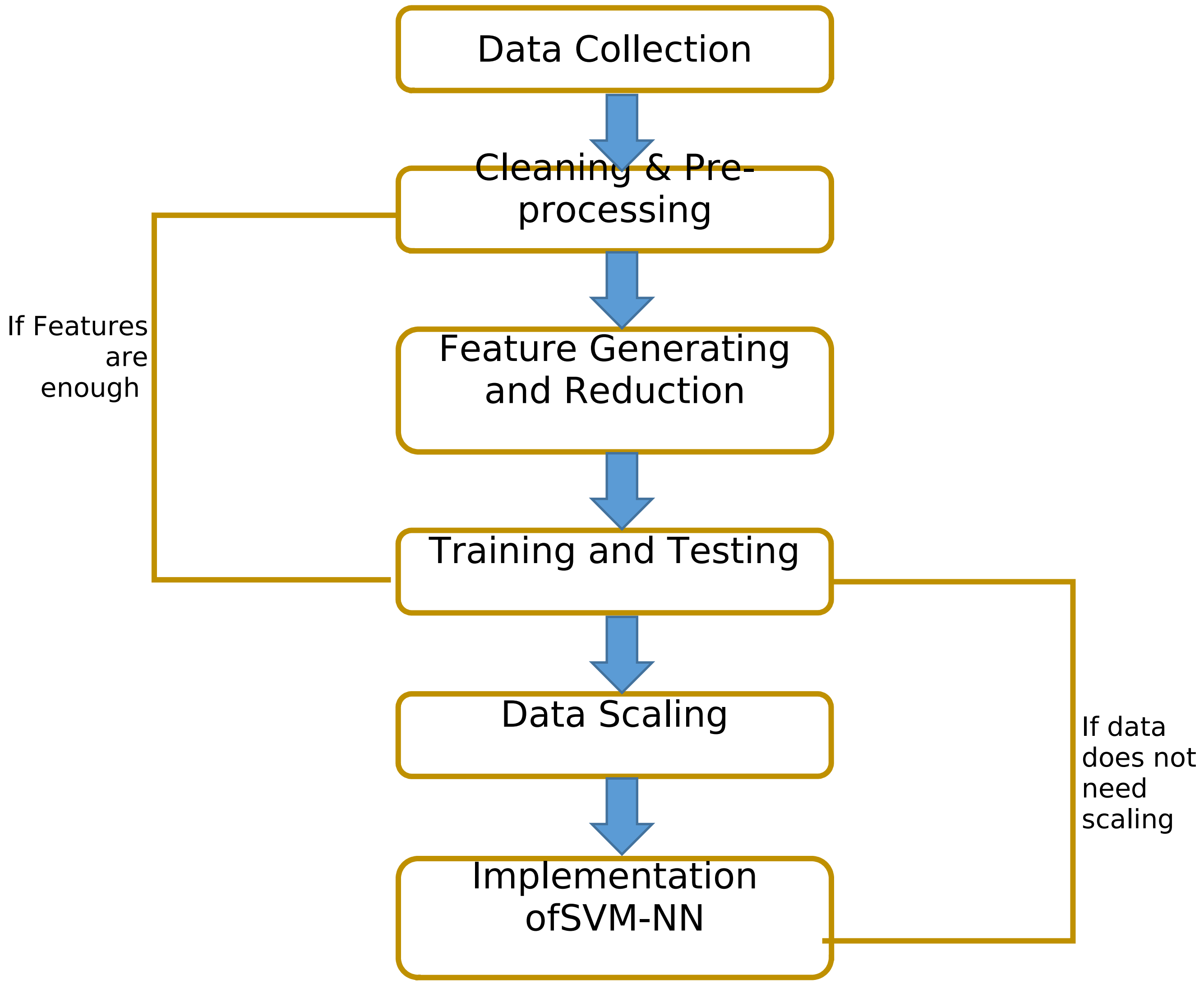
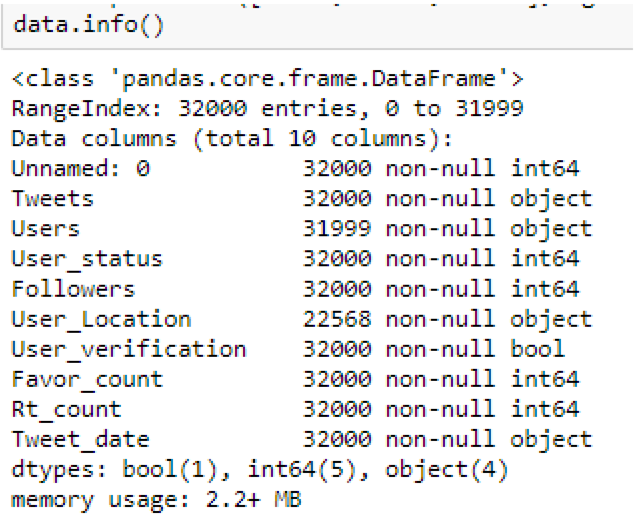


