ISSN: 2472-3703 DOI: 10.6911/WSRJ.202405_10(5).0004

# Social Bot Detection Techniques Incorporating Friendship Preferences

Teng Wang[1, *], Zhiyong Zhang[1]

[1]Information Engineering College, Henan University of Science and Technology, Luoyang, Henan, 471023, China

* Corresponding author: 2898129821@qq.com

## Abstract

**Social bots are automated accounts that mimic human behavior on social networks. Despite advancements in social bot detection, current state-of-the-art methods still face challenges in early detection, universality, and robustness, and are limited by their reliance on content text. In this paper, we combine friendship preference features with content features while utilizing a classifier based on Bayesian and improved threshold optimization algorithms. Experimental results demonstrate the outstanding performance of the classifier in recognition tasks. Cross-validation on various datasets validates the classifier's generalization ability and robustness, and discusses its capability for early detection of social bots. Finally, the algorithm's classification performance is shown to outperform other state-of-the-art detection methods on multiple datasets.**

## Keywords

**Social bot, social bot detection, fake account detection, social network security.**

## 1. INTRODUCTION

With the rapid development of Internet technology, especially the widespread use of mobile networks, the general public can easily and without obstacles access the Internet and join social networks. In the past few years, the number of social network users has increased dramatically, with almost every citizen having his or her own social account and billions of people around the world using various social platforms. Many illegal behaviours have come to light, including the use of social networks to promote personal or commercial interests [1-4], illegal collection of personal data [5], dissemination of false information [6-8], distribution of malware [9], and manipulation of public opinion [10-12]. In the vast resource of social network users and data, there exists a large number of automated accounts known as social bots. These accounts are usually operated by real people or controlled by software processes. Some of these bot accounts have malicious purposes, and they benefit the hidden manipulators by spreading irrelevant or harmful content. Research [13] states that up to 15% of Twitter accounts are actually operated by social bots rather than real people. These bots manipulate the Twitter platform ecosystem by changing metrics such as the influence and popularity of accounts. It has been reported that 71% of accounts on Twitter could be controlled by bots by 2019 [14]. The development of detection techniques for social network bot accounts has become particularly urgent in light of the fact that malicious social bot accounts pose a real threat to social stability, financial security, and personal privacy [15-16].

Currently, the detection of social robots has become a hot topic. Despite initial research results, as social robots become more intelligent, existing detection methods need to cope with the ever-changing robots. While pursuing detection accuracy, researchers often overlook the importance of generality, robustness, and early identification of methods. Existing studies have shown that improving detection accuracy usually requires collecting large amounts of feature information or using computationally expensive methods [17]. However, in addition to accuracy, generality, robustness and early identification of detection methods are also crucial. As the number of user accounts and their interconnections increase, detection methods need to better address scalability challenges. Meanwhile, emerging social bots aim to circumvent existing detection mechanisms [18]. Therefore, feature-versus-avoidance strategies play a crucial role. The versatility of detection methods is also particularly important due to the rapid development of evasion techniques and the diverse changes in application scenarios. In addition, if malicious social bots can be identified at an early stage, they can be throttled before they cause harm to the social network, which can ensure the safe and healthy operation of OSNs.

The main contributions of this paper are as follows:

1. The paper introduces a detection method that combines friend preference features with content features and utilizes a classifier based on Bayesian and enhanced threshold optimization algorithms. This method efficiently detects tweets containing text, achieving good results in terms of robustness, universality, early detection, and accuracy.

2. Cross-validation is conducted using diverse datasets collected at different time intervals to showcase the algorithm's generalization capabilities. The study explores the significant challenges of bypassing content and friend preference features, validates the algorithm's robustness, and enables early detection even in the absence of malicious activity.

This paper is divided into six main sections: introduction, related research, technical description, experimental validation, conclusion, and future research directions. The introduction section describes the background, importance, and current research status of the study and briefly describes the content of this study; in the related research section, the existing techniques are reviewed and the necessity and innovation of the study are emphasised; the technical description section exhaustively describes the proposed detection method and compares its differences with the existing techniques; the experimental validation section validates the proposed algorithm through experiments that The experimental validation section validates the effectiveness of the proposed algorithm through experiments, including the setting of the experimental environment and data set, and analyses the experimental data in order to draw conclusions; the conclusion section summarizes the research results of this paper and proposes the future research directions.

## 2. RELATED WORK

The proliferation of bot accounts on social platforms has triggered a significant amount of research work on their identification. Along with the advancement of AI technology, the steganography and detection techniques of bots are also developing rapidly. From the initial research, these methods can be broadly categorised into the following distinct categories: graph-based detection methods and feature-based detection methods.

Graph-based methods. Centred on the social relationship graph, which is based on a graph of associations established between users in a social network, the graph helps to gain insight and assess the interactive connections between users on social media. However, some scholars have argued that this model does not fully match the characteristics of real-world social networks [19-21]. Cai and Jermaine [22] argued that OSNs actually consist of many localised communities rather than two large, highly interconnected communities. Therefore, detection techniques based on social relationship graphs focus on the relationships between users; after all, no account exists in isolation in a social network, they are connected to each other, and the social relationship graphs of normal users and machine accounts are very different. However, there are some limitations to these approaches; graph-based detection techniques require a large amount of labelled data to construct and train the model, and obtaining this high-quality data is both time-consuming and expensive. In addition, when constructing a graph, the representation of nodes and edges and the hierarchical structure of the graph must be determined; these choices can affect the performance of the detection and there are no uniform standards or best practices.

