Hindawi
Security and Communication Networks
Volume 2018, Article ID 5923156, 8 pages
https://doi.org/10.1155/2018/5923156

Research Article

Identifying Fake Accounts on Social Networks Based on Graph Analysis and Classification Algorithms

Mohammadreza Mohammadrezaei,[1] **Mohammad Ebrahim Shiri**,[1,2] **and Amir Masoud Rahmani**[1,3,4]

[1]*Department of Computer, Borujerd Branch, Islamic Azad University, Borujerd, Iran*
[2]*Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran*
[3]*Computer Engineering Department, Science and Research Branch, Islamic Azad University, Tehran, Iran*
[4]*Computer Science, University of Human Development, Sulaymaniyah, Iraq*

Correspondence should be addressed to Mohammad Ebrahim Shiri; shiri@aut.ac.ir

Received 7 April 2018; Accepted 28 June 2018; Published 5 August 2018

Academic Editor: Tom Chen

Copyright © 2018 Mohammadreza Mohammadrezaei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Social networks have become popular due to the ability to connect people around the world and share videos, photos, and communications. One of the security challenges in these networks, which have become a major concern for users, is creating fake accounts. In this paper, a new model which is based on similarity between the users' friends' networks was proposed in order to discover fake accounts in social networks. Similarity measures such as common friends, cosine, Jaccard, L1-measure, and weight similarity were calculated from the adjacency matrix of the corresponding graph of the social network. To evaluate the proposed model, all steps were implemented on the Twitter dataset. It was found that the Medium Gaussian SVM algorithm predicts fake accounts with high area under the curve=1 and low false positive rate=0.02.

1. Introduction

The use of social networks such as Facebook, Twitter, Google+, Instagram, and LinkedIn is on the rise [1, #3]. Individuals and organizations use social networks to express their views, advertise their products, and express future policies of their companies and organizations. By expanding the use of social networks, malicious users seek to violate the privacy of other users and abuse their names and credentials by creating fake accounts, which has become a concern for users. Hence, social networks providers are trying to detect malicious users and fake accounts in order to eliminate them from social networking environments. Creating fake accounts in social networks causes more damage than any other cybercrime {Ramalingam, 2017 #18}.

Removing fake accounts has attracted the attention of many researches; thus, extensive researches have been carried out on the identification of fake accounts in social networks. Different approaches are proposed in [2, #19], [3, #21], [4, #22], [5, #23], and {Kharaji, 2014 #24} to find fake accounts based on attribute similarity, similarity of friend networks, profile analysis for a time interval, and similarity of attribute together with IP address. Kontaxis et al. [6, #25] proposed a scalable approach which can be used to discover a bunch of fake accounts made by a user. Their main technique was a supervised machine learning to classify clusters from malicious or legal accounts. Conti et al. [4, #22] provided a framework for discovering fake accounts based on the growth rate of the social network graph and the interaction of regular users on the network with their friends. Gurajala et al. [7, #26] used map-reduction techniques and pattern recognition approaches to discover fake profiles. To identify fake and actual accounts, the rate of the number of followers as well as collected friends per day was used for each account. They used [8, #27] a combination of pattern-matching (screen-names) and update times analysis in their methodology to discover fake accounts. Kagan et al. [9, #28] offered an unsupervised two-layer meta-classifier method

that can detect unruly nodes in a complex network by using the extracted properties of the graph topology. He also proved that the proposed algorithm is used to detect fake users and can recognize effective users in the network. Boshmaf et al. [10, #29] provided a robust and scalable defense system called "Íntegro" which puts fake accounts at the lowest rank with the use of users ranking. Sakariyah et al. examined the four main categories of malicious accounts on social networks {Adewole, 2017 #30}. Cao et al. [11, #31] introduced a forwarding message tree with six effective features which is used to investigate the relationship between accounts and detect suspicious accounts. The problems in discovering fake accounts in previous researches are stated below:

(1) The use of similarity measures that do not consider the strength of the network of friendships shared among users [3, #21], while we believe that the more the shared friendship network of the two users is connected, the greater the similarity of the users is.

(2) Due to the high volume of information, the use of machine learning techniques leads to overfitting problem [6, #25].

(3) In some previous works, in order to implement the proposed methods, some normal users were assumed to be fake and this is because the number of fake users is lower than that of the fake users in datasets. The above assumption is completely wrong and, thus, will dispute the logic of learning [3, #21], {Kharaji, 2014 #24}. The aim of this article is to provide a model for solving the proposed problems and improve the efficiency of solving them. This paper improves the efficiency of detecting fake accounts on social networks using the proposed method that preprocessed data by (1) using the definition of similarity measures in order to use the strength of relationship among account's friends, (2) using feature extraction methods to prevent the overfitting problem, and (3) generating artificial forged accounts to create a balance in the dataset by using resampling methods.

