International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Detection and Verification of Cloned Profiles in Online Social Networks Using MapReduce Based Clustering and Classification

Saravanan A.[1], Vineetha Venugopal*[2]

Submitted: 30/10/2022 **Revised:** 24/12/2022 **Accepted**: 03/01/2023

Abstract*:* The use of online social networks has become an unavoidable part of today's humans' life. The augmented usage of these online social networks has also increased the misusage through spreading spam messages, false reviews and fake news. Fake accounts are one such method in which attackers clone the profiles of innocent victims to harm them and damage their reputations. In some situations, fake accounts are also created to steal money from other users. In recent days, the detection of such cloned or fake profiles has perceived a wide range of attention from researchers. However, the existing methods lack accuracy and precision in detecting cloned profiles. In this paper, an effort has been made to identify the cloned profiles using clustering and classification process. The method collects the fake and possible cloned profiles and computes the additional relationship attributes for performing cloned profile detection. The parallel k-means clustering is carried out to group the suspicious profiles that are similar to the real ones. Then the parallel SVM classification is applied to the cluster in which the real profile is grouped. Finally, the classification results are verified using the attribute and network similarity measure. The various stages of the proposed model are implemented using the MapReduce framework which is more suitable for big data. The experimental analysis and results indicate that the proposed model has better performance than other competitors with an accuracy and precision of 98.19% and 98.96% for the MIB twitter dataset and 98.90% and 99.17% for the synthetic dataset created for the study.

Keywords: *Cloned profile detection, identity cloning attack, online social networks, parallel k-means, parallel SVM, mapreduce*

1. Introduction

In today's modern society, social media plays a vital role in everyone's life that even influences the culture, economy and social life [1]. The general purpose of social media is to keep in touch with friends, participate and share multimedia content such as text, images, videos and audio through an online application over Internet [2]. Due to its wide range of connectivity with people, even most organizations are officially encouraging their employees to use social media sites in improve their business. With advanced electronic devices such as smartphones and laptops, it is estimated that more than 50% of the world's population is using the Internet [3]. Among 4.66 internet users worldwide, around 4.14 billion people are active in social media applications such as Facebook, Twitter, Instagram and more [4]. Millions of users are active daily in such social media applications [5][6].

Though it has splendid advantages in sharing information, the widespread use of social media has become both a boon and a bane for society [7]. With online fraud, the spreading of false information or fake news is increasing at a rapid pace through online social media network (OSN). However, the most common illegal activities carried out on the internet and social media is the information and identity theft [8]. More specifically, the major source of false information on OSN is fake or cloned accounts, where scammers impersonate the victims through identity theft [9]. Apart from spreading fake information, OSN account profiles are cloned for deceptive advertisements, blackmailing, money laundering, terrorist propaganda, junk mailing, and misbehaviour. These activities are carried out with the intention of stealing information, harming the victim by damaging their reputation and reliability or ensuring th e trust that even influences the victim's friends and followers [10].

More commonly, the most significant ethical issue in online OSN sites is maintaining the real identity of the real user. However, social media sites undergo two significant problems as the unknown identity of the user and identity theft from the victim. The former is known as fake identity and the second is the cloned identity [11]. For fake identity, the adversary gathers the personal information of the victim from anyone OSN and creates a fake profile in a different OSN sites where the victim does not have an account in that OSN site. The phenomenon of creating a fake identity of the victim without his/her presence is known as fake profile attack (FPA). On the other hand, for creating cloned profiles, the personal OSN information of an existing victim's profile especially the popular person is gathered and the adversary creates one or more accounts by slightly modifying the profile data of the victim. These cloned accounts claim the identity of the victim in the OSN platform where the real account exists. This phenomenon of cloning the identity of the popular victim is known as

1. Sree Saraswathi Thyagaraja College, Pollachi, Tamil Nadu, India
ORCID ID : 0000-0002-8555-8796
2. Sree Saraswathi Thyagaraja College, Pollachi, Tamil Nadu, India
ORCID ID : 0000-0003-3362-4656
** Corresponding Author Email: vineetha72017@gmail.com*

identify cloning attack (ICA) [12]. The ICA is normally viewed as a special case of FPA since creating a fake profile in the same OSN where the victim has a real account is meant to be cloned profile [13].

The registration process in any OSN is very simple and easy to create an account to attract a large number of users. As a result, fake profile creation has also become easier which can be done within a few minutes. For any OSN site, the accounts created by the adversaries cannot be distinguished from real accounts and this makes the identity verification of real users more complex [14]. Thus, with the successful creation of profile cloning, the adversary tries to establish a connection with the victims' friends. Upon accepting the friend request sent by the adversary, the victim falls prey [15]. Recently, machine learning and statistics has become a predominant field in providing solutions to a wide range of problems including feature selection and classification [16], detection of attacks [17], spam detection [18] and so on.

Consecutively, there exists a wide range of research work on fake or cloned profile detection using machine learning and statistical approaches. However, most of the existing models are more complex due to the big data and still need to be improved in terms of detection accuracy. This indicates that there is a widespread need for a fake account detection model. Thus, an attempt has been made in this work to study the attributes of the profiles and provide a scalable and consistent model for identifying the cloned profiles. The profile attributes and the relationship attributes are extracted and for which the clustering and classification are applied to label the fake profiles. The model also verifies the result by applying attribute and network similarity measures. The method is implemented using the MapReduce framework which helps to tackle the problem of growing data rate in social networks to an extent.

The organization of the paper is as follows. Section 2 discusses the various existing models related to identifying fake or cloned profiles. The proposed models and the various phases of the proposed model are explained in section 3. Section 4 describes the experiments performed and the analysis of the results obtained along with the performance comparison with the notable existing detection models. Finally, the paper concludes with the conclusion section.

2. Related Works

Several authors have contributed their research on detecting fake or cloned profiles from social media networks like Facebook, Twitter, LinkedIn, etc. A few authors suggested simple statistical analysis by verifying the similarity between the profiles and verifying the behaviours and IP address to detect the cloned profiles [19]. In general, the various similarity metrics used for comparing the profiles are attribute similarity that assesses demographic information and network similarity that assess the friends' list [15] [20]. A similarity index parameter computation was proposed that makes use of weights for the attributes based on their importance in classifying the profiles [21]. A duplicated profile detection model on LinkedIn was proposed that employs profile feature similarities. This method applies a simple string-matching strategy for calculating the similarity value [22]. However, these model lacks the accuracy of results and they cannot be used independently in detecting fake profiles.

Many authors proposed their detection model by utilizing different attributes and insist that the inclusion of those attributes highly improves the performance of classifying fake profile from the real ones. These attributes are grouped under 5 categories as network-based, content-based, temporal-based, profile and action-based attributes [23]. Instead of utilizing text and categorical data, a detection model that analyzes the multimedia data was proposed in which the author reported that content-based and profile-based features offer higher precision in fake profile detection [24]. Moreover, several machine learning classification and clustering algorithms are evaluated in classifying fake profiles [25] [26]. A simple fake profile detection model based on a machine learning pipeline using random forest classifier was proposed [27]. Similarly, an agglomerative hierarchical clustering with Jaccard similarity metrics using weighted attributes was proposed to detect cloned profile detection [28]. Though the authors claim that the methods are effective, the work lacks intensive experimental analysis and comparison.