Feature-based approaches. Various features are extracted from user profiles, social interactions, and web content through either supervised or unsupervised learning. Supervised learning relies on labelled datasets to identify bot accounts and human accounts [23-26], and commonly used classification algorithms include Random Forests, Support Vector Machines, and K Nearest Neighbours algorithm (KNN). On the other hand, unsupervised learning is able to reveal latent patterns in the absence of labelled data [27-28], often using clustering algorithms to find meaningful clusters based on the similarity between features (defined by a suitable distance metric). This approach captures the behavioural patterns and features of social robots even as they become more advanced and are able to mimic the behaviour of real users, adapts to the evolution of social robotics by constantly updating features and models, and maintains the accuracy and effectiveness of the detection system. Although feature-based detection methods perform well in specific contexts and provide efficient automated detection solutions, they have limited universality capabilities and may require model adaptation or retraining when faced with new scenarios or different detection challenges.

# 3. FRIENDSHIP PREFERENCE BASED SOCIAL BOT DETECTION ALGORITHM

## 3.1. Preprocessing data

The extracted features are obtained from the profile attributes of the fans of a particular account. Table 1 demonstrates the set of profile attributes used. In addition to the friendship preference features, certain profile-based features derived directly from the profile attributes of the surveyed accounts are also utilised. Certain attribute values may be missing and therefore need to be transformed and filtered in order to calculate the probability distribution function for each attribute. The following are the steps to prepare the feature extraction data:

(a) Conversion of numerical attributes to classification types: first, the number of classification bins is determined. the Sturges method and the Freedman-Diaconis method are used to estimate the optimal number of bins.

The Sturges method was proposed by Sturges in 1953 and its estimated number of boxes H is the logarithm of the number of data n plus 1. Whereas the Freedman-Diaconis method was proposed by Freedman and Diaconis in 1981 and is given by

$$H = \frac{2IQR\left(x\right)}{\sqrt[3]{n}} \tag{1}$$

Where IQR denotes the interquartile range of the data. For small datasets (up to 1000 members), the Sturges method was used in this study to determine the number of boxes as it is more conservative and prevents over-segmentation of small datasets. Whereas for large datasets, the Freedman-Diaconis rule was used as it performs better with large amounts of data. After determining the number of bins, the numerical attributes are partitioned according to bins, where each numerical attribute value is assigned to a bin, and the range of values in each bin represents a classification.

(b) Conversion of URL attributes to binary type: for URL attributes, the presence or absence of each URL is converted to a binary value. If the URL exists (i.e., is not null), the binary value is 1; if the URL does not exist (i.e., is null), the binary value is 0.

(c) Conversion of creation date attribute to account lifetime: This is achieved by calculating the time interval from the creation date to the current date, using the current date minus the creation date. This time interval represents the lifetime of the account, i.e. how long the account has existed. The numeric attribute of the account lifetime is then converted to a categorical type according to step (a). By converting the creation date attribute to a binary type feature of the account lifetime, the duration of the account's existence can be better represented.

(d) Replacement of the description attribute with its length: a feature representing the length of the description is obtained by calculating the number of characters or words of the description text. Choose to calculate the number of characters or words of the description text. When counting the number of characters, all characters in the description text (including spaces and punctuation) are counted. When counting the number of words, the description text is segmented by spaces or other separators and the number of words after segmentation is counted. These length values are then converted to categorical types according to process (a), which provides a better representation of the informative size of the description by replacing the description attribute with a binary type feature of its length.

**Table 1.** Summary document attribute set

| Attribute Name | Pretreatment | category |
|---|---|---|
| Nickname | a | FP |
| Personal Description | d, a | FP |
| Creation time | c, a | FP |
| Friends | a | FP |
| Fans | a | FP |
| Favourites | a | FP |
| Likes | a | FP |
| Personal Description | d, a | P |
| Friends | a | P |
| Fans | a | P |
| Favourites | a | P |
| URL | b | FP |

## 3.2. Process of feature extraction

Suppose there are N users, M attributes and K content features are considered in the social network. For each user i, we denote its attribute vector as $x_i = [x_{1i}, x_{2i} ... x_{Mi}]$, and the content feature vector is denoted as $y_i = [y_{1i}, y_{2i}... y_{Ki}]$.

First, correlation analysis is used to select the most discriminative features from among the M attributes and K content features, and obtain the selected feature set S. For each feature in S, principal component analysis is used to determine its relative importance and obtain the weight vector $w = [w_1, w_2, ..., w_{|S|}]$. The baseline PDF is set as the distribution of features in a randomly selected subpopulation of the target social network, denoted as PDF base = $[p_1, p_2, ..., p_{|S|}]$, where pi denotes the probability value of the baseline PDF on the ith feature. For each user i, compute its attribute vector xi and content feature vector yi on the feature set S. The probability distribution vector $PDF_i = [p_{1i}, p_{2i}, ..., p_{|S|i}]$, in this paper, we use the two-sample Kolmogorov-Smirnov test, which will be validated in subsequent experiments. Use the cmp function to compare PDFi and PDF base to get the feature vector $F_i = [f_{1i}, f_{2i}, ..., f_{|S|i}]$. The feature vector Fi is weighted and summed with its weight vector w to obtain the final feature vector $F'_i = [f'_{1i}, f'_{2i}, ...,$

$f'_{|S|i}$], where $f'_{ji}=f_{ji}*w_j$. Finally, the feature vectors of all users F' = [$F'_1$, $F'_2$, ..., $F'_N$] are normalised to ensure the scale consistency of features across attributes and content features. The selection of content features is shown in Table 2.