In the proposed method, according to the graph adjacency matrix, the similarity matrices between accounts were calculated, and then PCA algorithm was used for feature extraction and SMOTE was used for data balancing successively. Then the linear SVM, Medium Gaussian SVM and regression, and logistic algorithms were used to classify the nodes. Finally, the performance of this method was evaluated using various classifier algorithms.

The remaining part of this paper is organized as follows: graph analysis and similarity types are reviewed in Section 2. Section 3 reviews resampling, principal component analysis, and machine learning concepts. The methodology is described in Section 4; in Section 5, the experiments on Twitter dataset are stated, which shows the performance results. Conclusions and future work are presented in Section 6.

2. Graph Analysis and Similarity Types

Graph Analysis is used in many applications, such as displaying circuit diagrams to detect SHAPE, image matching, and social network analysis {Jouili, 2009 #32}. The networks' graph is analyzed in order to solve most of the social network problems. Therefore, graph similarity measures reduce the complexity of graph analysis problems by using different techniques. Some of these graphs are defined below.

A social network G = (E, N) maps into a graph, so that a set of N nodes represents social network users, while the set of edges E⊂ N × N represents the relationships. In addition, the dot sign was used to refer to a particular component in a graph.

(1) A represents the sparse adjacency matrix for graph G. If (v, u) is an edge in G, then A (v, u) =1. Otherwise, A (v, u) =0.

(2) Friendship graph (FG): Considering the social network graph G and a node v∈ $G.N.$, the friendship graph is a vertex containing all vertices that are directly connected to that node and are defined in (1) [12, #33].

$$\begin{aligned} &\mathrm{FG}(v).\mathrm{N} = \{v\} \cup \{n \epsilon \mathrm{G.N} \mid n \neq v. \exists e \in \mathrm{G.E.e} = \langle v.n \rangle\} \\ &FG(v).E = \{\langle v.n \rangle \in G.E \mid n \epsilon FG(v).N\} \\ &\qquad \cup \left\{\left\langle n.n' \right\rangle \in G > E \mid n.n' \in FG(v).\mathrm{N}\right\} \end{aligned} \tag{1}$$

where FG (v). N and FG (v). E denote a vertex containing all vertices that are directly connected to the node v and the relationship between these nodes.

(3) Common friends (CF): One of the measures for similarity in social networks is the number of friends shared. Given a social network G and two nodes v, u∈ $G.N$, all vertices that are on a path with the length of two between these two nodes are common friends of that nodes, as shown in (2) [13, #34], [14, #35].

$$\mathrm{CF}(u.v) = |\mathrm{FG}(v).\mathrm{N} \cap \mathrm{FG}(u).\mathrm{N}| \tag{2}$$

(4) Total friends (TF): It shows the number of different friends between the two v and u nodes as shown in (3) [12, #33].

$$\text{Total friends}(v.u) = |\mathrm{FG}(v).\mathrm{N} \cup \mathrm{FG}(u)\,\mathrm{N}| \tag{3}$$

(5) Jaccard similarity (JS): Jaccard coefficient represents the similarity between the sample sets, and in fact it is used to calculate the ratio of the common friends of the two nodes to their entire friends, as shown in (4) [13, #34].

$$\text{Jaccard} - \text{coef}(v.u) = \frac{|\mathrm{FG}(u).\mathrm{N} \cap \mathrm{FG}(v).\mathrm{N}|}{|\mathrm{FG}(u).\mathrm{N} \cup \mathrm{FG}(v)\,\mathrm{N}|} \tag{4}$$

(6) Cosine similarity: Another similarity measure between nodes is the cosine similarity graph. The cosine similarity actually counts the similarity between the two product vectors as shown in (5) [12, #33].

$$\mathrm{Cos}(v.u) = \frac{|\mathrm{FG}(v).\mathrm{N} \cap \mathrm{FG}(u).\mathrm{N}|}{\sqrt{|\mathrm{FG}(v).\mathrm{N}| \,.\, |\mathrm{FG}(u).\mathrm{N}|}} \tag{5}$$

(7) L1 norm similarity: This measure is obtained by dividing the overlapping part of two nodes according to their sizes as shown in (6) [12, #33].

$$\text{L1 norm}(\text{v.u}) = \frac{|\text{FG}(\text{v}).\text{N} \cap \text{FG}(\text{u}).\text{N}|}{|FG(v).N| \,.\, |FG(u).N|} \quad (6)$$

Edge weight measure: First, the edge weights are calculated as two separate attributes for each of the two edges as shown in (7) and (8) [15, #36].