*Figure 1: showing all the steps to the implementation*

### 3.1 Data Collection

As we have mentioned that the data was extracted from twitter timelines according to the words that the tweets contained. API.Search method was very helpful on the function that collected the tweets. Since this process takes a very long time to implement, data was collected into several DataFrames that contained 1000 tweets each. As we called our function, the two parameters (‘word’ to search and the name of the file for that data frame) were passed and a CSV file was saved to the location of the working directory. The process was repeated with several ‘words’ that we used to extract the tweets. The data was then combined to formulate one consolidated CSV file named data. Then the data was manipulated as one pandas DataFrame. *Figure 2* shows some of the features that were extracted.

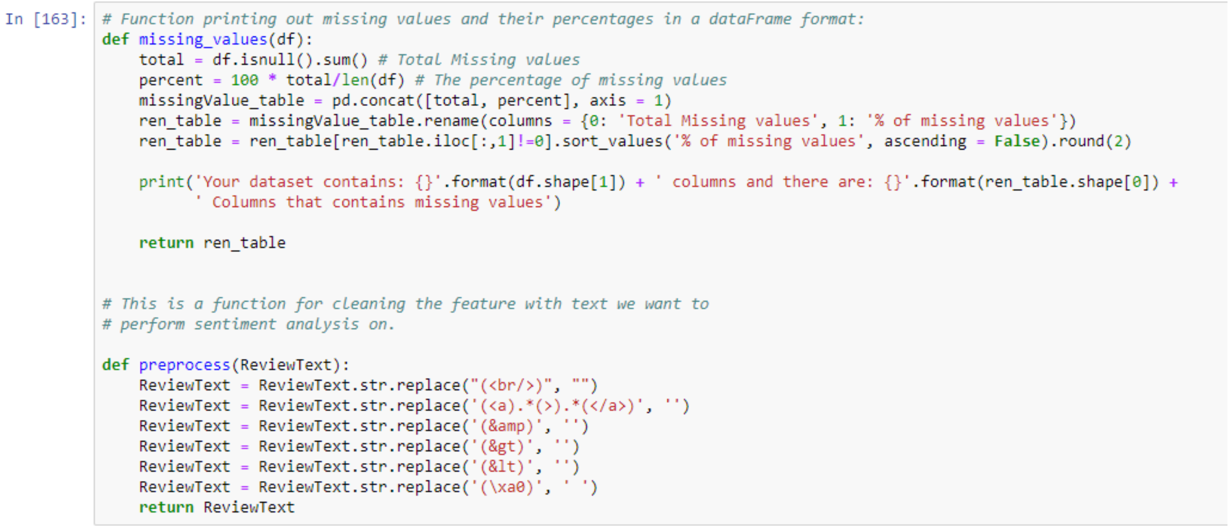


```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32000 entries, 0 to 31999
Data columns (total 10 columns):
Unnamed: 0           32000 non-null int64
Tweets               32000 non-null object
Users                31999 non-null object
User_status          32000 non-null int64
Followers            32000 non-null int64
User_Location        22568 non-null object
User_verification    32000 non-null bool
Favor_count          32000 non-null int64
Rt_count             32000 non-null int64
Tweet_date           32000 non-null object
dtypes: bool(1), int64(5), object(4)
memory usage: 2.2+ MB
```