**Table 2**. Content feature attribute set

| Attribute Name | Preprocessing |
|---|---|
| Semantics of tweets | a |
| Vocabulary use | a |
| Syntactic rules | a |
| Length of tweets | a |
| Source of information | a |

Advantages of using plain Bayesian theory to construct a recognition model include: firstly, the model follows the key assumption of plain Bayesian theory, which is that individual target values are independent of each other when performing a priori probability calculations; secondly, in contrast to methods such as decision trees and logistic regression, a plain Bayesian model is able to make predictions across multiple categories at the same time, rather than just for binary classification problems. This gives it an advantage when dealing with multi-category classification tasks; finally, the plain Bayesian model has good scalability and remains valid even if the set of feature attributes changes, which makes the model more flexible in coping with new feature attributes or adapting existing ones.

The recognition process of the Bayesian-based classifier is as follows: firstly, the optimisation methods such as genetic algorithm and simulated annealing algorithm are used to optimise generation by generation in order to obtain the threshold matrix for distinguishing between bots and real users. This threshold matrix is used to judge whether the data belongs to bots or real users; using the classification algorithm based on the Bayesian model, the probability matrix is obtained by recording the frequency of the feature attributes of the machine account and the real account falling in some thresholds respectively, and the inconsistency of the sampling numbers of the machine account and the real account is resolved by the confusion matrix, meanwhile, the F1 is obtained. The detailed flow is shown in Figure 1.

Firstly, based on the number of feature attributes and the threshold value of the corresponding attribute to establish the threshold matrix W of the real existence of the user, and set the rows and columns are 0 to get the probability matrix T of the real account, and from the result of the manual recognition to get the real account set $f_1$, where, for the real user data, the probability of the ith feature attribute falls on $W_{ij}$ and $W_{i(j+1)}$, denoted by $T_{ij}$, and similarly, the result of the manual recognition to get the set of machine accounts $f_2$, the machine threshold matrix L of machine accounts can be established, and the probability matrix B of machine users is obtained by setting both rows and columns to 1. Based on the robot user data, the probability that the ith feature attribute falls on $L_{ij}$ and $L_{i(j+1)}$, denoted by $B_{ij}$. (i denotes rows and j denotes columns).

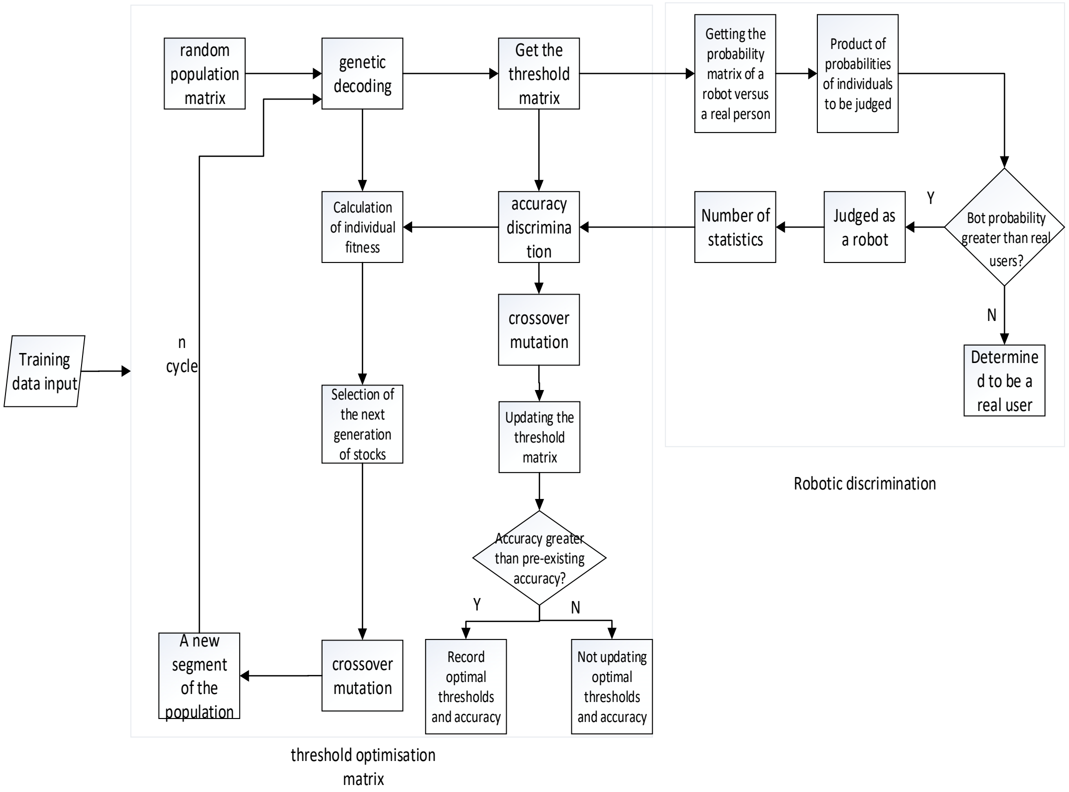
random population matrix
genetic decoding
Get the threshold matrix
Getting the probability matrix of a robot versus a real person
Product of probabilities of individuals to be judged
Calculation of individual fitness
accuracy discrimina tion
Number of statistics
Judged as a robot
Y
Bot probability greater than real users?
N
Determine d to be a real user
crossover mutation
Training data input
n cycle
Selection of the next generation of stocks
Updating the threshold matrix
Robotic discrimination
Accuracy greater than pre-existing accuracy?
Y
N
A new segment of the population
crossover mutation
Record optimal thresholds and accuracy
Not updating optimal thresholds and accuracy
threshold optimisation matrix

