Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 6974809, 11 pages
https://doi.org/10.1155/2019/6974809

Research Article

A Multiuser Identification Algorithm Based on Internet of Things

Kaikai Deng, Ling Xing, Mingchuan Zhang, Honghai Wu, and Ping Xie

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

Correspondence should be addressed to Ling Xing; xingling_my@163.com

Received 18 January 2019; Revised 2 April 2019; Accepted 11 April 2019; Published 2 May 2019

Guest Editor: Vishal Sharma

With the rapid development of the Internet of Things (IoT) in 4G/5G deployments, the massive amount of network data generated by users has exploded, which has not only brought a revolution to human's living, but also caused some malicious actors to utilize these data to attack the privacy of ordinary users. Therefore, it is crucial to identify the entity users behind multiple virtual accounts. Due to the low precision of user identification in the many-to-many mechanism of user identification, a random forest confirmation algorithm based on stable marriage matching (RFCA-SMM) is proposed in this study. It consists of three key steps: we first employ the stable marriage matching model to calculate the similarity between multiple users and utilize a scoring model to calculate the overall similarity of the users, after which candidate matching pairs are selected; second, we construct the random forest model that exploits a user similarity vector training set; afterward, the candidate matching pairs combine the secondary confirmation of the random forest model, which both improve the precision of the many-to-many user identification and protect private user data in the IoT. Extensive experiments are provided to demonstrate that the proposed algorithm improves precision rate, recall rate, and F-Measure (F1), as well as Area Under Curve (AUC).

1. Introduction

As 4G/5G technology continues to evolve, it provides people with more efficient performance and increased speed in order to meet higher standards of data services for more users. At the same time, the higher speed and more reliable transmission of mobile communication further promote the development of the Internet of Things (IoT) in the large-scale era. The amount of user data is constantly increasing in the IoT context. Leveraging user data to analyze the social behavior of users can enable the provision of a safer social environment. According to statistics, 42% of users use multiple social networks simultaneously [1]. The IoT integrates different social methods to meet people's different needs to the greatest extent possible [2]. For instance, RenRen and Sina Microblog are services used to share personal statuses and publish blogs anytime and anywhere in China. However, as there is no direct link between user data on these services, a complete social network map is difficult to obtain. Multiuser identification is therefore employed to identify users of multiple virtual accounts [3, 4], allowing user data to be better protected in the IoT era.

User identification is also referred to as user matching. Many studies have addressed the user identification problem by examining user profile information attributes, primarily the user's personal information and published content, which includes username, geographical location, blog posts, etc. [5–13]. Although missing data is an issue in the process of filling in these attributes, they can still be filled in through the use of appropriate methods. Moreover, some attributes play an extremely important role in the process of user identification. Therefore, user identification based on user attribute information can better accomplish the work of identification. Some of the research on this topic focuses on the use of network topology for user identification. This research mainly relies on the user's circle of friends to identify a specific user [14–19]. The similarity between user accounts is judged by analyzing nodes between users. However, due to the heterogeneity of the network structure in practical applications, this method requires improvement in terms of its precision.

This study is divided into three sections: user profile data, user-generated data, and user-associated data, according to the type of IoT data involved. User profile data refers to the data that the user needs to enter or select when they register for their accounts. User-generated data refers to the data that is the user's published content. User-associated data refers to the data that users associate with other users. Among the numerous user attributes, some attributes are more important to the task of user identification, such as web links, blog posts, etc. Conversely, some attributes such as gender, age, etc. are less important in this context. Therefore, reasonable allocation of corresponding weights to users can also improve the precision of user identification to a certain extent.

A stable marriage matching algorithm is mainly used to find and solve a stable matching pair problem and has been widely used in the fields of economics and computing for this purpose [20]. Stable matching refers to cases in which the two identified items are mutually optimal choices rather than the first choice in the current match and where there is no better match to any one element in the unmatched elements. Meng Bo proposed that the ranking-based cross-matching algorithm could also adopt the concept used in the stable marriage matching algorithm. The algorithm uses profile attribute similarity (PAS) and user surrounding score (USS) to select candidate matching users and then uses the user matching score (UMS) to determine the users that match with the candidate users. In order to further improve the matching precision, a cross-matching process inspired by the stable marriage matching algorithm is added. Finally, the matching user pairs are obtained by a simple pruning process. However, the idea behind the stable marriage matching algorithm is that all users should match; thus, it is difficult to guarantee the precision of the final result. The existing user identification algorithm, which is based on supervised learning, is guaranteed in accuracy, but can only achieve one-to-one user matching.