*Figure 2 showing some of the features that were extracted*

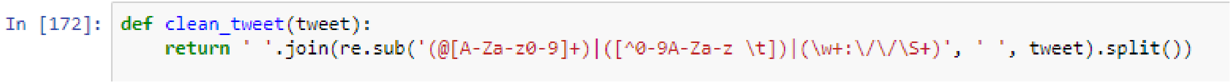
### 3.2 Data Cleaning and pre-processing

Cleaning data requires one to be fully aware of the data and all the features of the data and what they stand for or mean. Our data required lots of cleaning since some of the features had some missing data. This was caused by the fact that some people doesn’t like to put everything on their profiles while they are creating social media profile accounts. Sometimes we can have to fill the missing values (either with the most occurring values or mean or a median value depending on the importance of the feature) if the feature is more important to in the accuracy of the model when training.  After data has been extracted we removed all the features that were not important to our model. Leaving these features on the dataset would have the negative impact on our model during training and testing. Figure 3 and 4 shows some of the functions that we used to make during our data cleaning process.



```
In [163]: # Function printing out missing values and their percentages in a dataFrame format:
          def missing_values(df):
              total = df.isnull().sum() # Total Missing values
              percent = 100 * total/len(df) # The percentage of missing values
              missingValue_table = pd.concat([total, percent], axis = 1)
              ren_table = missingValue_table.rename(columns = {0: 'Total Missing values', 1: '% of missing values'})
              ren_table = ren_table[ren_table.iloc[:,1]!=0].sort_values('% of missing values', ascending = False).round(2)

              print('Your dataset contains: {}'.format(df.shape[1]) + ' columns and there are: {}'.format(ren_table.shape[0]) +
                    ' Columns that contains missing values')

              return ren_table


          # This is a function for cleaning the feature with text we want to
          # perform sentiment analysis on.

          def preprocess(ReviewText):
              ReviewText = ReviewText.str.replace("(<br/>)", "")
              ReviewText = ReviewText.str.replace('(<a).*(>).*(</a>)', '')
              ReviewText = ReviewText.str.replace('(&amp)', '')
              ReviewText = ReviewText.str.replace('(&gt)', '')
              ReviewText = ReviewText.str.replace('(&lt)', '')
              ReviewText = ReviewText.str.replace('(\xa0)', ' ')
              return ReviewText
```

*Figure 3: Shows functions for revealing missing values and cleaning object type features*

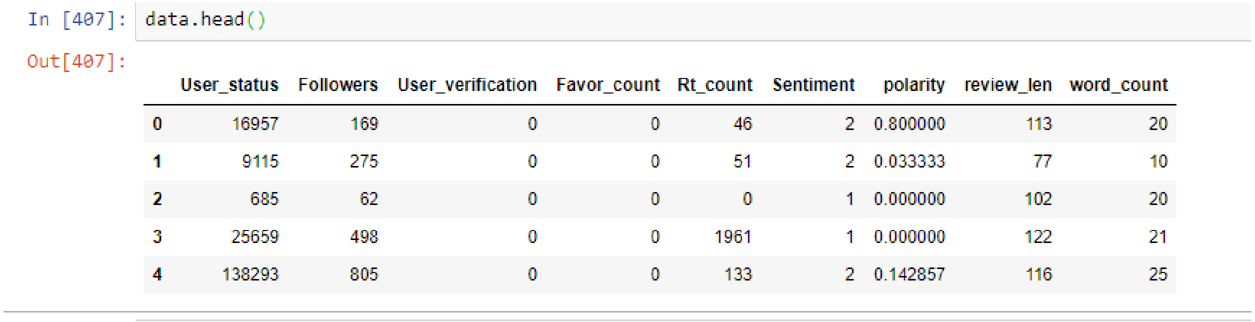


```
In [172]: def clean_tweet(tweet):
              return ' '.join(re.sub('(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)', ' ', tweet).split())
```