**Figure 1.** Classifier framework

The data x represents the set of mixed machine account data and real user account data, x = $\{c_1, c_2, ...., c_{m-1}, c_m\}$, the value of the feature attribute is denoted by c. $y_1$ and $y_2$ are the data is real user data and robot account data, respectively, comparing $P(y_1|x)$ and $P(y_2|x)$ and taking the larger value, which is given by Bayes' theorem:

$$P(y_i \mid x)=\frac{P(x \mid y_i)\,p(y_i)}{P(x)} \tag{2}$$

We assume that the feature attributes are directly independent of each other and P(x) is a constant, and since the machine and real accounts in the selection data are deterministic and fixed, then $P(y_i)$ is also deterministic, and the value of $P(y_i|x)$ varies with the value of $P(x_i|y)$.

$$P(x \mid y_i)P(y_i)=P(c_i \mid y_i)\ldots P(c_m \mid y_i) \tag{3}$$

$$P(c_j \mid y_i)P(y_i)=P(y_i)\prod\nolimits_{j=1}^{m}P(c_j \mid y_i) \tag{4}$$

In the above formula, the probability of $c_j$ in the jth attribute is represented by the probability matrix $P(c_j|y_i)$, and the number of machine accounts is added to 1 if it is judged to be a social bot, otherwise, the number of real accounts is added to 1 to get the number of bots and the number of real people data.

### 3.3. Improved threshold optimisation algorithm

Based on the in-depth analysis of the attribute characteristics of the collected data and Bayesian theory, this study improves the initial threshold matrix by combining the genetic algorithm and simulated annealing algorithm, and by performing individual selection and cross mutation operations through the genetic algorithm in each round of iteration, to achieve a better searching effect during the whole optimisation process, so as to improve the accuracy of the classification model.

When using genetic-based algorithms and performing threshold optimisation, a threshold matrix var with i rows and 4 columns is obtained based on the number of attributes, and four attributes can take the values $\text{var}_{i,1}$, $\text{var}_{i,2}$, $\text{var}_{i,3}$, and $\text{var}_{i,4}$, where $\text{var}_{i,1}=0$, and $\text{var}_{i,3}$ and $\text{var}_{i,4}$ denote the maximum value of the attribute corresponding to the robot account and the real user account, respectively. A population matrix P with 8 rows and 12 columns with values ranging between 0 and 1 was randomly generated, with the rows and columns representing the 8 individuals and the 3 genetic values of the attributes for each individual, and the baseline value of the attribute was represented by S.

$$\text{var}_{i,2} = \frac{S+1}{16\,\text{var}_{i,3}} \tag{5}$$

The threshold matrix var will be obtained and fed into the social robot detection model for evaluation, the detecting individuals get different length representations between 0 and 1 according to their accuracy, the individuals in the interval P are randomly selected to participate in the next selection, the individuals with high accuracy get longer lengths, and if the probability of being selected is large, the chance of inheriting their high genes to the next generation is even greater, and a random operation will be performed on the new population P, including the lengths of any two lines Swap and mutation of a certain row of genes as a result of this new population P1 to continue the next cycle, at the same time, to obtain and record the current optimal threshold matrix and the highest accuracy rate.

To continue the threshold optimisation using the simulated annealing algorithm, set the initial temperature T and the total number of iterations N.

a. Under T, stochastic operations are performed on the population matrix, including swapping two rows and gene mutation. Generate a new population matrix and calculate the accuracy.

b. Determine whether to accept the new solution: compare the difference between the accuracy of the new solution and the current best accuracy. If the accuracy of the new solution is higher, accept the new solution directly. If the accuracy rate of the new solution is lower, accept the new solution with a certain probability.

c. Reduce the temperature T according to the strategy of the simulated annealing algorithm.

d. Update the threshold matrix var according to the new population matrix and record the current best threshold matrix and accuracy rate.

e. Repeat steps a-d until the set total number of iterations N is reached.

$$\text{var}_{i,2} = avg + k \times std \tag{6}$$

Calculate the mean avg and standard deviation std of $\text{var}_{i,2}$ in the current optimal matrix, set the constant k. Setting the range of k to between (0, 1) allows for a dynamic relaxation of the limit of var to expand the search space or for better fine tuning. After completing N rounds of iterations, the output of the model can be obtained when the threshold matrix and accuracy rate tend to be stable.

### 3.4. Algorithm Pseudo-Code

In this section, the classifier is trained by a machine learning approach that accurately distinguishes between bots and real users, a process commonly used in online environments such as social media, forums, or email systems. As described in Algorithm 1, a machine learning model based on genetic algorithms is described and incorporates the incorporation of diversity preservation, parallelisation of processing and optimisation of reproduction strategies, aiming to improve the detection of bots' behaviour through iterative optimisation.