$$\text{w}(\text{v}) = \frac{1}{\sqrt{1 + FG(v).N}} \quad (7)$$

$$\text{w}(\text{u}) = \frac{1}{\sqrt{1 + FG(u).N}} \quad (8)$$

Then, the weight of the edge between the two vertices of u and v must be calculated in two ways:

total weights: the sum of weights is equal to sum of the two weights which is defined for u and v as shown in

$$\text{W}(\text{v.u}) = \text{w}(\text{v}) + \text{w}(\text{u}). \quad (9)$$

weight coefficient: this parameter is defined in (10); it is multiplication of the two weights defined above, as illustrated in (7) and (8).

$$\text{W}(\text{v.u}) = \text{w}(\text{v}) * \text{w}(\text{u}). \quad (10)$$

3. Introduction of Resampling, Principal Component Analysis, and Machine Learning

3.1. Resampling. One of the problems in data classification is the unbalanced distribution of data, in which items in some classes are more than those of other classes. This problem arises in two-class applications more than the others; it means that one class has more items than the other class. The resampling approach means changing the distribution of training sample sets by processing data. There are several approaches towards improving the class efficiency by balancing the datasets [16, #37]. Resampling data may balance the distribution of the data class by removing the samples of majority class by the use of undersampling approach or increasing the samples of minority class using oversampling to balance. There is another approach known as the minority class artificial sampling which creates the Synthetic Minority Oversampling Technique (SMOTE) of artificial data based on similarity of the characteristics between minority class items. In the proposed model, due to the use of similarity feature of the nodes and the unwillingness to remove information, SMOTE method is used. Due to the replication of minority class samples from the main data in all oversampling approaches, it may increase noise data and processing time and result in overfitting and decrease in efficiency.

Chawla [17, #38] proposed the SMOTE algorithm. This algorithm can randomly create items of a minority class based on certain rule and combine these new sample items with the original dataset to produce new training steps. This approach can be used to produce new minority class items. In minority classes, different samples have different roles in the process of oversampling, and these marginal samples take more roles than the items at the center of minority class. Examples obtained on the margin of a minority class may improve the theme recognition decision and classification rate for minority class prototypes.

3.2. Principal Component Analysis. The key idea of the *principal component analysis (PCA)* is one of the multivariate classical methods and perhaps the most ancient and most popular one [18, #39]. Multiple data analysis has a fundamental role in data analysis. There are many modes or variables in multiple datasets to be observed. If there are n variables in each set of data, each variable can have multiple dimensions. Due to the fact that it is often difficult to perceive multidimensional space, the principal component analysis method reduces the dimensions of all observations based on the combination index and the classification of similar observations [18, #39][19, #40]. The PCA method is one of the most valuable results of linear algebraic application that is used abundantly in all analytical forms, because it is an easy and nonparametric method for extracting relevant information from a complex dataset. In this method, the variables in a multistate space are summed up to a set of unconnected components, each of which is a linear combination of the main variables. The uncorrelated components obtained are the main components which are derived from special covariance matrices or correlation matrices of the main variables. This method is mainly used to analyze the main components of reducing the number of variables and finding a communication structure among the variables. The main components have the largest variance in the entire dataset, and there is no dependence on them. One of the most important issues in the PCA method is selection of the number of core components. Several criteria have been proposed for selecting the number of main components that can be categorized as formal and informal categories. In an unofficial approach, first, an appropriate precision which is suitable for data and the desired results is determined and then the total number of variations is selected based on the cumulative percentage, with the highest precision being considered to be between 80 and 90% of the total variations. Another method used to choose the number of PCs is part of the formal group methods which uses Eigen values higher than one for PC selection called *Rule Kaiser's*.

3.3. Machine Learning. Most machine learning methods train the classifiers by the use of machine learning algorithms. The classifiers are based on various social networks attributes such as attribute similarity, network friend similarity, and IP address analysis. Machine learning classifiers, a number of algorithms which are used in the proposed model, are introduced below.

3.3.1. Support Vector Machine. Support vector machine (SVM) proposed by [20, #41] is a learning algorithm based on

statistical learning theory. SVM implements the principal of structure risk minimization which minimizes the empirical error and the complexity of the learner at the same time and achieves good generalization performance in classification and regression tasks. The goal of SVM for classification is to construct the optimal hyperplane with the largest margin. In general, the larger the margin, the lower the generalization error of the classifier {Huang, 2018 #46}.

In this article, SVM was used with a linear and Gaussian kernel in training. Gaussian uses normal curves around the data points and sums these data points so that the decision boundary can be defined by a type of topology condition such as curves where the sum is above 0.5.