*Figure 4: shows a function for cleaning tweets using regular expression.*

### 3.3 Feature Generating and Reduction

This is one of the most important process that must be implemented before training and testing your dataset. This is a two way process, you can either generate more features or you can reduce features. These two can be implemented at the same time by generating more features and removing other features on the same dataset. In this article, we are performing both these operations. Generated features and other features remained after feature reduction must yield to correctly classifying the detection of social media imposters. If this process was not required, we would go straight to training and testing process. Figure 5 shows the datasets that were used on training the model.



```
In [407]: data.head()
Out[407]:
```

| | User_status | Followers | User_verification | Favor_count | Rt_count | Sentiment | polarity | review_len | word_count |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 16957 | 169 | 0 | 0 | 46 | 2 | 0.800000 | 113 | 20 |
| **1** | 9115 | 275 | 0 | 0 | 51 | 2 | 0.033333 | 77 | 10 |
| **2** | 685 | 62 | 0 | 0 | 0 | 1 | 0.000000 | 102 | 20 |
| **3** | 25659 | 498 | 0 | 0 | 1961 | 1 | 0.000000 | 122 | 21 |
| **4** | 138293 | 805 | 0 | 0 | 133 | 2 | 0.142857 | 116 | 25 |

*Figure 5: Features that were used to create a model.*

### 3.4 Training and Testing data

Since the data was just extracted without knowing what it looks like and what the results should look like or would be, we had to split the data to a training and testing datasets. 80% of the data was split to training data and 20% to testing data using the train_test_split python library from scikit-learn. The model was trained using the train dataset in order for the model to learn to classify the data accordingly and also to minimize the overfiting. Testing data was then used to check if the model really does classify the data accurately without noise and overfiting.

### 3.5 Data Scaling

Scikit-learn provides a better and easy way to standardize datasets. We used this library to standardize the train and test dataset so that they will have the same scale in order to avoid different impact of features to our model.

We have scaled the datasets so that we make sure that the classifier does not suffer because of the different scales on the dataset.

### 3.6 Implementing SVM-NN

As the potential for improving the accuracy, we have developed a new algorithm named SVM-NN, whereby we used the SVM trained decision values and trained a Neural Network model, and a Support Vector Machine testing values were used to test the Neural Network test model as shown in figure 6. This means that, a hybrid classification algorithm was developed and used on this research, by running the NN classification algorithm on the decision values coming from the SVM classification algorithm.

### 3.7 Implementation Tools

The research was implemented using only one programming language which is Python programming language. We have used a number of python libraries for scientific data manipulation for example scikit-learn library was used for most of the scientific operations, Tweepy library was used especially for data gathering. Python programing language was implemented on Jupyter Notebook (Anaconda environment), which is a web based python Integrated Development Environment (IDE). Scikit-learn is a commonly used library for data analysis and provides various tools for machine learning [25]. Scikit-learn provides various machine learning algorithms such as Regression, clustering, and classification, also provides different feature selection algorithms and evaluation methods/tools.

We also used Pandas python library which offers powerful data structures for data manipulation and data analysis such as a DataFrame. We used Pandas library to load the data from CSV and the library converted the CSV file to a

DataFrame which contains Rows and Columns just like a normal table. Pandas library also provides its own inbuilt functions that we used to manipulate the data until we reached the end of the implementation of our research. The study was carried out on an Acer EX 2519 Series laptop. Programming language used Python 3.6 and all the supporting software’s were run on Windows 10 64-bit operating System (OS)