Algorithm: Social Bot Detection Algorithm

```
Input： Contains the set of bot accounts and normal users N
Output： set of bots
for i ← 1 to n
$S = \max \partial_{M,K} = \frac{E(M-\mu_M)(K-\mu_K)}{\partial_M \partial_K}$          //Get the set of selected features
$\mathrm{cov}(i,i+1) = \frac{\sum_{b=1}^{a}(X_b-\bar{x})(Y_b-\bar{y})}{a-1}$          //Proof of relevance
$w = [w_1, w_2, ..., w_{|S|}]$ // Generate weight vector
$F_i$ = cmp (PDF$_i$, PDF base)
$F_i' = \sum_{i=1}^{s} w \times F_i$
$F = [F'_1, F'_2, ..., F'_n]$
endfor
for j ← 1 to m
  $P(c_j | y_i)P(y_i) = P(y_i)\prod_{j=1}^{m}(c_j | y_i)$
  for P($c_j$|$y_i$) > var do
    count++
  endfor
endfor
```

## 4. EXPERIMENT

### 4.1. Experimental data

Table 3 presents a basic overview of the data set and also includes the categorisation of the data set and its accuracy assessment. It can be observed through the table that there is a significant increase in the percentage of suspended or removed fake accounts compared to real accounts. It is supposed that Twitter's prevention and control measures may be the main reason for the susp ension or removal of fake accounts. Thus, those fake accounts that are suspended or removed are more likely to be accounts with distinctive bot features, while the bot accounts that continue to exist are closer to mimicking human behaviour.

Cresci et al. constructed a labelled data set containing both real and fake accounts, which we call Cresci2017 [29-30]. The classification of this data set includes real accounts (genuine), fake fans (fake follower), spambots, and traditional spambots. Fake followers refer to fake accounts that are used to artificially increase the number of followers of another account, and spambots refer to fake accounts that are controlled by spambots.

This study validates the ability to identify newly created fake accounts, where Spambots and Traditional Spambots represent two types of fake accounts generated at different time periods and independent of each other.

**Table 3.** Data set selection

| Tab | H/F | Real? | All | Select |
|---|---|---|---|---|
| Real users | H | Yes | 3474 | 1025 |
| Fake Fans | F | Yes | 3351 | 231 |
| Spambot | H | Yes | 4912 | 2600 |
| Traditional Spambot | F | Yes | 2631 | 1123 |

## 4.2. Experimental analyses

The efficiency and usefulness of the proposed feature classes are evaluated by using five different classifier methods and three cmp functions. Detection performance is evaluated on the collection data set presented in 4.1. Three pairs of real and fake account sets were formed, each represented by a combination of real and fake account tag names.

As can be seen in Tables 4, 5, and 6, the best f-measures obtained by different classification methods are shown in bold. The results are shown in Figures 2, 3, and 4, respectively. In almost all cases, the bayesian and improved threshold optimisation based algorithm classifiers in this paper outperform the other classifiers. However, in some cases, the f-measure results of this paper using the classifiers and Random Forest are the same and differ more than the Random Forest results in Genuine-Fake Follower. The reason for getting the worst results using the KNN classifier could be its sensitivity to noise and irrelevant features. As this paper references friendship preference features, these features are scalable, derivable and robust in the early stages, allowing the algorithm to identify the account before it carries out a malicious act, enabling early detection; experiments have been conducted on different datasets and multiple classifiers with good results, validating the algorithm's universality and robustness.

Table 7 gives a comparison of the f-measures of Botomer, the stand-alone approach and the friendship preference detection based algorithm for each data set, the line graphs of the comparison on the different datasets are shown in Figure 5, and Figure 6 gives a general overview of the comparison, and the results show that friendship preferences represent another aspect of Twitter accounts, which is a valuable distinction between fake and real accounts, and that, moreover, they cannot be be resolved by any of the 1150 features utilised by botometer, furthermore, combining content features while utilising a classifier based on bayes and improved threshold optimisation algorithms, the recognition results outperform the other baseline algorithms on all three datasets.

**Table 4.** Considering cmp-functions and classification algorithms in Genuine-Fake Follower comparison

| Genuine-Fake Follower | RF | AdaBoost | SVM | KNN | Bayes |
|---|---|---|---|---|---|
| KS | 0.96 | 0.95 | 0.94 | 0.93 | 0.96 |
| Chiz | 0.95 | 0.94 | 0.89 | 0.89 | 0.94 |
| Entropy | 0.96 | 0.96 | 0.95 | 0.94 | 0.96 |

**Table 5.** Considering cmp-functions and classification algorithms in Genuine Spambot comparison

| Genuine Spambot | RF | AdaBoost | SVM | KNN | Bayes |
|---|---|---|---|---|---|
| KS | 0.99 | 0.99 | 0.97 | 0.97 | 0.99 |
| Chiz | 0.99 | 0.99 | 0.94 | 0.99 | 0.98 |
| Entropy | 0.98 | 0.98 | 0.96 | 0.95 | 0.98 |

**Table 6.** Considering cmp-functions and classification algorithms in Genuine-Traditional Spambot comparison

| Genuine-Traditional Spambot | RF | AdaBoost | SVM | KNN | Bayes |
|---|---|---|---|---|---|
| KS | 0.98 | 0.98 | 0.97 | 0.95 | 0.99 |
| Chiz | 0.97 | 0.97 | 0.89 | 0.90 | 0.97 |
| Entropy | 0.97 | 0.97 | 0.96 | 0.95 | 0.98 |