3.3.2. Logistic Regression. Given a set $S=\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ of m training samples with $x^{(i)}$ as feature inputs and $y^{(i)} \in \{0, 1\}$ as labels, logistic regression can be shown as

$$P(y = 1 \mid x, \alpha) = \frac{1}{(1 + \exp(-\alpha^T x))} \quad (11)$$

where $\alpha \in R^n$ are the model parameters.

Without regularization, logistic regression tries to find parameters by using the maximum likelihood criterion, while with regularization, there is a tradeoff between connections, and the variables in the model [21, #43] are fewer.

4. The Proposed Method

Based on the characteristics of a fake account detection problem, our proposed method was introduced in this section. First, the adjacency matrix of the social networks graph was computed. Then, the measures of network friend's similarities between nodes (social network users) were calculated. After that, the similarity matrix was calculated for each of the defined measures such as common friends' similarity, Jaccard similarity, cosine similarity, and other measures. At the end of this step, several matrices that represent similarity between the nodes were shown.

Given that, in such cases, data are not balanced and also about 98-99% of the data belong to the same majority class (normal users), and because the work on such data causes the ignorance of the clarification of minority class (fake users) and the increase of the overall accuracy of classifications, the tagging of all data was labeled normal. To solve this problem, the SMOTE was used to balance the data. The method of creation of an artificial fake user is shown in Table 1.

After applying the SMOTE on each of these similarity matrices and balancing the data, there were seven similarity matrices, as a result, each of which showed seven-point similarity measures. In the previous stage, new similarity features were extracted. Then, with the use of the PCA method, the ten first columns with the highest variance were selected from each of these matrices so that a new property matrix is formed. Then, the data tags were applied and sent to the classifier. In the classification stage, the nodes were classified by using linear SVM algorithm, Medium Gaussian SVM, and logistic regression and then by separating the normal and forged users; the list was sent to the next step.

Table 1: Generation of synthetic examples (SMOTE).

Consider a sample arr1 and let arr2 be its nearest neighbor.
Arr1 is the sample for which K-nearest neighbors are being identified.
Arr2 is one of its K-nearest neighbors.
Arr1= (0.0045, 0.0014, 0.0145, 0.0046)
Arr2= (0.003, 0.0004, -0.0135, 0.0057)
Let: f1_1=0.0045 f2_1=0.003 f2_1-f1_1= -0.0015
f1_2=0.0014 f2_2=0.0004 f2_2-f1_2= -0.001
f1_3=0.0145 f2_3= -0.0135 f2_3-f1_3= -0.028
f1_4=0.0046 f2_4=0.0057 f2_4-f1_4=0.0011
The new samples will be generated as $(f1', f2', f3', f4')$ = Arr1 + rand (0-1) $*$ (-0.0015, -0.001, -0.028, 0.0011)
Rand (0-1) generates a random number between 0 and 1.

Arr1 and Arr2 denote similarity between v1 and v2 four nodes.

Here, the fake users were identified and, in the final step, all their friends were given an alert which informs them of the fake friend. Figure 1 and Box 1 show the steps of the proposed method.

5. Evaluation

Effectiveness of the proposed method is accomplished by machine learning methods. The classifier was trained by 10-fold cross-validation; then efficiency metrics were calculated. In this part, first, cross-validation technique was defined, and the basic metrics and evaluation of the classifier performance were presented.

5.1. Cross-Validation. Cross-validation is a technique used to evaluate predictive models. In this technique, the original samples are divided into two categories: training set for model training and test set for evaluation [22, #44]. The original sample is randomly divided into k subsamples with equal size. One of these subsamples is considered as evaluative data in order to test the model, and the rest of them, k-1 subsamples, are considered as training data. The cross-validation process is repeated k times for k subsamples, each time for one of them as evaluative data. The first advantage of this method is that all samples are used for both training and validation process, and the second one is that each sample is used for validation just once.

5.2. Evaluation Metrics. The evaluation is based on a confusion matrix and associated metrics [23, #45]. The variables TP, FP, TN, and FN in the confusion matrix refer to the following:

true positive (TP): number of fake nodes that are identified as fake nodes,

false positive (FP): number of normal nodes that are identified as fake nodes,

true negative (TN): number of normal nodes that are identified as normal nodes,

false negative (FN): number of fake nodes that are identified as normal nodes.

```
Input:
    Graph G (N, E) with labeled node
Output:
    List of the fake node (account)
       Procedure:
       A⟵ adjacency matrix (G)
       For all similarities
           SM⟵ calculate similarity matrix
          SM ⟵feature extraction by PCA Algorithm on (SM)
       End
       SM ⟵select ten columns in all similarity matrix (SM )
       SM ⟵balance SMOTE on (SM)
       Training Classifier
       Return List of fake nodes
       Send alarm for fake accounts friends.
```