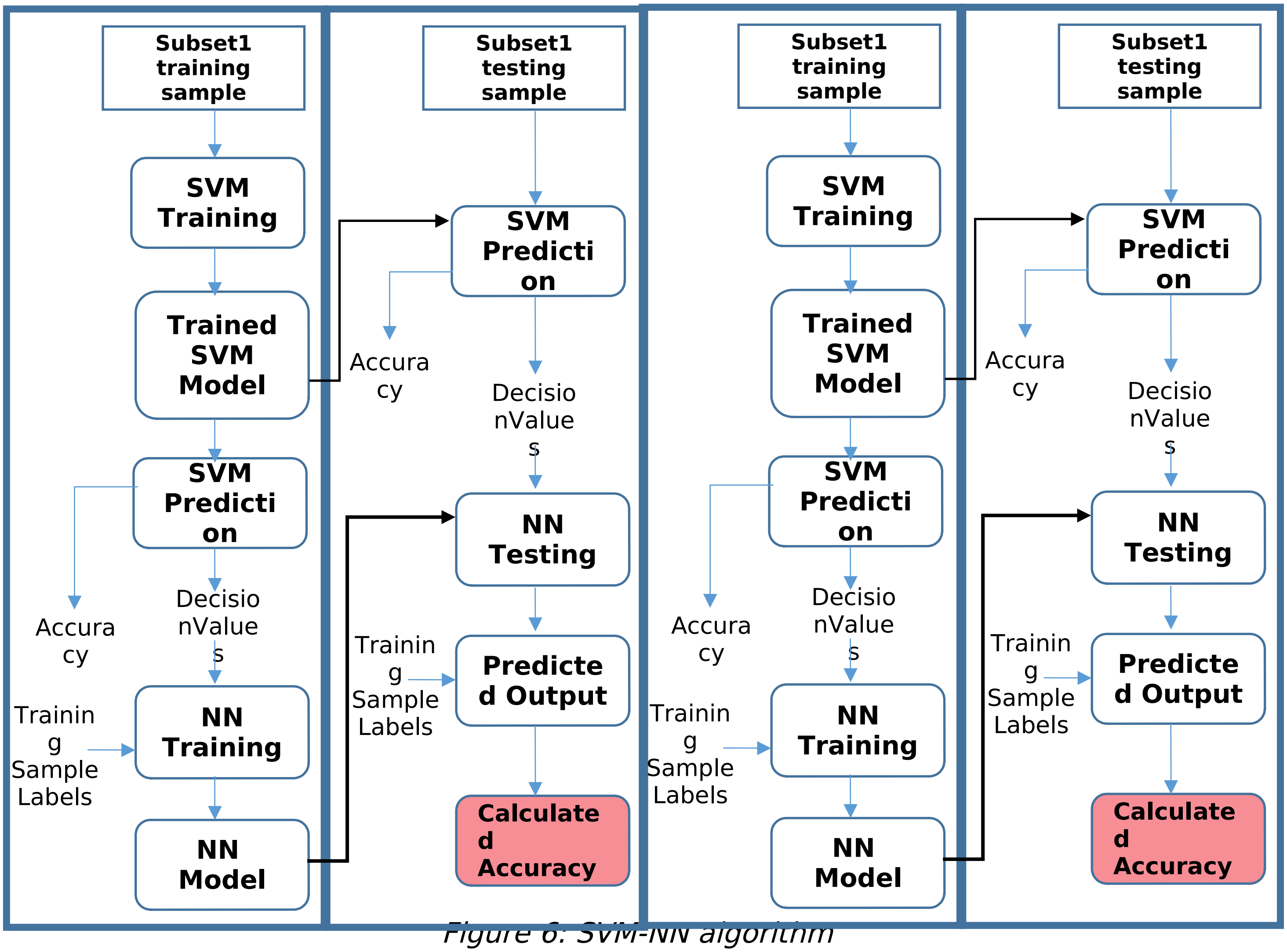


*Figure 6: SVM-NN algorithm*

## 4. Results and Evaluation

I this section the results and findings of this work would be explained and evaluated. There were three classification algorithms that have been trained and tested using all the remaining features after reduction and generating features process. Neural Network classification algorithm and Support Vector Machine classification algorithm were both used as the principles mining technique in many social media analysis researches. In this study, they have been applied on the dataset and compared with the proposed SVM-NN algorithm.