Since there is no proper dataset available for evaluating the fake profile detection models, each work extracts numerous attributes. However, not all of these methods are important for classification. Correlation is most important field of study in statistics to assess the dependency between the two set of values which is widely used in selecting the features [29]. Thus, a few feature selection techniques such as correlation and principal component analysis were applied before executing the classification algorithm [5]. A gain metrics-based feature weighting was evaluated in selecting significant features and the method was evaluated with various classifiers such as random forest, decision tree, naïve Bayes, neural network and support vector machines. It was found that among 22 collected attributes only 7 attributes were effective in detecting fake profiles [30].

A method that evaluates the relationship strength between two profiles with active friends lists and the number of likes to detect the cloned profile was proposed [31]. A smart system called FBChecker was proposed that makes use of behavioural and informational features with supervised learning algorithms to detect fake Facebook profiles. The method was executed by filling in the missing values using the KNN schema and by filtering the records having missing values [32]. However, the above works lack thorough experimental analysis to support the findings. The authors also extended their work with unsupervised clustering algorithms and the results indicate that ID3 offers improved detection accuracy [33]. Similarly, with 30 profile attributes, the analysis was carried out with advanced machine learning models such as boosting and bagging in which AdaBoost was proved to provide improved detection accuracy of fake detection [34].

Artificial intelligence along with natural language processing (NLP) was proposed to detect fake profiles. The model utilizes principal component analysis for feature selection was proposed. The model was evaluated using classifiers such as random forest (RF), support vector

machines (SVM) and optimized naïve Bayes algorithm among which the author insisted that the SVM classifier offers improved accuracy [35]. A novel idea of using the PageRank algorithm instead of using classifiers was proposed in which the model extracted the features and applied a clustering process to group similar attributes [36]. The PageRank algorithm was used finally to detect cloned profiles. Though the model was implemented with the MapReduce framework, the evaluation was made with the celebrity's profile. A privacy-protected system for detecting vulnerable and fake users and cloned profiles was proposed by employing a data mining algorithm [37]. The author reported that the model reduces the false rate to 1%. Among all these methods, only a few works support big data. Many of these models take more time to compare the profiles with similarity measures or classify the data using machine learning models.

With the knowledge gained from the literature study, it was found that each method lacks accuracy, and precision or suffered high computation complexity. This motivates this research to propose a model that detects the cloned profile using the MapReduce framework to ease the computation.

3. Proposed Framework of Cloned Profile Detection

The overall working procedure is classified into four main steps. The generic model starts by selecting and extracting profile data that can be clones of the victims' profiles. Here, the victim profile can be any profile for which we wish to know whether it has been cloned or not and identify the set of cloned profiles if it has. As a second step, the derived attributes such as friends count, followers count, like score and the group score are computed. Third, the model applies a clustering algorithm on the profiles based on their profile and derived attributes. Fourth, the classification algorithm is applied to the cluster containing the victim's profile for identifying the fake or cloned profile. Finally, the result of the classification is verified using similarity scores. The framework of the proposed system of detecting cloned profiles is presented in Fig. 1.

This proposed model has numerous advantages. First, it is an automatic process that works effectively to detect cloned profiles. Second, it applies a classification algorithm on a cluster having similar profiles instead of using the entire network of profiles. Thus, it highly reduces the classification time than other models. Moreover, with the clustering process, the profiles having similar user profile attributes falls under the same cluster which indicates the profiles are suspicious. And even it makes the detection process simple instead of using complex features. Third, it employs MapReduce programming and this aids in reducing the execution time of the entire model. Fourth, it applies attribute and network similarity metrics to verify the results produced by the classification algorithm to ensure the reliability of the results. The detailed procedure of each step in the proposed framework is discussed below in subsections.

3.1 Profile Selection and Extraction

The profiles that are similar or related to the victim's profile are selected for the study to identify the victim's cloned profiles. This can be selected in multiple ways. The proposed model employs three approaches to extract the users' profile data. Facebook Graph API is a simple way to extract profile data from the Facebook platform. Though it is considered to be the primary approach for accessing Facebook data, the main drawback is that it considers only active users. Thus, to extract all-possible profiles that can be clones of the victim's profile, the search query with the victim's name on the portal is made, to extract all the profiles having the same name as the victim. However, the method is also not reliable since only the best match is displayed and the results depend on the algorithm used. The algorithm may rule out the few other possible clones since the clones may not have the same name as the victim's name. For example, the victim's name can be the first name whereas the cloned profile name can have the last name.

Finally, another possible way to extract the user profiles is to fetch friends of friends who might have a high possibility to be the clones. More specifically the mutual friends are alone extracted since, in cloning attacks, the friend requests are probably sent to the victim's friends [31]. Generally, mutual friends are the friends who happen to exist both in the victim's list of friends and the friend list of the victim's friends. The reason for this way of extracting users' profiles for clone detection is that the fake users of the victim usually try to be the friends with the victim's friends to obtain the privacy data [20] as well as the cloned profile have at least few common friends with the victim [11]. The various profile attributes considered in the model are first name, last name, user name, location, age, gender, education, job, and email.

The data collected from these methods end up with duplicates. Thus, before proceeding further, the duplicate profiles are removed. This is carried out by comparing the extracted profiles based on a simple attribute-attribute match. Here, each attribute of real profile is compared with the same attribute of the extracted profiles. The numeric value 1 is assigned if the attribute values of the profiles are equal, else 0 is assigned. Finally, the values are summed to identify the duplicate data. The final summed value equal to the number of attributes extracted from the profiles indicates that the profiles are duplicates and thus the record of that specific profile can be removed.

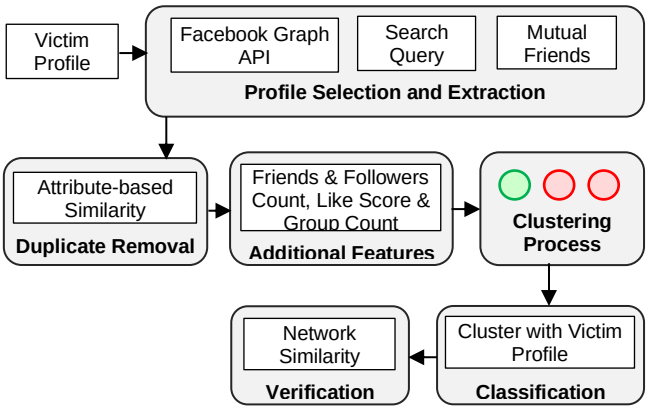


Fig. 1. A framework of the Proposed Cloned Profile Detection Model

3.2 Computation of Additional Features

In addition to users' profile data, the proposed method utilizes a few more features such as friends count, followers count, like score, post count and group count for effective identification of fake profiles. These attributes are used since it reflects the strength of the relationship between the user with other members of the OSN. This helps to identify fake or cloned profiles. The fake profile has fewer friends and followers than the real user. Moreover, as the fake profile users do not post or share the status consistently or frequently, the fewer posts and less like scores indicate the fake profiles apparently. Also, the real user joins more groups than the fake users. So, the fake profile or cloned profile has a lesser number of group counts than the real users. Thus, the additional attributes highly contribute to the detection of cloned or fake profiles.

3.2.1 Friends and Followers Count

In this step, the friends count and the followers count of each profile selected through the previous step are sequentially computed. The obtained counts are considered additional features and are added along with the other profile feature for each user.