Accordingly, a random forest confirmation algorithm based on stable marriage matching (RFCA-SMM) is proposed in this study. The proposed algorithm combines stable marriage matching and a scoring model to obtain the overall similarity of users in addition to candidate matching pairs. The similarity calculation of user attribute data is used to construct the user similarity vector, while the training set of the user similarity vector is leveraged to generate the random forest model. The final matching results of candidate matching pairs can be obtained by means of random forest confirmation.

2. Related Works

Multiuser identification technology is significant in both research and practice in many important fields. Current studies of multiuser identification can be divided into three categories according to the way feature information is used: user identification based on user profile information, network structure, and user-generated information.

2.1. User Profile Information-Based User Identification. Research based on user profile information for the purpose of solving user identification problems primarily focuses on personal information. The classification model is constructed, after which the corresponding matching strategy is used to complete user identification. Raad et al. [20] proposed the Friend of a Friend (FOAF) attribute matching strategy. Ye et al. [21] proposed an objective weighting method for user attributes to integrate user attribute information and complete the calculation of user profile similarity. Cortis et al. [8] proposed an identity recognition algorithm that assigns weights to individual attributes of user profile information and then calculates the similarity among attributes with reference to both grammatical and semantic aspects. Able et al. [22] aggregated user profile information in order to match users. Zamani et al. [23] took the user's unit, interests, and other attribute information into full consideration, integrating the similarity of multiuser attributes via the equal evaluation model and complex mixed training model; this improves the possibility of correctly identifying users owing to the personalized characteristics of many users' attribute information. Therefore, leveraging user attribute information to achieve user identification is a good choice.

2.2. Network Structure-Based User Identification. Network structure-based studies on user identification mainly focus on recognizing identical users by examining the user's circle of friends. The user's friend relationships are easy to obtain, the problem of malicious imitation and forgery is less likely to occur, and the importance of information coupling of local topology on network development has been certified. Narayanan et al. [14] proved for the first time that user identification can be accomplished by relying on the topology structure of the network; however, the precision of the matching results required improvement. Bartunov et al. [15] proposed the construction of the objective function by combining attribute information and network structure information and then optimized the function to obtain the optimal matching pair. Cui et al. [16] integrated user profile information similarity and graph similarity to achieve mapping from an email network to a Facebook network. Liu et al. [17] proposed the HYDRA approach to modeling user behavior by employing user attributes and user-generated content. Korula et al. [18] abstracted the problem of user identification into a mathematical definition, arguing that different social networks are generated by user graph structure through probability and that the selection process of graph edges is one of approximate probabilities. Tan et al. [19] modeled users' social relations and mapped users to low-dimensional space to improve the efficiency of user identification in the network. However, there is heterogeneity between nodes in the actual network structure, and the influence of this heterogeneity is ignored in the calculation process; therefore, the precision of this method in the context of user identification requires improvement.

2.3. User-Generated Information-Based User Identification. User identification based on user-generated information

mainly relies on the content published by users. Now that the Internet of Things (IoT) has a close relationship with our daily lives, people can immediately post their own dynamic content and comment on the content posted by the friends around them. As user behavior habits are not easy to change and hide [24], these habits can correspond strongly with the characteristics of the users themselves. Therefore, the use of data mining algorithms to discover these hidden association rules [25] can greatly improve the recognition rate of user identification. Goga et al. [10] used the geographic locations, timestamps, and writing habits of users' published statuses for user identification purposes. Li et al. [13] proposed a supervised machine learning algorithm to solve the user identity recognition problem based on user-generated content. In recent years, the development of mobile communication technology has made a great contribution to the incorporation of geo-tagging when users publish their statuses. As the user's track of action is not easy to imitate, the application of geographical location attributes to user identification opens up a new method of identifying users. Cao et al. [26] proposed an identification method for processing multisource data by utilizing the cooccurrence frequency of two user trajectories. Hao et al. [27] proposed that user trajectories are transformed into sequences composed of multiple grids, which are in turn transformed into vectors by using a TD-IDF model, after which the similarity of user trajectories is calculated via cosine similarity. Han et al. [28] proposed that each geographic coordinate point should be represented as a corresponding semantic position. The user's trajectory can thus be represented by the text composed of the semantic position, with the LDA model being used to represent the user's topic distribution; finally, the similarity of the user trajectories is calculated to determine whether the two users are the same. Therefore, the analysis of user behavior information for multiuser identification is ideal in this context.