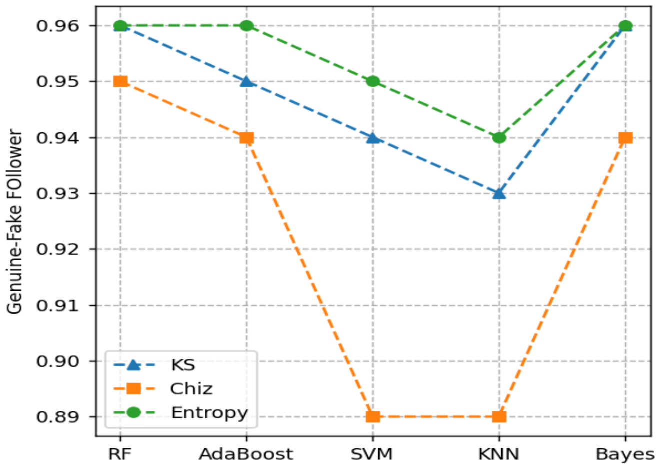

**Figure 2.** Comparison of different classifiers and cmp in Genuine-Fake Follower

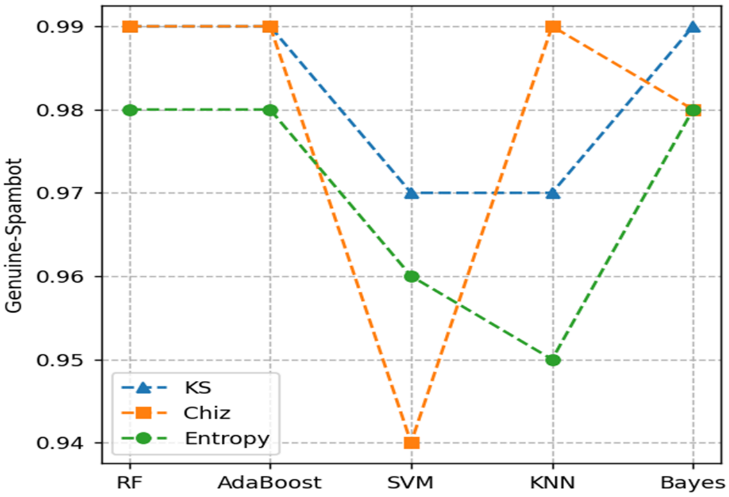

**Figure 3.** Different classifiers and cmp in Genuine-Spambot comparison

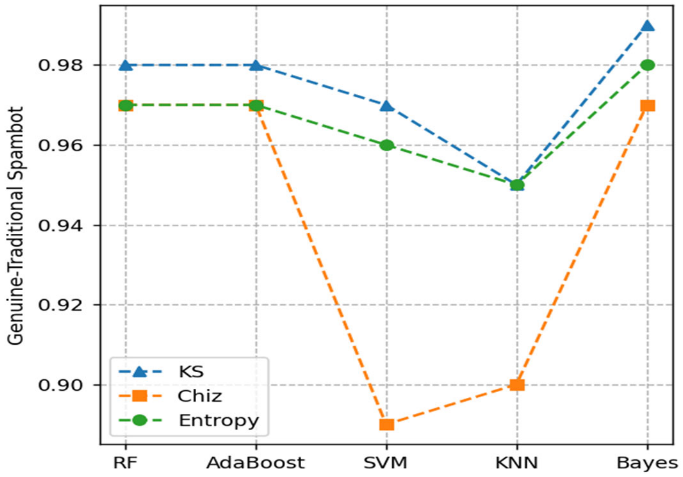

**Figure 4.** Comparison of different classifiers and cmp in Genuine-Traditional Spambot

**Table 7.** Comparison of different algorithms on different datasets

| Dataest | Botomer | Standard | BFPA |
|---|---|---|---|
| Genuine-Fake Follower | 0.87 | 0.96 | 0.97 |
| Genuine-Spambot | 0.98 | 0.99 | 0.99 |
| Genuine-Traditional Spambot | 0.90 | 0.98 | 0.98 |