For computing friends count, the friend list dataset is considered. The records are distributed and are assumed in key-value pairs with the User ID as key and the friend ID as value. The Mapper class reads each input record and produces <User_ID, 1> as output in <key, value> format. The output of the Mapper class is shuffled, sorted and grouped based on the keys and results in <key, list(values)>. Next, the keys are given as input for the Reducer class which counts the number of 1's (value) corresponding to the given input key or user ID. For example, with <A, B>, <A, C> and <B, C> as input, the Mapper produces <A,1><A,1><B,1> as output. After shuffling, the pairs become <A, [1,1]> <B, [1]> and are fed into the Reducer class. The Reducer counts the number of 1's in each key producing 2 and 1 as the final output indicating A has 2 friends and B has 1 friend.

For computing the followers count, the model utilizes the follower list dataset in which each record is represented in key-value pair indicating the user ID and the follower ID. The same procedure used in friends count computation is applied to computing the followers count [36] [38]. The pseudocode for computing the friends and followers count is presented in Algorithm 1 and Algorithm 2.

Algorithm 1. Friend Count Computation

```
Function MAPPER(User_ID, Friend_ID)
    Process input < User_ID, Friend_ID) >
    Emit < User_ID, 1 >
End Function
Function REDUCER(User_ID, val_list))
    Set frd_count = 0
    For each item i in val_list do
        val = val_list[i]
        frd_count = frd_count + val
    End For
    Write user_ID and frd_count in HDFS
End Function
```

Algorithm 2. Follower Count Computation

```
Function MAPPER(User_ID, Follower_ID)
    Process input < User_ID, Follower_ID) >
    Emit < User_ID, 1 >
End Function
Function REDUCER(User_ID, val_list))
    Set folw_count = 0
    For each item i in val_list do
        val = val_list[i]
        folw_count = folw_count + val
    End For
    Write user_ID and folw_count in HDFS
End Function
```

3.2.2 Like Score and Post Count

Like score is a significant feature in social media like Facebook. It signifies the average number of likes obtained by the user from different posts through his/her friends and followers. The formula to compute the like score for the given UserID is given in (1).

$$\text{LikeScore(UserID)} = \frac{\sum_{i=1}^{k} n(likes)}{k} \quad (1)$$

Here k is the number of posts and n(likes) represents the number of likes received for each post i.

For computing like scores through MapReduce, the like list dataset is considered. The dataset records are distributed which are assumed in key-value pairs indicating the likes given by the followers or friends F_ID for the post Post_ID as <User_ID, (Post_ID, F_ID)> [38]. The Mapper class reads each input record and produces <User_ID, Post_ID> as output in <key, value> format. The output of the Mapper class is shuffled, sorted and grouped based on the keys and results in <key, list(values)>. The list(values) represents the list of Post_ID. Next, the keys are given as input for the Reducer class which computes the average number of likes obtained by the User_ID from the other followers. For example, consider <A, (P_1, B)>, <A, (P_1, C)> <A, (P_2, B)> <A, (P_2, D)> as input. It indicates that for the first post P_1 posted by A, B and C have given likes, and in the second post P_2 posted by A, B and D have given likes. Thus, the Mapper produces <A, P_1><A, P_1><A, P_2><A, P_2> as output. After shuffling, the pairs become <A, [P_1, P_1, P_2, P_2]> and are fed into the Reducer class. The Reducer counts the number of posts as 2 and the average likes as 2 (=4/2). The pseudocode for computing the like score is presented in Algorithm 3.

Algorithm 3. Like Score Count

```
Function MAPPER(User_ID, [Post_ID, F_ID])
    Process the input < User_ID, (Post_ID, F_ID) >
    Emit < User_ID, Post_ID >
End Function
Function REDUCER(User_ID, Post_ID_list)
    Set like_count = 0, post_id_count = 0
    Unique_Post_ID = new ArrayList < string >
    For each Post_ID i in Post_ID_list do
        If (! Unique_Post_ID contains Post_ID) then
            post_id_count + +
            Unique_Post_ID.add(Post_ID)
        End If
        like_count + +
    End For
    like_score = like_count/post_id_count
    Write like_score and post_id_count in HDFS
End Function
```

3.2.3 Group Count

The group is another important metric of social media. It signifies the number of friends groups joined by the user. For computing group count through MapReduce, the group list dataset is considered. The dataset records are distributed which are assumed in key-value pair indicating the <User_ID, G_ID> where G_ID is the different group id. The Mapper class reads each input record and produces <User_ID, 1> as output. The output of the Mapper class is shuffled, sorted and grouped based on the keys and results in <key, list(values)>. Next, the keys are given as input for the Reducer class which computes the number of groups in which the user has joined. For example, with <A, G_1>, <A, G_2)> <B, G_2)> as input, it is understood that A has joined in G_1 and G_2 and B has joined in G_2. Thus, the Mapper produces <A,1><A,1><B,1> as output. After shuffling, the pairs become <A, [1,1]> <B, [1]> and are fed into the Reducer class. The Reducer counts the number of 1's in each key producing 2 and 1 as the final output indicating A has joined in 2 groups and B has joined in 1 group. The pseudocode for computing the group count is presented in Algorithm 4.

Algorithm 4. Group Count Computation

```
Function MAPPER(User_ID, Group_ID)
    Process input < User_ID, Group_ID) >
    Emit < User_ID, 1 >
End Function
Function REDUCER(User_ID, val_list))
    Set group_count = 0
    For each item i in val_list do
        val = val_list[i]
        group_count = group_count + val
    End For
    Write user_ID and group_count in HDFS
End Function
```

3.3 Clustering Process

Once the profile data and the additional attributes are extracted, the next step is to apply a clustering algorithm. In the proposed work, parallel k-means clustering, a simple yet powerful technique has been utilized using the MapReduce programming model [39]. This model is considered to be effective for large-scale data. In general, the k-means algorithm assigns k records as the initial cluster centre. For the remaining records, the distance between the records and the cluster centres are computed. The records are then assigned to the corresponding clusters having a minimum distance from the centres [40]. The procedure is iterated until the stopping criteria are met. Notably, the computation of the distance between a record with the cluster centres is independent of the distance computation between other records and cluster centres. Thus, in parallel k-means, the distance computations of records are performed parallel and the iterations are performed successively. The pseudocode for the parallel k-means clustering is presented in Algorithm 5.