3. Data Preprocessing

3.1. Filling Missing User Data. Data filling is commonly applied to user profile data processing. When a user registers an account, data may be missing for various reasons such as, e.g., privacy protection. Therefore, an appropriate filling method should be adopted for each different type of data from each dimension, as follows:

(1) Similarity filling: filling in user data by utilizing the relational degree [29] between other users and users with missing data. For example, user A and user B are friends, and they will generate social behaviors such as comments, reposts, and thumb up on social networks, where the comments indicate that the content posted by the friend on the social network is explained, reposts indicate that friends have similar interests, and thumb up indicates that they agree with the content posted by friends. The relational degree between users is calculated through the behaviors of "comment C_{AB}", "repost R_{AB}", and "thumb up L_{AB}" between users. The three types of user behavioral information are sequentially assigned the weights "3", "2", and "1". Select n users with the highest relational degree and take the mode number for filling. If the user with missing data has a low relational degree with other users, the data will not be filled. The relevant formula is as follows:

$$R_{AB} = 3\sum C_{AB} + 2\sum R_{AB} + \sum L_{AB} \tag{1}$$

(2) Speculated filling: the missing data is inferred from other attributes. This method is mainly used for user gender filling. The blog posts published by the user best reflect the characteristics of the user's personality; thus, by using the user's blog post information, the Bayesian classification model [30] can be employed to accomplish user gender speculation.

The Bayesian classification model is constructed as follows:

$$p(m \mid w_i) = \frac{p(w_i \mid m)\, p(m)}{p(w_i)} \tag{2}$$

where $p(m \mid w_i)$ denotes the probability that the user is identified as male when the word w_i occurs, $p(w_i \mid m)$ denotes the probability of the word w_i occurring in all males, $p(w_i)$ denotes the probability of the user being male, and $p(m)$ denotes the probability of the occurrence of word w_i.

Given the complexity of calculating $p(w_i \mid m)$, this article calculates the conditional independence naïve hypothesis. The formula is as follows:

$$\begin{aligned} p(w_i \mid m) &= \prod_{i=1}^{n} p(w_i \mid m) \\ &= p(w_1 \mid m)\, p(w_2 \mid m) \cdots p(w_n \mid m) \end{aligned} \tag{3}$$

$$p(w_i) = \prod_{i=1}^{n} p(w_i) = p(w_1)\, p(w_2) \cdots p(w_n) \tag{4}$$

Therefore, the Bayesian classification model constructed is as follows:

$$\begin{aligned} &p(m \mid w_i) \\ &= \frac{p(w_1 \mid m)\, p(w_2 \mid m) \cdots p(w_n \mid m)\, p(m)}{p(w_1)\, p(w_2) \cdots p(w_n)} \end{aligned} \tag{5}$$

The statistical results of the training set can be used to derive the probability required for the calculation in the Bayesian classification model. The prediction results of the corresponding attributes can be obtained via this model.

3.2. User Data Similarity Calculation. In view of the problem that the user data in the IoT has a different format, the user data format needs to be generalized before the similarity between each attribute in this study can be calculated; this processed data is more suitable for similarity calculation. The relevant calculation methods are as follows:

(1) Dice coefficient [31]: when calculating strings, they can be divided into two categories. When calculating the multivalued strings n_i and n_j, the sum of the two times of the intersection information and divided by the sum of

the elements of n_i and n_j yields the two strings of Dice coefficients, which are calculated as follows:

$$Simfunc\left(n_i, n_j\right) = 2\frac{\left|n_i \cap n_j\right|}{\left|n_i\right| + \left|n_j\right|} \tag{6}$$

Example. In two multivalued attribute strings "vivid music movie" and "movie travel", the intersection information is {"movie"}, so the similarity is 2/5=0. 4.

For single-valued attribute strings, the Dice coefficient is calculated as above, except that the intersection information is different.

Example. For the single-valued strings "johe" and "joh", the intersection information is "jo, oh", so the similarity is 4/5 = 0.8.