Box 1

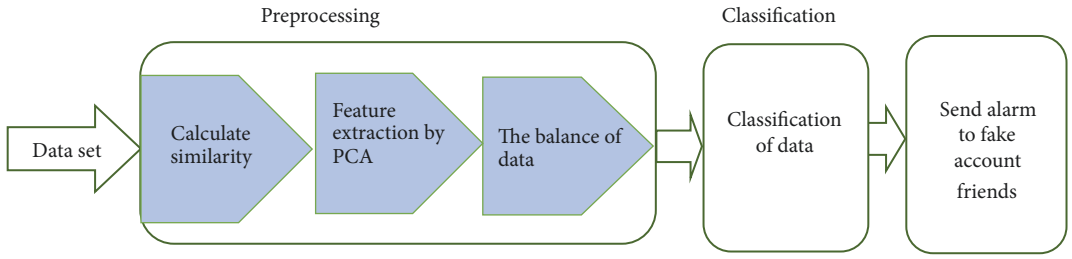


Figure 1: The diagram of the detection approach.

To evaluate the classifier, accuracy and area under curve (AUC) are used. The AUC is performance metrics for binary classifiers; the closer this AUC is to one, the more favorable the final performance of the classification will be. By comparing the ROC curves with the AUC, it captures the extent to which the curve is up in the northwest corner. The metrics which are introduced below are used to calculate the ROC.

True negative rate (TNR) =TN/ (TN+FP).

False positive rate (FPR) =FP/ (FP+TN).

True positive rate (TPR) = TP/ (TP+FN).

False negative rate (FNR) =FN/ (FN+TP).

There is another measure which is used to evaluate the performance:

Accuracy= (TP+TN)/ (TP+FP+TN+FN).

5.3. Performance of the Proposed Model. In order to evaluate the proposed method, Twitter dataset—a real world labeled dataset—was used.

The Twitter data used to support the findings of this study have been deposited in the GitHub repository (https://github.com/kagandi/anomalous-vertices-detection/tree/master/data).

There are 5,384,162 users in this dataset with 16,011,445 links among them. 1,000 nodes were obtained from this dataset where 990 of them were normal and 10 were fake. This dataset has a ratio of 1:100 between normal and fake nodes. Information on the existing relationships between the nodes is shown in this data, then the adjacency matrix of graph is obtained, and the measures of the similarity between nodes are calculated. Then, new features are extracted using the PCA technique. After that, artificial data were generated by using the SMOTE. By applying the SMOTE, data distribution is changed. It means the 99% normal users and the 1% fake users are changed to 75% normal ones and 25% fakes, and these balanced data were sent to the next step. To evaluate the performance of the model, cross-validation technique was used to calculate FPR, TPR, accuracy, and AUC. Comparison of the results of the classifiers showed that some classifiers were more accurate and some others had a higher sublevel AUC. The closer this sublevel gets to one, the higher the performance accuracy is. Figures 2–5 show AUC diagram for three algorithms.

Table 2: Comparison of performance of classifier.

AUC	Accuracy	FPR	TPR	Algorithm
98%	95.8%	4%	96%	Linear SVM
1	97.6%	2%	97%	Medium Gaussian SVM
96%	96.6%	3%	94%	Logistic Regression

Table 2 and Figure 5 show the testing TPR, FPR, accuracy, and AUC for the three algorithms. The use of nonlinear methods such as Medium Gaussian SVM, due to the fact that they map data to higher-dimensional feature spaces, has higher ability to differentiate the data and results in the performance of the model using this method rather than linear methods such as linear SVM.

Table 3: The results of comparison of Medium Gaussian SVM in two cases, balance/unbalance data.

Algorithm	TPR	FPR	Accuracy	AUC	Balance/unbalance data set
Medium Gaussian SVM	97%	2%	97.6%	1	Balance
Medium Gaussian SVM	0	0	99%	47%	Unbalance

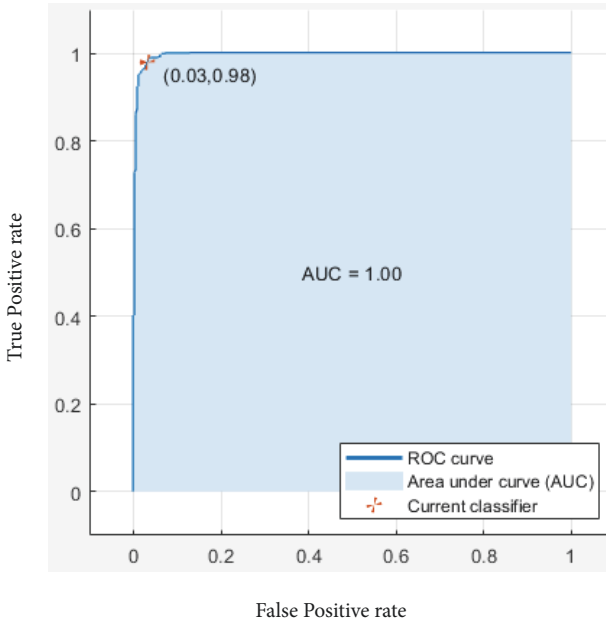


Figure 2: AUC diagram for Medium Gaussian SVM classifier.