Algorithm 5: Parallel K-Means Clustering

```
Function MAPPER(key, value)
    Extract the profile p from value and a
ssign maximum value to min_dis
    index = −1
    For each i in k cluster centres do
        Compute distance (p, centres [i])
        If distance < min_dis
            minDis = distance and index = i
    End For
    key′ is the index, and value′ is the profile
    Emit < key′, value′ >
End Function
Function COMBINER(key, V)
    Create an array attr_sum and initialize it to 0
    num = 0
    For each profile in V and each attribute in the pro
            attr_sum = attr_sum + profile[i]
            num + +
    End For
    key′ is key; value′ is attr_sum and num
    Emit < key′, value′ >
End Function
Function REDUCER(key, V)
    Create an array Attr_sum and initialize it to 0
    Num = 0
    For each profile i in V and for each attribute j in p
            Attr_sum[j] = Attr_sum[j] + profile[i, j]
            Num + +
    End For
    New cluster centre is Attr_sum/Num
    key′ is key and value′ is new values of centres
    Emit < key′, value′ >
End Function
```

In the above algorithm, the records in the dataset are depicted in the format <key, value> and are stored on HDFS. In each instance of the formatted dataset, the *value*

represents the individual profile records in the form of a string and the *key* is the offset of the profile indicating the starting point of the dataset file. The datasets are split and distributed for parallel distance computation. Each Mapper function takes the information of cluster centres, key (offset) and value (instance string) as input. It then identifies the closest centre for the instance and returns the closest centre and instance as intermediate results. The intermediate results are fed to a Combiner which combines the results of the same Mapper. The Combiner function takes the cluster index (key) and list of instances assigned to the same cluster (V) as input. Then computes the partial sum for each attribute of the instances belonging to the same cluster. Thus, the index (key′) and the sum of attributes of the cluster along with the number of instances (value′) are generated as output. The Reducer processes the output received from the Combiner of each host. The Reducer function adds all the attributes of the instances belonging to each cluster and computes the mean attribute value which serves as the new cluster centre for the next iteration. The detailed procedure was explained by Zhao et al. (2009).

3.4 Classification

The next step applies a classification algorithm for the cluster in which the victim profile belongs to. This highly reduces the computational complexity of the model. Also, the proposed method utilizes a Support Vector Machine (SVM) to classify the cloned profiles from the real ones. The reason behind using SVM is that the SVM classifier offers better performance with decreased training time specifically when the number of instances is less. Moreover, it is proved in the literature that it offers increased accuracy, and precision over the random forest, naïve Bayes, decision tree, and KNN classifiers [32] [35].
Owing to the merits of the SVM classifier, the proposed model utilizes distributed parallel SVM algorithm using the MapReduce framework [41]. The algorithm offers an effective solution for binary problems. However, before applying the classification algorithm, the training dataset is prepared containing a set of profiles with all the profile attributes, additional attributes and class labels indicating the profile is fake or real. To train the model, the subset of the training dataset is distributed over the nodes and the global support vectors (SVs) are initialized to 0. Then the global support vectors are merged with the training subsets which are then trained using SVM. The nodes identify the new support vectors and are then finally merged and the results are stored in global support vectors. This process continues until there is no change in the hypothesis. Here the hypothesis is to have minimal empirical risk. The algorithm pseudocode for the distributed SVM is presented in Algorithm 6.

Algorithm 6. Parallel SVM Classifier

$\boldsymbol{Function\ MAPPER(key, value)}$
  $Initialize\ Global\ Support\ Vector\ GSV = \phi$
    $while\ hypo(t) \neq hypo(t-1)$ //
      t represents the iteration
    $For\ each\ node\ l$
      $\in L$ // L *is the number of nodes*
    // merge subset TD with GSV at iteration t
    $Merge\ TD_l$ (t) with GSV(t)
  $End\ For$
  $End\ While$
  $Emit < key', value' >$
$\boldsymbol{End\ Function}$
$\boldsymbol{Function\ REDUCER(key, V)}$
    $while\ hypo(t) \neq hypo(t-1)$ //
      t represents the iteration
    $For\ each\ node\ l$
      $\in L$ // L *is the number of nodes*
    $Train\ the\ model\ with\ merged\ data$
    $Compute\ the\ support\ vectors\ and\ hypothesis$
    SV_l, hypo(t) = BinarySVM(TD_l)
  $End\ For$
  $For\ each\ node\ l$
      $\in L$ // L *is the number of nodes*
    $Merge\ SV_l$ with GSV /
      / merge local SVs with GSV
  $End\ For$
$\boldsymbol{End\ Function}$

Thus, in the above algorithm, the Mapper combines the training subsets with global support vectors and the Reducer trains the model and evaluates the local support vectors which are then combined with the global SVs. Here, the SVM hyperparameters such as C and γ are evaluated using 10-fold cross-validation.

3.5 Attribute Network Similarity based Verification

Once the classification accuracy is computed, the next step performs the verification of classified results by examining the similarity between the real profiles and the possible clones as well as their mutual friends [20]. Each profile attributes of the real profile are compared with the profile attributes of the possible clones using different methods and the formula is presented in (2).

$$Profile\_sim(P_r, P_v) = \frac{\sum_{i=1}^{k} d_i(P_r, P_c)}{n} \quad (2)$$

Here P_r, P_v represents real and cloned profiles, n represents the number of attributes, $and\ d_i$ is the similarity function used for the i[th] attribute.
For simple attributes like location and gender, the algorithm just verifies whether the values are equal. If they are equal, it returns 1 or else, it returns 0. Similarly, for computing the similarities of the attributes such as education and job, the id of the retrieved fields is compared. Alternatively, for computing the similarity for

the numerical values like age, the ratio of the difference between the ages to the total of the compared age is computed. The formula is given in (3)

$$d_{age}(P_r, P_c) = 1 - \frac{|P_r(age) - P_c(age)|}{P_r(age) + P_c(age)} \quad (3)$$

To compare the names and email IDs, the model utilizes the dice similarity coefficient which compares the unigrams of the strings instead of comparing the name strings [42]. The formula to compute the dice coefficient is presented in (4).

$$d_{string}(P_r, P_c) = \frac{2 \times |P_r(ugram) \cap P_c(ugram)|}{|P_r(ugram)| + |P_r(ugram)|} \quad (4)$$

Thus, to compare the strings such as first name, last name, and user name, the strings are split into a set of unigrams. The set intersection between the unigrams set is performed as in (4) for computing the similarity between the strings.
Additionally, the friend similarity is also computed using a mutual friend list as in (5).

$$frnd\_sim(P_r, P_c) = \frac{|MF(P_r, P_c)|}{|F_r|} \quad (5)$$

Here $|MF(P_r, P_c)|$ represents the number of mutual friends of the real and possible cloned profile, and F_r is the number of friends of the real profile.
Moreover, the scores obtained from the relationship attributes such as friends count, followers count, like score, post count and group count are utilized to verify the results obtained from the classification model. The average of the relationship score is computed for cloned profile verification.
Thus, if the similarity values of $Profile\_sim(P_r, P_v)$ and $frnd\_sim(P_r, P_c)$ are greater than 0.5 and the average relationship score of the possible cloned profile is less than that of the real profile, then the possible cloned profile is actually cloned profile. The threshold value of 0.5 is taken since it is proved in literature as an optimal value [20]. The overall workflow of the proposed model is presented in Fig. 2.