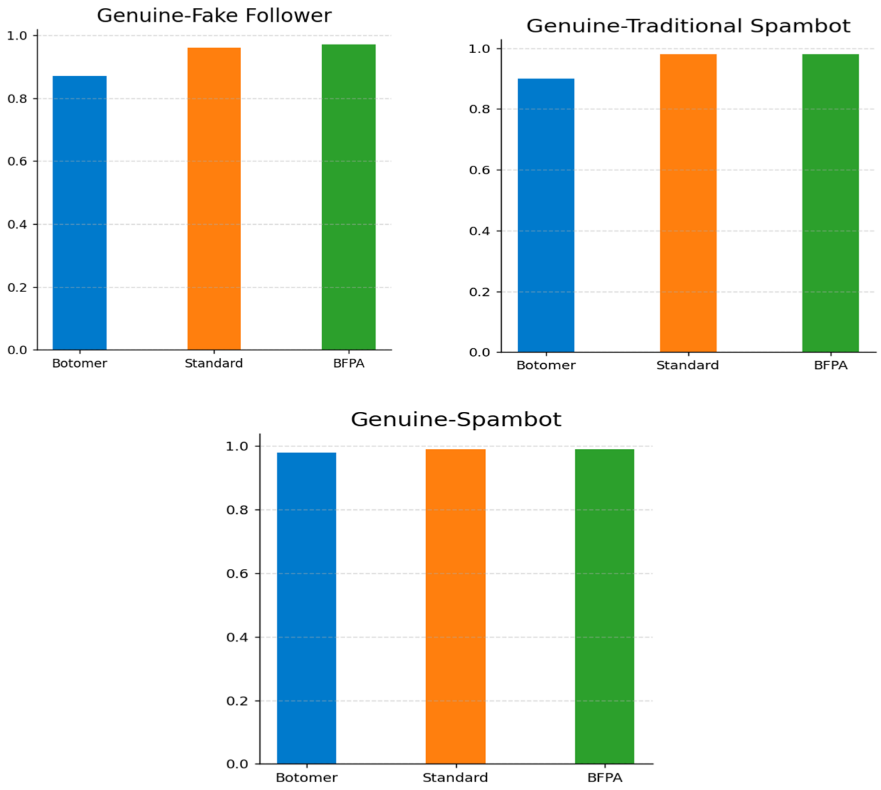

**Figure 5.** Comparison of different algorithms on different datasets

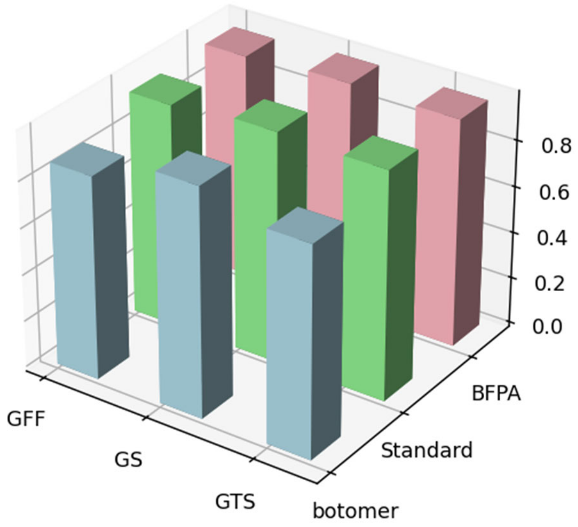

**Figure 6.** Comparison of f-measure results for Botometer, independent methods and BFPA on each data set

In order to test the stability of the detection performance of the algorithm in this paper, the effect of the maximum number of followers on the performance was discussed, and the data set was filtered from 50 to 400, as can be seen in Figure 7, which means that the algorithm is not too sensitive to the change of the maximum number of followers, and the detection performance is relatively stable regardless of the maximum number of followers, which proves the algorithm's robustness and reliability.

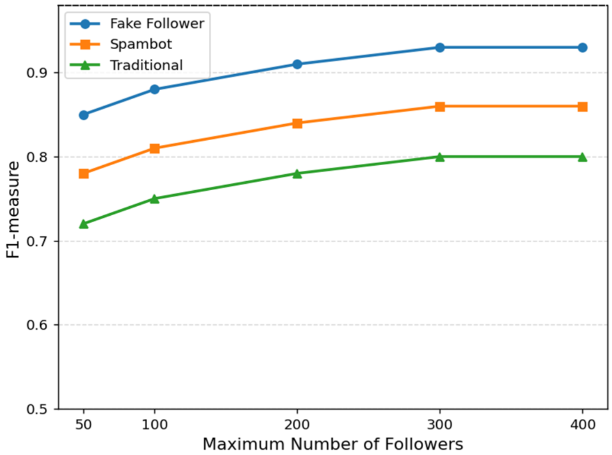

**Figure 7.** Effect of the maximum number of fans constraint on the f-measurement

The strength of BFPA lies in its ability to utilise Bayes' theorem to deal with uncertainty and probabilistic inference, thus making more accurate predictions with limited observational data. The combined consideration of friendship preference features and content features also enables a more comprehensive depiction of the behavioural patterns of social robots and improves the accuracy of detection. However, there are some limitations to this approach, such as model interpretability: Bayesian models, especially complex ones, are often difficult to interpret, and the black-box nature of the model may lead to difficulties when model decisions need to be interpreted, which can be a problem for application scenarios that require transparency and interpretability. To overcome this limitation, we need to explore new approaches to improve the interpretability and dynamic adaptability of models, as well as to enhance the defence against adversarial attacks.

## 5. CONCLUSION

In this paper, we have conducted in-depth experiments on several real datasets to evaluate the effectiveness of BFPA under different implementation parameters. In summary, these results show that the ability of the detection method developed techniques incorporating friendship preference features to distinguish between fake and legitimate accounts is outstanding, as expected, and several experimental studies have demonstrated the excellent capability of this paper's method in the face of large self-networks and scalability challenges. The work in this paper has shown that the combination of friendship preference features with content features, along with, the use of Bayesian and Genetic-based threshold optimisation algorithm classification with simulated annealing algorithm, has great potential for social bot detection. Both machine accounts and their corresponding detection techniques are constantly evolving, as in an arms race, where the two sides are at loggerheads but also progressing together, so the fight against social bots will be a long-term endeavour.

## REFERENCES

[1] Cresci S, Lillo F, Regoli D, et al. Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter[J]. ACM Transactions on the Web (TWEB), 2019, 13(2): 1-27.