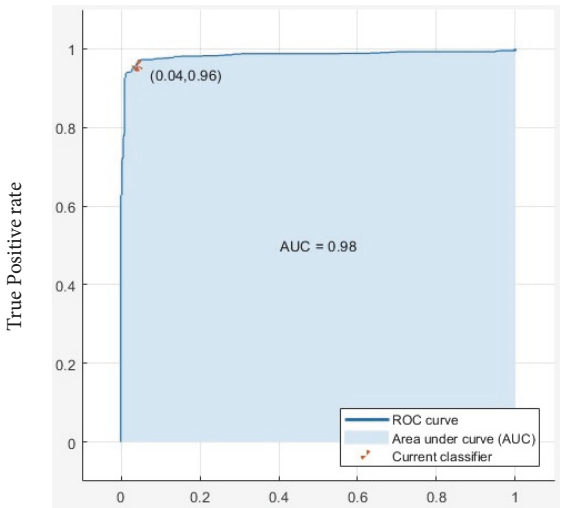


Figure 3: AUC diagram for linear SVM classifier.

According to the obtained values in Table 3 and Figure 6, the unbalanced data was the reason for ignoring the data from the minority class by classifier, predicting all with the normal label, and not labeling any node with fake label. This act only increases the accuracy of the whole system. Figure 7 shows

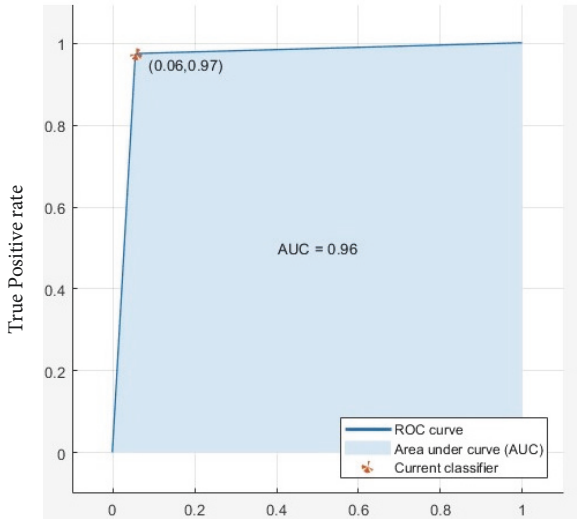


Figure 4: AUC diagram for logistic regression classifier.

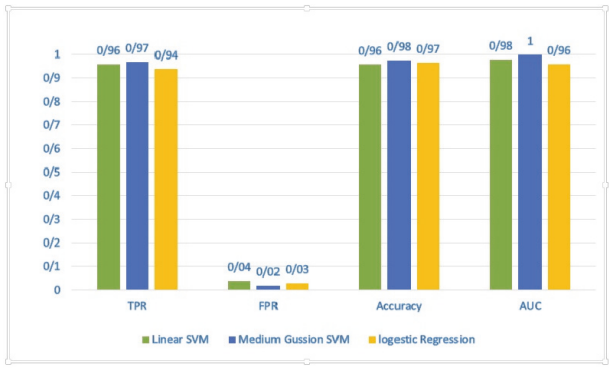


Figure 5: Comparison of performance of classifiers.

that the performance of classifier is very low in the case where data are unbalanced.

6. Conclusion and Future Work

To identify fake accounts in social networks, a method based on similarity of the user's friends was provided. In this method, at first, friend similarity criteria were calculated from the adjacency matrix of the network graph and new features were extracted from the PCA method. In the following stage, the data were balanced using the SMOTE and sent to the classifier. Using the cross-validation technique, the classifier was trained and tested, which showed that the Medium Gaussian SVM classifier has an AUC = 1. In the proposed

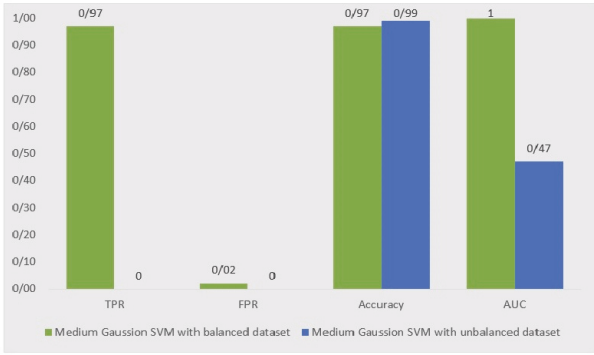


Figure 6: Comparison of Medium Gaussian SVM in two cases, balance/unbalance data.