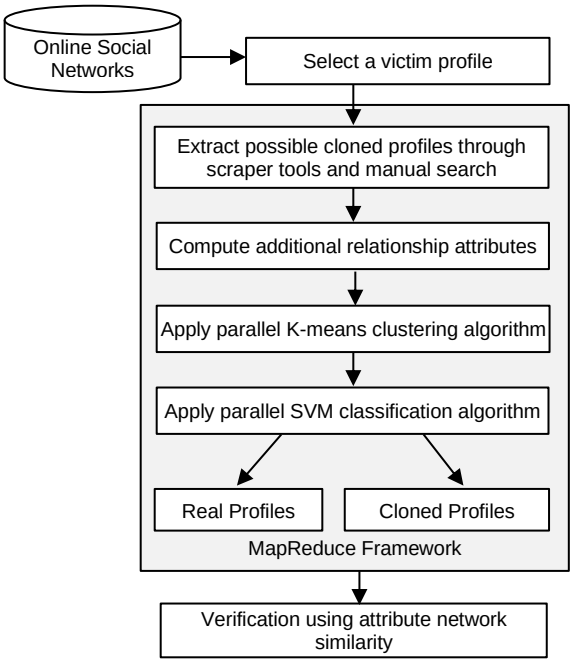


Fig. 2. Workflow of the Proposed Cloned Profile Detection Model

4. Result and Discussion

To evaluate the performance of the proposed cloned profile detection model, a dataset has been created through various forms such as Facebook Graph API [43], manual search and extracting the details of mutual friends. The dataset used for the study contains 1000 records with 400 real samples and 600 cloned or fake profiles which are labelled manually [34]. The fake or cloned profiles are created manually by changing the parameters for evaluating the fake profile detection algorithms. These are considered as the training set to train the classification model used in the proposed model. The test data is also extracted which contains 10 real profile data with 10 fake profiles for each real profile created manually with a total of 100 fake profiles. Once the additional attributes are computed for the collected profile, the profile data are converted to numerical values to facilitate the clustering process [36].
The experimental analysis used in the performed is multi-fold. The performance of each stage such as the performance of attributes, clustering algorithm and classification algorithm are evaluated individually. Finally, the efficiency of the proposed algorithm is also assessed by comparing the results of the proposed model with that of the other existing models. These experiments are conducted with a System with Intel Pentium 6405U Processor with 2.40 GHz, 8 GB RAM, Windows 10 operating system. For running the MapReduce framework Hadoop 2.5.2 is used.
To evaluate the effectiveness of the attributes used in the study, an analysis is carried out by varying the attributes. This includes a dataset with profile data only, profile data with friends count, profile data with followers count, profile data with like & post score, profile data with group score, profile data with friends and followers count, profile data with like and group score and all attributes including profile data, friends, followers count and like, group score. Here the analysis is simply performed with different

classification algorithms employed generally for fake profile detection. The classifiers used for the analysis include support vector machine (SVM), Random Forest (RF), Naïve Bayes (NB) and k Nearest Neighbour (KNN). The obtained accuracy for the training dataset with 10fold cross-validation is presented as a graph in Fig. 3.

From the results, it is found that including at least one relationship attribute to the profile data increases the accuracy from a minimum of 14% to a maximum of 19% for SVM, 14% to 22% for RF, 7% to 14% in NB and 14% to 21% in KNN. On the other hand, including the two additional attributes such as friend and followers count increases the accuracy by 26%, 23%, 24% and 30% for SVM, RF, NB and KNN respectively. Similarly, including the like & post score and group score also increases the accuracy of the SVM, RF, NB and KNN classifiers by 28%, 27%, 29% and 36% respectively. Thus, the overall increase in the accuracies using all the additional attributes with profile data is 37%, 37%, 40% and 45% for SVM, RF, NB and KNN respectively. The overall accuracy of the SVM, RF, NB and KNN classifiers with all the attributes are 98.5%, 97.2%, 96.3% and 94.5% respectively. The result indicates that 1) the profile data with additional relationship attributes such as friend and followers count and like and group score extremely increases the accuracy of the classifier and 2) the SVM classifier outperforms other classifiers such as RF, NB, and KNN.

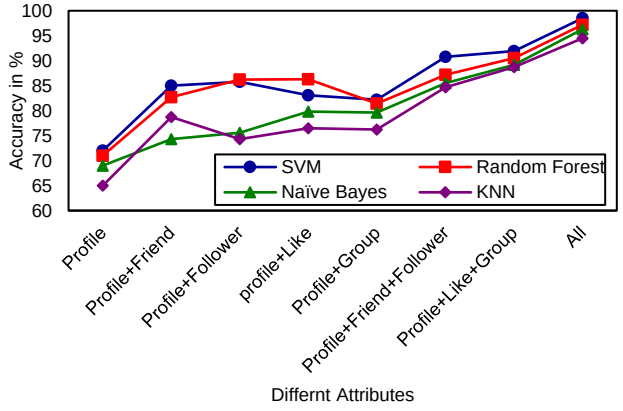
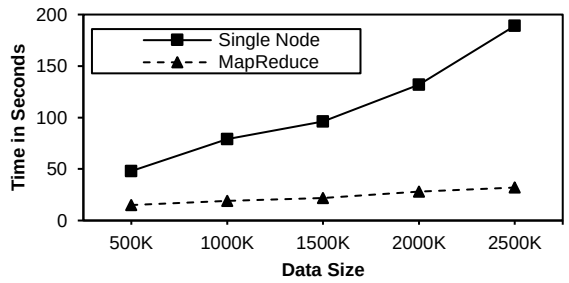
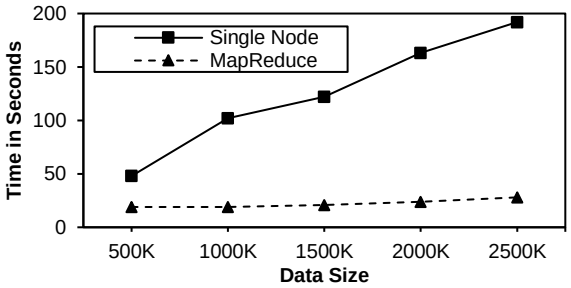


Fig. 3. Performance Analysis with Varied Attributes

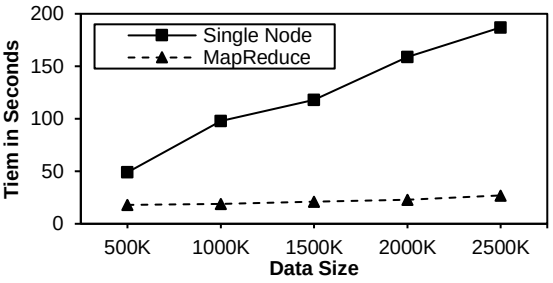
In general, collecting the additional relationship attributes of the profile is a time-consuming task with a single node. However, with the use of the MapReduce framework, this task seems to be simple and flexible. Thus, the effectiveness of using MapReduce in extracting the additional attributes is evaluated. The results of the analysis on the execution time of computing friends count, followers count, like & post score and group score are presented in Fig. 4 (a-d). From the results, the execution time of calculating various additional attributes is very less by using the MapReduce framework than by a single node. Moreover, the execution time is also consistent even with the increase in the number of data which is not in the case for the single node. Thus, the MapReduce framework used for calculating relationship attributes is more suitable for big data.



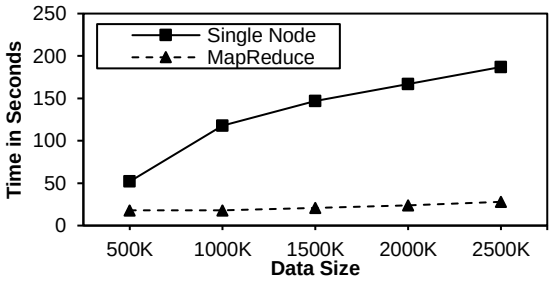
a) Friends Count



b) Friends Count



c) Like Score



d) Group Score

Fig. 4. Performance of Computing Additional Relationship Attributes

Once the attributes are extracted for the profile data, the next step is to apply a clustering algorithm to group the more similar profiles with that of the real profile. The parallel k-means clustering used in the model is analyzed by varying the number of clusters (k value) from 2 to 5 and the results are compared with c-means clustering and k-medoids clustering algorithm. The analysis is carried out for the test records having 10 real users and by varying the number of cloned profiles (c) as 1, 2, 5, 10, 20, 50 and 100 [36]. The obtained results are presented in Table 1. Here each accuracy value is the average of 10 iterations performed with the respective model.