### A. SVM classification

As it is mentioned in related work, most of the researchers used SVM classification to differentiate between fake and real accounts. This is the reason why we also used Support Vector Machine in classifying our dataset and compared with Neural Network and SVM-NN. Radial Basis Function was exploited as Support Vector Machine classifier kernel, and trained using libSVM machine learning algorithm. The results of our algorithms are shown on Table 1.

| Feature Set | SVM | | | NN | | | SVM-NN | | |
|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | False-positive | False-negative | Accuracy | False-positive | False-negative | Accuracy | False-positive | False-negative |
| PCA | 0.886 | 0.111 | 0.001 | 0.775 | 0.195 | 0.203 | **0.925** | 0.086 | 0.001 |
| Correlation | 0.914 | 0.039 | 0.036 | 0.568 | 0.054 | 0.758 | **0.945** | 0.034 | 0.017 |
| Regression | 0.923 | 0.034 | 0.045 | 0.754 | 0.067 | 0.097 | **0.978** | 0.013 | 0.002 |
| Wrapper-SVM | 0.947 | 0.039 | 0.012 | 0.879 | 0.062 | 0.082 | **0.962** | 0.027 | 0.008 |

*Table 1: Accuracy results of applying SVM, NN, and SVM-NN on the features*

### B. Neural Networks

There are various number of Neural Network algorithms that that are used to train models and also predict the results, based on the previously trained models. For this study, we have chosen a feed-forward back propagation algorithm as the base algorithm. We compared if the predicted values with the actual values, whether the accounts are fake or real. We calculated prediction accuracy in the following manner.

$$\%Accuracy = \frac{All\, correctly\, identified\, accounts}{TotalTotal\, number\, of\, accouts} X\, 100$$

As shown in Table 1, the results shows that Support Vector Machine classifier has the highest accuracy when using the Wrapper-SVM feature set and the lowest accuracy when using the PCA feature set. The Neural Network accuracy is showing that it’s lower than the SVM classifier, with highest accuracy of 0.879 from Wrapper-SVM feature set and lowest accuracy using correlation feature set. Comparing the accuracy results of all the three classifiers, it is brightened that SVM-NN classification algorithm has the highest classification accuracy than all the other classifications in all the feature subsets as shown in Figure 8.

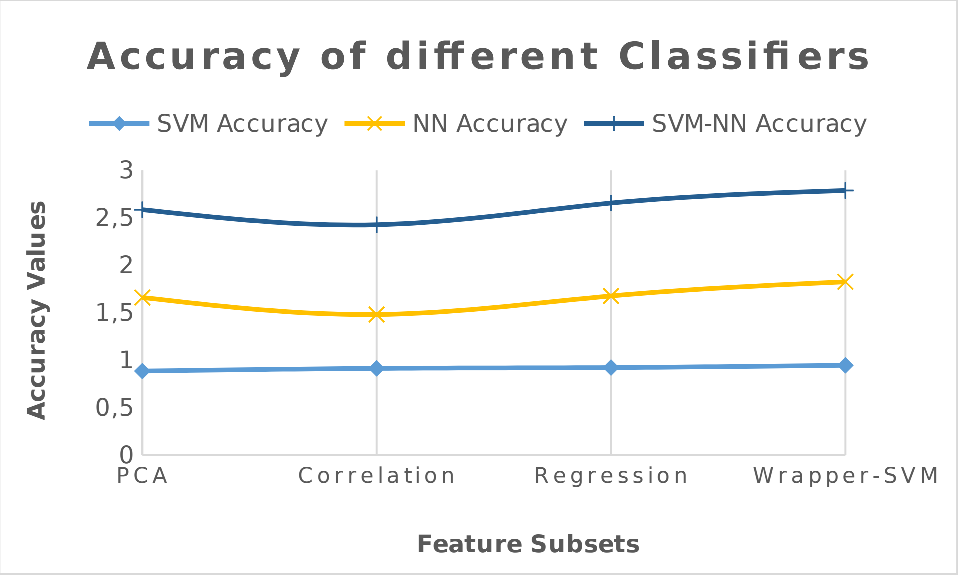


*Figure 8: SVM-NN classifier has the highest classification accuracy results on all the features as compared with other classifiers.*

## 5 Conclusion

The aim of the study was to detect social media imposters using machine learning techniques. The new classification method that we created has shown a great improvement in detecting impersonated accounts on social media. We used the Support Vector Machine trained model decision values to train the Neural Network model and also the Support Vector Machine testing decision values were used to test the Neural Network model. To reach our goal, we have converted some of our features from categorical to numerical so that we can work with. The results of our analysis showed that, our new algorithm SVM-NN has archived better accuracy score as compared with the other two classification algorithms. We can notice that Neural Network algorithm gave us a lower accuracy results as compared with Support Vector Machine and SVM-NN.