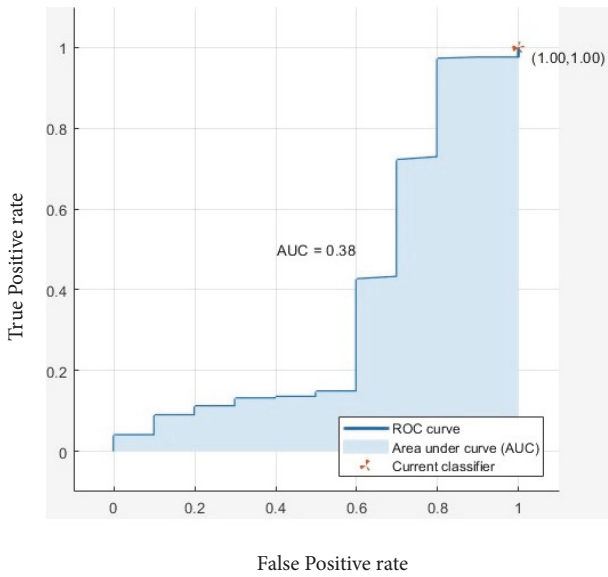


Figure 7: AUC diagram for Medium Gaussian SVM with unbalance data.

method, the user friend network structure was analyzed and the fake users were predicted by computing similarity and the classifier algorithms. In this method, fake accounts must work in the network so that it will be possible to recognize them as legitimate or fake ones, by analyzing their friend's networks. This is a weakness of the proposed method. In future researches, a new method will be presented; which can recognize the legitimate or fake account before any activity of the user in the network or at the time of registration.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. H. Parmelee and S. L. Bichard, *Politics and the twitter revolution: How tweets influence the relationship between political leaders and the public*, Lexington books, 2011.

[2] 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*, pp. 551–560, Madrid, Spain, 2009.

[3] L. Jin, H. Takabi, and J. B. Joshi, "Towards active detection of identity clone attacks on online social networks," in *Proceedings of the the first ACM conference*, p. 27, San Antonio, TX, USA, Feburary 2011.

[4] M. Conti, R. Poovendran, and M. Secchiero, "FakeBook: Detecting fake profiles in on-line social networks," in *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012*, pp. 1071–1078, Turkey, August 2012.

[5] Z. Shan, H. Cao, J. Lv, C. Yan, and A. Liu, "Enhancing and identifying cloning attacks in online social networks," in *Proceedings of the the 7th International Conference*, pp. 1–6, Kota Kinabalu, Malaysia, January 2013.

[6] G. Kontaxis, I. Polakis, S. Ioannidis, and E. P. Markatos, "Detecting social network profile cloning," in *Proceedings of the 3rd International workshop on security and social networking*, USA, 2011.

[7] S. Gurajala, J. S. White, B. Hudson, and J. N. Matthews, "Fake Twitter accounts: Profile characteristics obtained using an activity-based pattern detection approach," in *Proceedings of the 2015 International Conference on Social Media & Society (SMSociety'15)*, Toronto, Ontario, Canada, 2015.

[8] S. Gurajala, J. White, B. Hudson, and J. Matthews, "profile characteristis of fake Twitter accounts," *Big Data Society*, pp. 1–13, 2016.

[9] D. Kagan, Y. Elovichi, and M. Fire, "Generic anomalous vertices detection utilizing a link prediction algorithm," *Social Network Analysis and Mining*, vol. 8, no. 1, 27 pages, 2018.

[10] Y. Boshmaf, D. Logothetis, G. Siganos et al., "Íntegro: Leveraging victim prediction for robust fake account detection in large scale OSNs," *Computers & Security*, vol. 61, pp. 142–168, 2016.

[11] J. Cao, Q. Fu, Q. Li, and D. Guo, "Discovering suspicious Account in online social networks, Information Science," *Information Science*, pp. 1–23, 2017.

[12] C. G. Akcora, B. Carminati, and E. Ferrari, "User similarities on social networks," *Social Network Analysis and Mining*, vol. 3, no. 3, pp. 475–495, 2013.