Table 1. Accuracy of Different Clustering Algorithms

K value	Clustering Algorithms	Accuracy by varying fake profile count						
		c=1	c=2	c=5	c=10	c=20	c=50	c=100
2	c-means	100.0	93.4	87.2	84.0	83.9	84.3	85.6
	k-medoids	97.8	95.1	93.5	82.6	81.7	92.1	89.9
	parallel k-means	99.2	95.8	94.2	87.7	82.7	93.2	90.8
3	c-means	89.6	88.3	85.4	80.5	81.2	82.6	83.9
	k-medoids	95.2	93.8	92.0	87.9	85.6	92.4	87.8
	parallel k-means	98.3	95.7	93.5	88.4	84.3	91.5	89.8
4	c-means	85.5	84.7	88.3	88.0	89.2	87.2	89.2
	k-medoids	92.3	91.7	87.4	89.3	88.7	88.0	91.7
	parallel k-means	93.1	91.0	87.9	89.2	88.7	88.0	92.0
5	c-means	88.0	87.2	86.5	86.7	85.3	86.2	87.6
	k-medoids	89.0	90.1	88.7	86.3	86.2	86.0	86.4
	parallel k-means	90.0	90.2	90.7	90.8	90.7	90.2	89.0

The increase in the number of clusters and the increase in the number of fake profiles decreases the accuracy value. Thus, the result indicates that the clustering algorithms provide better accuracy with the 2 clusters (k=2). Also comparing the results, it is clear that parallel k-means clustering offers better and improved accuracy than c-means and k-medoid clustering algorithms. Moreover, the execution time of the parallel k-means algorithm using the MapReduce framework and the traditional k-means clustering executed in a single node is also evaluated. The result indicates that the parallel k-means algorithm reduces the computational overhead and is shown in Fig. 5.

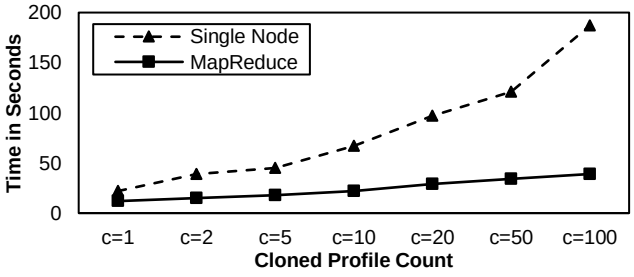


Fig. 5. Execution Time Analysis of the Clustering Process

To analyze the performance of the proposed model, various classifiers are evaluated and their performances are analyzed. The metrics such as accuracy, precision, recall, f-measure, FP rate, TN rate, FN rate and error rate are used for analyzing the performance. The classifiers are trained with a training dataset and the 10-fold cross-validation is used to evaluate each classifier on test data. The results are reported in Table 2.

Table 2. Performance Analysis of Various Classifiers

Classifier	Accuracy	Precision	Recall	F-measure	FP rate	TN rate	FN rate	Error Rate
SVM	98.90	99.17	99.00	99.08	1.25	98.7	1.00	1.10
RF	98.10	98.66	98.17	98.41	2.00	98.0	1.83	1.90
NB	97.20	98.47	96.83	97.65	2.25	97.7	3.17	2.80
KNN	95.70	97.77	95.00	96.37	3.25	96.7	5.00	4.30
DT	94.90	97.25	94.17	95.68	4.00	96.0	5.83	5.10
LR	98.00	98.49	98.17	98.33	2.25	97.7	1.83	2.00
XGBoost	98.40	98.67	98.67	98.67	2.00	98.0	1.33	1.60

From the results, it is clear that though the accuracy of the classifiers such as SVM, RF, LR and XGBoost seems to be similar, the precision, recall and f-measure differ among them. Thus, with the overall analysis, the SVM classifier offers better performance and effective results in identifying fake and real profiles appropriately. The obtained accuracy, precision, recall and the F-measure are depicted as a graph in Fig. 6.

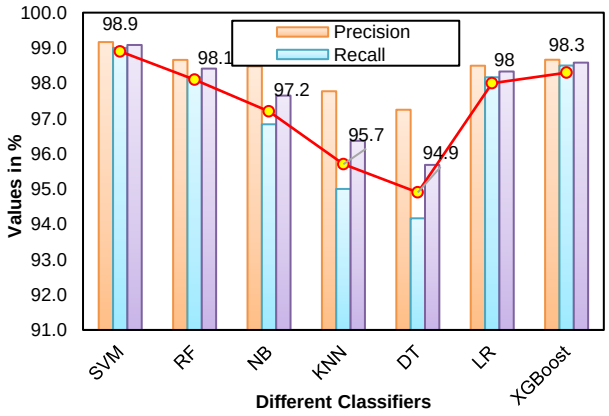


Fig. 6. Performance Evaluation of Different Classifiers

Upon performing classification, the obtained results can be verified by performing the attribute network similarity measures as mentioned in Section 3.5. If the results of the similarity measure are not the same as the classified results, then the profiles can be varied manually.

The proposed cloned profile detection model is evaluated using the MIB Twitter dataset created by a fake project [44]. The dataset contains 1481 real profile data along with 3000 fake profile data that are collected from different sources [45]. Various existing models that are analyzed using this dataset are used for comparing the results of the proposed model. The existing models used for the comparison are minimum weighted features using gain measure [30], correlation-based feature selection [5], PCA-based feature selection [5], MapReduce-based PageRank [36]. The obtained results for the proposed model with various classifiers such as Support Vector Machine, Random Forest, Naïve Bayes, KNN, Decision Tree and Logistic Regression. The results are presented in Table 3.

Table 3. Performance Analysis with Fake Project Twitter Dataset

Method and Dataset used	Classifiers	Accuracy	Precision	Recall	F-measure
Minimum Weighted Feature (Azab et al., 2016)	Random Forest	-	96.16	71.04	81.71
	Decision Tree	-	93.96	67.04	78.25
	Naïve Bayes	-	91.85	61.09	73.38
	Neural Networks	-	92.21	67.63	77.85
	Support Vector Machine	-	86.62	72.83	79.13
Correlation-based feature selection (Homsi et al., 2021)	J48	*98.0*	*98.0*	*98.0*	*98.0*
	Random Forest	*98.7*	*98.7*	*98.7*	*98.7*
	K Nearest Neighbor	93.6	93.5	93.6	93.5
	Naïve Bayes	82.1	82.7	82.1	82.4
PCA-based feature selection (Homsi et al., 2021)	J48	93.1	93.2	93.2	93.2
	Random Forest	95.4	95.5	95.4	95.4
	K Nearest Neighbor	93.5	93.5	93.5	93.5
	Naïve Bayes	83.9	83.8	83.9	83.8
MapReduce based PageRank (Zare et al., 2020)	PageRank	-	97.15	51.55	67.36
Proposed Model	*Support Vector Machine*	*98.19*	*98.96*	*98.33*	*98.65*
	Random Forest	97.30	98.29	97.67	97.98
	Naïve Bayes	96.41	97.62	97.00	97.31
	K Nearest Neighbor	95.51	96.95	96.33	96.64
	Decision Tree	94.62	96.28	95.67	95.97
	Logistic Regression	97.30	98.45	97.50	97.97

The results of the proposed model indicate that the proposed model with SVM classifier offers improved results than other classifiers such as random forest, naïve bayes, k nearest neighbor, decision tree and logistic regression. When comparing the results with the existing models, it is also clear that the proposed model offers improved performance with many other existing models that mainly focuses on feature selection except correlation based feature selection model.