[2] Cheng C, Luo Y, Yu C. Dynamic mechanism of social bots interfering with public opinion in network[J]. Physica A: statistical mechanics and its applications, 2020, 551: 124163.

[3] Shi W, Liu D, Yang J, et al. Social bots' sentiment engagement in health emergencies: A topic-based analysis of the COVID-19 pandemic discussions on Twitter[J]. International Journal of Environmental Research and Public Health, 2020, 17(22): 8701.1

[4] Pastor-Galindo J, Zago M, Nespoli P, et al. Spotting political social bots in Twitter: A use case of the 2019 Spanish general election[J]. IEEE Transactions on Network and Service Management, 2020, 17(4): 2156-2170.

[5] Ferrara E, Varol O, Davis C, et al. The rise of social bots[J]. Communications of the ACM, 2016, 59(7): 96-104.

[6] Shao C, Ciampaglia G L, Varol O, et al. The spread of low-credibility content by social bots[J]. Nature communications, 2018, 9(1): 1-9.

[7] Wang G, Wang Y, Liu K, et al. Multidimensional influencing factors of public opinion information dissemination in social media: Evidence from Weibo dataset[J]. International Journal of Modern Physics B, 2019, 33(31): 1950375.

[8] Schäfer F, Evert S, Heinrich P. Japan's 2014 general election: Political bots, right-wing internet activism, and prime minister Shinzō Abe's hidden nationalist agenda[J]. Big data, 2017, 5(4): 294-309.

[9] Cresci S, Di Pietro R, Petrocchi M, et al. Social fingerprinting: detection of spambot groups through DNA-inspired behavioral modeling[J]. IEEE Transactions on Dependable and Secure Computing, 2017, 15(4): 561-576.

[10] Rong Liu, Bo Chen, Ling Yu, et al. Research on malicious social bots detection techniques[J]. Journal of Communication, 2017, 38(Z2): 197-210.

[11] Lingyu Xu, Yixin Zhong, Zhicheng Chen. A study on the influencing factors of social robots on social opinion [J]. Journal of Intelligent Systems,2024: 1-10.

[12] Yanmei Zhang, Yingying Huang, Shijie Gan, et al. Research on microblogging network sailor identification algorithm based on Bayesian model[J]. Journal of Communication, 2017, 38(01): 44-53.

[13] Varol O, Ferrara E, Davis C, et al. Online human-bot interactions: Detection, estimation, and characterization[C]//Proceedings of the international AAAI conference on web and social media. 2017, 11(1): 280-289.

[14] Cresci S, Lillo F, Regoli D, et al. Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter[J]. ACM Transactions on the Web (TWEB), 2019, 13(2): 1-27.

[15] Moghaddam S H, Abbaspour M. Friendship preference: Scalable and robust category of features for social bot detection[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 20(2): 1516-1528.

[16] Cresci S, Petrocchi M, Spognardi A, et al. Better safe than sorry: an adversarial approach to improve social bot detection[C]//Proceedings of the 10th ACM Conference on Web Science. 2019: 47-56.

[17] Latah M. Detection of malicious social bots: A survey and a refined taxonomy[J]. Expert Systems with Applications, 2020, 151: 113383.

[18] Yang K C, Varol O, Hui P M, et al. Scalable and generalizable social bot detection through data selection[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(01): 1096-1103.

[19] Zeng Z, Li T, Sun J, et al. Research on the generalization of social bot detection from two dimensions: Feature extraction and detection approaches[J]. Data Technologies and Applications, 2023, 57(2): 177-198.

[20] Feng S, Wan H, Wang N, et al. Satar: A self-supervised approach to twitter account representation learning and its application in bot detection[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 3808-3817.

[21] Peng H, Zhang Y, Sun H, et al. Domain-Aware Federated Social Bot Detection with Multi-Relational Graph Neural Networks[C]//2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022: 1-8.

[22] Cai Z, Jermaine C. The latent community model for detecting sybil attacks in social networks[C]//Proc. NDSS. 2012.

[23] Yang K C, Varol O, Davis C A, et al. Arming the public with artificial intelligence to counter social bots[J]. Human Behavior and Emerging Technologies, 2019, 1(1): 48-61.

[24] Fazil M, Abulaish M. A hybrid approach for detecting automated spammers in twitter[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(11): 2707-2719.

[25] Al-Qurishi M, Alrubaian M, Rahman S M M, et al. A prediction system of Sybil attack in social network using deep-regression model[J]. Future Generation Computer Systems, 2018, 87: 743-753.

[26] Arin E, Kutlu M. Deep learning based social bot detection on twitter[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 1763-1772.

[27] Minnich A, Chavoshi N, Koutra D, et al. BotWalk: Efficient adaptive exploration of Twitter bot networks[C]//Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017. 2017: 467-474.

[28] Choi J, Jeon C. Cost-based heterogeneous learning framework for real-time spam detection in social networks with expert decisions[J]. IEEE Access, 2021, 9: 103573-103587.

[29] Cresci S, Di Pietro R, Petrocchi M, et al. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race[C]//Proceedings of the 26th international conference on world wide web companion. 2017: 963-972.

[30] Bharathi P S, Shibi S R. Reinforcement Learning with URL Features in Twitter Network to Detect Malicious Social Bots using Random Forest in Comparison with Convolutional Neural Network to Improve Accuracy[C]//2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM). IEEE, 2023: 1-5.