[13] J. Santisteban and J. Tejada-Cárcamo, "Unilateral weighted Jaccard coefficient for NLP," in *Proceedings of the 14th Mexican International Conference on Artificial Intelligence (MICAI '15)*, pp. 14–20, IEEE, Cuernavaca, Mexico, October 2015.

[14] Liyan Dong, Yongli Li, Han Yin, Huang Le, and Mao Rui, "The Algorithm of Link Prediction on Social Network," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–7, 2013.

[15] W. Cukierski, B. Hamner, and B. Yang, "Graph-based features for supervised link prediction," in *Proceedings of the International Joint Conference on Neural Network (IJCNN '11)*, IEEE, San Jose, Calif, USA, 2011.

[16] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational Intelligence An International Journal*, vol. 20, no. 1, pp. 18–36, 2004.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[18] I. Jolliffe, *Principal Component Analysis*, 2002.

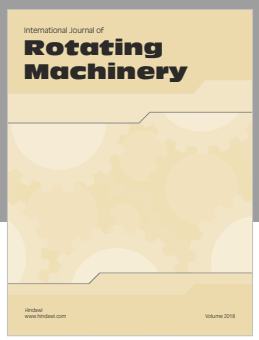
[19] L. Sadowski, M. Nikoo, and M. Nikoo, "Principal Component Analysis combined with a Self Organization Feature Map to determine the pull-off adhesion between concrete layers," *Construction and Building Materials*, vol. 78, pp. 386–396, 2015.

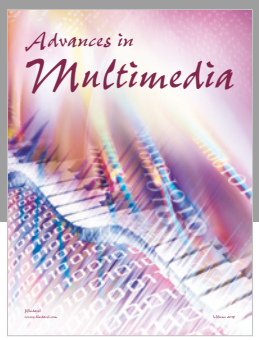
[20] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, NY, USA, 1998.

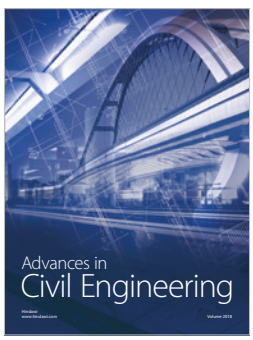
[21] S. Sperandei, "Understanding logistic regression analysis," *Biochemia Medica*, vol. 24, no. 1, pp. 12–18, 2014.

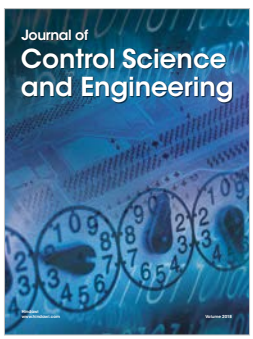
[22] R. Kohavi, "A study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *Proceedings of the in 14th international joint conference on Artificial intelligence*, pp. 20–25, 1995.

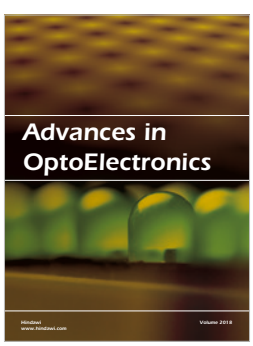
[23] E. Costa, A. Lorena, A. Carvalho, and A. Freitas, "A Review of Performance Evaluation Measures for Hierarchical Classifiers," *Association for the Advancement of Artificial Inteligence*, 2007.

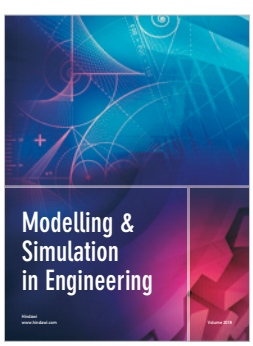


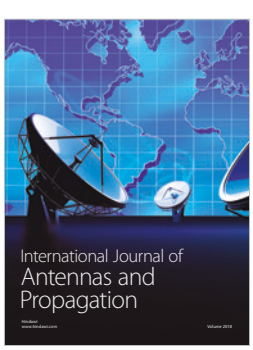


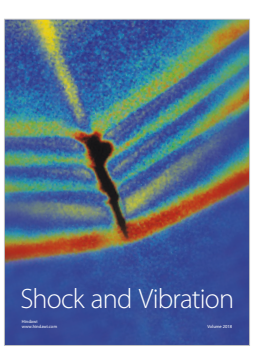


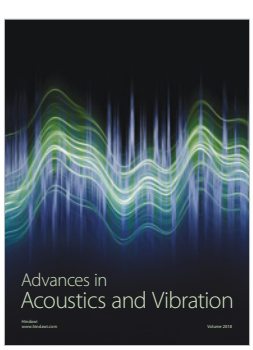












Hindawi

Submit your manuscripts at
www.hindawi.com