The proposed cloned profile detection model is also compared with various other existing models that employ their own generated dataset collected. Here the generated dataset used for analyzing the proposed model contains 410 real profiles and 700 fake profiles (training and test dataset). The existing models used for the analysis are KNN schema for handling missing values [32], API for chrome extension with multiple attributes [24], Behaviour-based analysis [46], Generic statistical approach [47], Machine Learning Approach [34] and Privacy Protected system [37]. The obtained results are shown in Table 4.

Table 4. Performance Analysis with Synthetic Dataset

Models	Classifier	Accuracy	Precision	Recall	F-Measure
KNN schema (Albayati & Altamimi, 2019)	Decision Tree	96.5	97.41	96.58	97
	K Nearest Neighbor	84	88.99	82.91	85.54
	Support Vector Machine	98.50	97.50	100	98.73
	Naïve Bayes	97.50	95.90	100	97.91
API for chrome extension (Shaoo et al., 2020)	Jrip	*99.5*	*99.6*	*99.6*	*99.6*
	Random Forest	*99.4*	*99.7*	*99.3*	*99.5*
	Logistic Regression	91.9	87.4	99.5	93
	K Nearest Neighbor	98.1	97.4	99.2	98.3
Behaviour-based analysis (Stringhini et al., 2010)	Jrip	87.6	89.4	85.8	87.6
	Bayesian Network	89.1	89.1	89.1	89.1
	Random Forest	89.1	88.8	89.3	89.1
Generic statistical approach (Ahmed & Abulaish, 2013)	Random Forest	97.2	97.4	96.9	97.1
	Bayesian network	97.1	97.3	96.9	97.1
	J48	95.8	95.6	96	95.8
Machine Learning Approach (Singh & Banerjee, 2019)	AdaBoost	-	99	94	99
	Bagging	-	96	96	96
	XGBoost	-	96	96	96
	GradientBoost	-	96	96	96
Privacy Protected system (Suriakala & Revathi,	Random Forest	87.56	-	-	90.45
	Support Vector Machine	84.9	-	-	88.09
	Sequential	78.03	-	-	84.3

2018)	Minimal Optimization Rule Generation algorithm and Node Similarity	97.3	-	-	95.02
Proposed Model	*Support Vector Machine*	*98.90*	*99.17*	*99.00*	*99.08*
	Random Forest	98.10	98.66	98.17	98.41
	Naïve Bayes	97.20	98.47	96.83	97.65
	K Nearest Neighbor	95.70	97.77	95.00	96.37
	Decision Tree	94.90	97.25	94.17	95.68
	Logistic Regression	98.00	98.49	98.17	98.33

The results presented show that an API for the Chrome extension used for data extraction and classification offers improved results. Nevertheless, the proposed model also offers improved results than many other existing fake profile detectors. However, the execution time of the proposed model is very less when compared with the API for the Chrome extension. Thus, the efficiency and performance of the proposed clone profile detection model are still effective and are even more suitable for processing big data with reduced execution time.

5. Conclusion

This paper presents the detection of cloned profiles in online social networks and verification using attribute and network-based similarity. For reducing the computational complexity, the model is implemented using the MapReduce framework. Initially, the system collects the basic profile data of real profile and possible cloned profiles through Facebook graph API and manual search. Moreover, the additional attributes that indicate the relationship strength such as friends count, followers count, like score and graph score are also utilized. Once the dataset is synthesized, then the parallel k-means clustering algorithm is applied to the dataset to group similar profiles. The profiles that are grouped with the real profile are suspicious and thus, the parallel SVM classification algorithm is applied to the cluster containing real profile to classify the fake profiles. Finally, the classified profiles are verified using attribute and network-based similarity measures. The experimental analysis shows that the additional attributes used in the model are effective and the implementation using MapReduce indicates consistent yet less execution time. The use of parallel k-means and parallel SVM also proved to be effective and shows improved performance. Thus, the proposed model offers the accuracy and precision of 98.19% and 98.96% on the MIB twitter dataset and 98.90% and 99.17% on the synthetic dataset created for the study. The future work focuses on using other attributes including multimedia data such as profile images used in other existing models to identify fake profiles. Moreover, the weights can be applied to the additional attributes such as groups in computing more precise scores and counts. Moreover, the accuracy and precision of the proposed model are not 100% and thus future work focuses on improving the accuracy of predicting fake profiles and implementing them with real-time datasets.

References

[1] C. C. Yang, S. M. Holden, M. D. Carter, and J. J. Webb, "Social media social comparison and identity distress at the college transition: A dual-path model," *Journal of Adolescence*, vol. 69, pp. 92-102, 2018.

[2] A. K. Jain, S. R. Sahoo, and J. Kaubiyal, "Online social networks security and privacy: comprehensive review and analysis," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2157-2177, 2021.

[3] T.R. Soomro, and M.Hussain, "Social Media-Related Cybercrimes and Techniques for Their Prevention," *Appl. Comput. Syst*, vol. 24, pp.9-17, 2019.

[4] J.Johnson, "Global digital population", https://www.statista.com/statistics/617136/digitalpopulation-worldwide. Jan 27 2021.

[5] A. Homsi, J. Al Nemri, N. Naimat, , H.A. Kareem, M. Al-Fayoumi, and M.A. Snober, "Detecting Twitter Fake Accounts using Machine Learning and Data Reduction Techniques", *In DATA*, pp. 88-95, 2021.

[6] J. Heidemann, M. Klier, and F. Probst, "Online social networks: A survey of a global phenomenon", *Computer networks*, vol.56, no.18, pp. 3866-3878, 2012.

[7] K .Kaur, and P.Kumar, "Social media: a blessing or a curse? Voice of owners in the beauty and wellness industry", *The TQM Journal*, 2021.

[8] P. Patel, K. Kannoorpatti, B. Shanmugam, S. Azam, and K.C. Yeo, "A theoretical review of social media usage by cyber-criminals", *in 2017 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, pp. 1-6, 2017.

[9] S. Maniraj, P. Harie Krishnan, G. T. Surya, and R. Pranav, "Fake Account Detection using Machine Learning and Data Science", *International Journal of Innovative Technology and Exploring Engineering (IJITEE*), vol. 9, no. 1, 2019.

[10] M. Zabielski, Z. Tarapata, R. Kasprzyk, and K. Szkółka, "Profile cloning detection in online social networks," *Computer Science and Mathematical Modelling*, 2016.

[11] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All your contacts are belong to us: automated identity theft attacks on social networks," *in Proceedings of the 18th international conference on World wide web*, ACM, pp. 551–560, 2009.

[12] L. Jin, H. Takabi, J.B. Joshi, "Towards active detection of identity clone attacks on online social networks", *in Proceedings of the first ACM conference on Data and application security and privacy*, pp. 27-38, February 2011.