Using feature set provided by PCA resulted to lower classification accuracy results, while correlation feature set resulted to high classification accuracy. This occurred because PCA performs dimension reduction and creates new feature base on linear combination of original features but correlation and other feature selection methods only select the best set of original features, not linear combination of all features.

## References

**[1]** (2018) Political advertising spending on Facebook between 2014 and 2018. Internet draft. [Online]. Available: https://www.statista.com/statistics/891327/political-advertisingspending-facebook-by-sponsor-category/

**[2]** Rohit Raturi, “Machine Learning Implementation for Identifying Fake Accounts in Social Network” in Conference paper – August 2018

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

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

**[5]** Y. Boshmaf, D. Logothetis, G. Siganos, J. Ler´ıa, J. Lorenzo, M. Ripeanu, K. Beznosov, and H. Halawa, “´Integro: Leveraging victim prediction for robust fake account detection in large scale osns,” Computers & Security, vol. 61, pp. 142–168, 2016.

**[6]** G.Wang, T. Konolige, C.Wilson, X.Wang, H. Zheng, and B. Y. Zhao, “You are how you click: Clickstream analysis for sybil detection.” in USENIX Security Symposium, vol. 9, 2013, pp. 1–008.

**[7]** S. Fong, Y. Zhuang, and J. He, “Not every friend on a social network can be trusted: Classifying imposters using decision trees,” in Future Generation Communication Technology (FGCT), 2012 International Conference on. IEEE, 2012, pp. 58–63.

**[8]** B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, “Towards detecting anomalous user behavior in online social networks.” in USENIX Security Symposium, 2014, pp. 223–238.

**[9]** Ahmed El Azab, Amira M. Idrees, Mahmoud A. Mahmoud, Hesham Hefny, “Fake Account Detection in Twitter Based on Minimum Weighted Feature set”, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:10, No:1, 2016

**[10]** Sarah Khaled, Hoda M. O. Mokhtar, and Neamat El-Tazi, “Detecting Fake Accounts on Social Media” Conference Paper · December 2018.

**[11]** R. Kaur and S. Singh, “A survey of data mining and social network analysis based anomaly detection techniques,” Egyptian informatics journal, vol. 17, no. 2, pp. 199–216, 2016.

**[12]** A social network used in china. Internet draft. [Online]. Available: http://www.renren-inc.com/en/

**[13]** S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “Fame for sale: efficient detection of fake twitter followers,” Decision Support Systems, vol. 80, pp. 56–71, 2015.

**[14]** (2012) How to recognize twitter bots: 7 signals to look out for. Internet draft. [Online]. Available: http://www.stateofdigital.com/how-to-recognizetwitter- bots-6-signals-to-look-out-for/

**[15]** (last check 2018) Fake followers check: A new free tool from socialbakers. Internet draft. [Online]. Available: https://www.socialbakers.com/blog/1099-fake-followers-checka-new-free-tool-from-socialbakers?showMoreList-page=1

**[16]** A. S. M. Salih and A. Abraham, “Novel ensemble decision support and health care monitoring system,” Journal of Network and Innovative Computing, vol. 2, no. 2014, pp. 041–051, 2014.

**[17]** A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” in ACM SIGCOMM Computer Communication Review, vol. 34, no. 4. ACM, 2004, pp. 219–230.

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

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

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

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

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

**[23]** S. Adikari and K. Dutta, “Identifying fake profiles in linkedin.” in PACIS, 2014, p. 278.

**[24]** Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, “Aiding the detection of fake accounts in large scale social online services,” in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012, pp. 15–15.

**[25]** Anon, scikit-learn. Scikit-learn: machine learning in Python-scikit-learn 0.20.0 documentation. Available at: http://scikit-learn.org/stable/ [Accessed November 27, 2019].