[13] M. Conti, R. Poovendran, and M. Secchiero, "Fakebook: Detecting fake profiles in on-line social

networks", presented in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 1071-1078 , August 2012.

[14] S. Gurajala, J.S. White, B. Hudson, B.R. Voter, and J.N. Matthews, "Profile characteristics of fake Twitter accounts", Big Data and Society, vol. 3, pp. 2053951716674236, 2016.

[15] P. Sowmya and Chatterjee, "Detection of Fake and Cloned Profiles in Online Social Networks", *Proceedings 2019: Conference on Technologies for Future Cities (CTFC)*, March 2019.

[16] S.S Bama, and A. Saravanan, "Efficient classification using average weighted pattern score with attribute rank-based feature selection", *International Journal of Intelligent Systems and Applications*, vol. 11, no. 7, pp .29, 2019.

[17] A. Saravanan, S. S. Bama, S. Kadry, and L.K.Ramasamy, "A new framework to alleviate DDoS vulnerabilities in cloud computing", *International Journal of Electrical & Computer Engineering*, vol. 9, no. 5, 2019.

[18] S. Sathya Bama, M.I. Ahmed, and A. Saravanan, "A Mathematical Approach for Filtering Junk E-Mail Using Relevance Analysis", *Asia Pacific Journal of Research*, vol. 1, no. 35, 2016.

[19] A. Rauf, S. Khusro, S. Mahfooz, and R. Ahmad, "A Robust System Detector for Clone Attacks on Facebook Platform", *Journal of Research*, vol.13, no.4, pp. 71-80, 2016.

[20] P. Bródka, M. Sobas, and H. Johnson, "Profile cloning detection in social networks", *in 2014 European Network Intelligence Conference*, pp. 63-68, September 2014.

[21] N. Kumar, and P. Dabas, "Detection and Prevention of Profile Cloning in Online Social Networks", *in 2019 5th International Conference on Signal Processing, Computing and Control (ISPCC)*, pp. 287-291, October 2019.

[22] G. Kontaxis, I. Polakis, S. Ioannidis, and E.P. Markatos, "Detecting social network profile cloning", *in 2011 IEEE international conference on pervasive computing and communications workshops (PERCOM Workshops)*, pp. 295-300, March 2011.

[23] A.N. Hakimi, S. Ramli, , M. Wook, N. Mohd Zainudin, N.A. Hasbullah, N. Abdul Wahab, and N.A. Mat Razali, " Identifying Fake Account in Facebook Using Machine Learning", *in International Visual Informatics Conference*", pp. 441-450, Springer, Cham, November,2019.

[24] S.R. Sahoo, B.B Gupta," Fake profile detection in multimedia big data on online social networks", *International Journal of Information and Computer Security*, vol. 12, no. 2-3, pp.303-331.

[25] Y. Elyusufi, Z. Elyusufi, and M.H.A Kbir, "Social networks fake profiles detection based on account setting and activity", *in Proceedings of the 4th International Conference on Smart City Applications*, pp. 1-5, October,2019.

[26] S. Kiruthiga, and A. Kannan, "Detecting cloning attack in Social Networks using classification and clustering techniques", *in 2014 International Conference on Recent Trends in Information Technology*, pp. 1-6, April 2014.

[27] S. Ranjana, R. Sathian, and M.D. Kamalesh, " Fake Profile Detection in Facebook", *In International Conference on Emerging Trends and Advances in Electrical Engineering and Renewable Energy*, pp. 725-732, Springer, Singapore, 2020, March.

[28] C.R Liyanage, and S.C. Premarathne, "Clustered Approach for Clone Detection in social media", *International Journal of Advanced Science Engineering Information technology*, vol. 11, no. 1, pp. 99-104, 2021.

[29] S.S. Bama, M.I. Ahmed, and A. Saravanan, "A mathematical approach for mining web content outliers using term frequency ranking", *Indian Journal of Science and Technology*, vol. *8*, no. 14, pp.1-5, 2015.

[30] A. ElAzab,"Fake accounts detection in twitter based on minimum weighted feature. World Academy of Science, Engineering and Technology", *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 10, no. 1, pp. 13-18, 2016.

[31] M.Y. Kharaji, and F.S. Rizi, "An iac approach for detecting profile cloning in online social networks", *International Journal of Network Security & Its Applications*, vol. 6, no.1, pp. 75-90, 2014.

[32] M.B. Albayati, and A.M. Altamimi, "An empirical study for detecting fake Facebook profiles using supervised mining techniques", *Informatica*, vol. 43, no. 1, pp. 77–86.03, 2019.

[33] M.B. Albayati, and A.M. Altamimi, "Identifying Fake Facebook Profiles Using Data Mining Techniques", *Journal of ICT Research & Applications*, vol. 13, no. 2, pp. 107-117, 2019.

[34] Singh, and Banerjee, "Fake (Sybil) Account Detection Using Machine Learning", *Proceedings of International Conference on Advancements in Computing & Management (ICACM)*, Available at SSRN:https://ssrn.com/abstract=3462933 or http://dx.doi.org/10.2139/ssrn.3462933, October 2, 2019.

[35] F. Ajesh, S.U. Aswathy, F.M. Philip, and V. Jeyakrishnan , "A hybrid method for fake profile detection in social network using artificial intelligence", *Security Issues and Privacy Concerns in Industry 4.0 Applications*, pp.89-112, 2021.

[36] M. Zare, S.H. Khasteh, and S. Ghafouri, "Automatic ICA detection in online social networks with PageRank", *Peer-to-Peer Networking and Applications*, vol. 13 no.5, 1297-1311, 2020.

[37] M. Suriakala, and S. Revathi, "Privacy protected system for vulnerable users and cloning profile detection using data mining approaches", *in 2018 Tenth International Conference on Advanced Computing (ICoAC)*, pp. 124-132, December, 2018.

[38] A. Zaman, M.A. Siddique, and Y. Morimoto, "Finding key persons on social media by using MapReduce skyline", *International Journal of Networking and Computing*, vol.7, no.1, pp.86-104, 2017.

[39] W. Zhao, H. Ma, and Q. He, "Parallel k-means clustering based on mapreduce", *In IEEE international conference on cloud computing*, pp. 674-679, Springer, Berlin, Heidelberg, 2009 December.

[40] X. Hou, "An improved k-means clustering algorithm based on hadoop platform", *In The International*

Conference on Cyber Security Intelligence and Analytics, pp. 1101-1109, Springer, Cham, *2019, February.*

[41] F.O. Çatak, and M.E. Balaban, "A MapReduce-based distributed SVM algorithm for binary classification", *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 24, no. 3, pp. 863-873, 2020.

[42] MySafeFriend: http://apps.facebook.com/mysafefriend. Accessed: 2012- 01-10.

[43] J. Weaver, P. Tarjan, "Facebook linked data via the graph API. Semantic Web", vol. 4, no. 3, pp. 245-250, 2013.

[44] 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.

[45] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "A Fake Follower Story: improving fake accounts detection on Twitter," 2014.

[46] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks", *in Proceedings of the 26th annual computer security applications conference*, pp. 1-9, December 2010.

[47] F. Ahmed, and M. Abulaish, "A generic statistical approach for spam detection in online social networks", *Computer Communications*, vol. 36, no. 10-11, pp. 1120-1129, 2013.