(2) Levenshtein distance [32]: the number of character edit steps required to calculate the equality of two strings is used as an operational cost to measure the difference between strings. The formulae for calculating the similarity of strings n_i and n_j are as follows:

$$Simfunc\left(n_i, n_j\right) = 1 - \frac{d\left(n_i, n_j\right)}{\max\left(\left|n_i\right|, \left|n_j\right|\right)} \tag{7}$$

where $d(n_i, n_j)$ denotes the distance between the strings n_i and n_j and $\max(|n_i|, |n_j|)$ denotes the maximum value of characters contained in the strings n_i and n_j.

(3) Cosine similarity [33]: this is mainly used to calculate the vector composed of user attributes. Assuming that A and B are two n-dimensional vectors, such that A is $[A_1, A_2, ..., A_n]$ and B is $[B_1, B_2, ..., B_n]$, then the cosine value of angle θ between A and B is the similarity value between vectors. The formula is as follows:

$$\cos\theta = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \times \sqrt{\sum_{i=1}^{n} {B_i}^2}} \tag{8}$$

The closer the angle between two vectors is to 1, the higher the similarity between two users is. The closer the angle between the two vectors is to 0, the smaller the cosine value of the included angle is and the lower the user similarity is. By comparing the size of cosine values, it can be determined whether the two accounts are identical.

(4) Term frequency-inverse document frequency (TF-IDF) [34]: this is mainly used to measure the importance of a certain word in the document and is often used to deal with multiword attribute fields such as personal profiles. The specific steps are as follows.

Step 1. Calculate the term frequency (TF) of each word in the document;

$$TF = \frac{n}{N} \tag{9}$$

where n denotes the number of occurrences of a certain word and N denotes the total number of words in the document.

Step 2. Calculate the inverse document frequency (IDF) of each word in the document;

$$IDF = \log\left(\frac{D}{P+1}\right) \tag{10}$$

where D denotes the total number of documents in the corpus, P denotes the number of documents containing a word in the document, and 1 is added to avoid cases in which the denominator is 0.

Step 3. Calculate the TF-IDF of each word in the document;

$$TF - IDF = TF \times IDF = \frac{n}{N} \times \log\left(\frac{D}{P+1}\right) \tag{11}$$

Step 4. Select keywords in each document to construct a term frequency vector for calculating similarity.

Step 5. Calculate the similarity value by cosine similarity.

(5) User blog data similarity calculation: frequent item sets of user blog data are mined by means of frequent pattern mining to calculate user similarity. Due to the difference in the amounts of user-published content, the one-item set is also used as a calculation indicator in this study. "1" is added to avoid a high-frequency item set in the calculation of similarity. The formula is as follows:

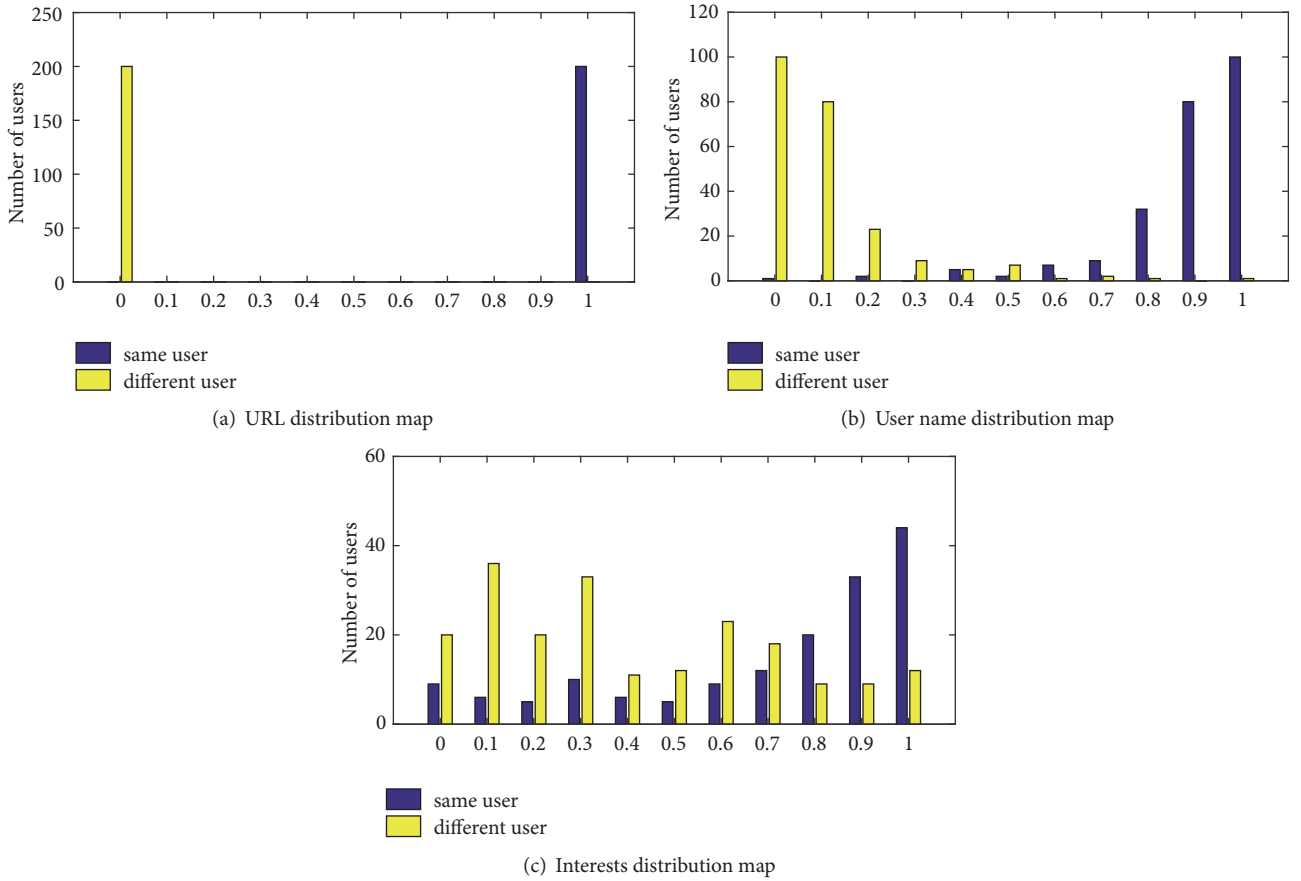
$$S_{AB} = \sum_{E_i \in A \cap B} \left(\left(1 + CA_{E_i}\right) \times \left(1 + CB_{E_i}\right)\right)^{C_{E_i}} \tag{12}$$

where CA_{E_i} denotes the support degree count of the frequent item set E_i of user A, CB_{E_i} denotes the support degree count of the frequent item set E_i of user B, and C_{E_i} denotes the item set number of E_i. The similarity threshold is set at 5,000 based on historical data. If $S_{AB} > 5000$, return "1"; otherwise, return "0".

(6) State timestamp similarity calculation: the time points of users' publishing status also have certain personalized characteristics. The average dynamic number can be obtained by dividing the dynamic number generated by users in a certain period of time by the total dynamic number. The average dynamic number is then used to form a user timestamp vector of 24 dimensions. The similarity is calculated; users are determined to be the same user when Sim<0. 1 according to the statistical results. The formula is expressed as

$$Sim_t(a, b) = \sum_{i=1}^{24} \left|u_{ai} - u_{bi}\right| \tag{13}$$

where u_{ai}, u_{bi} denote the average dynamic number of the ith time period of users a and b.



(a) URL distribution map

(b) User name distribution map

(c) Interests distribution map

Figure 1: User attribute similarity distribution.

4. Multiuser Identification Method

4.1. Building User Similarity Vectors. Research and analysis of user data in the Internet of Things (IoT) context can assist in meeting people's network needs. However, some malicious users will employ the user data to attack normal users via the IoT. Therefore, it is necessary to identify and analyze IoT users.

In this study, the profile information and behavior information of user data are utilized to achieve multiuser identification. After filling in user profile information, the precision of user identification can be increased to a certain extent. User behavior information has the characteristic of being personalized, which allows for highlighting of the user's own behavior habits and is thereby conducive to the improvement of user identification precision. A reasonable similarity calculation method is used for the data of each dimension of the user. The data for each dimension is provided with a threshold value when calculating similarity. After comparing the calculated similarity of attributes with the set threshold, qualified results return "1"; otherwise, return "0". Thus, user similarity vectors composed of "0" and "1" can be formed and used for the input of subsequent algorithms.

4.2. Weight Analysis of User Attributes. By calculating the similarity between user attributes, the whole similarity vector of user attributes can be obtained. As different user attributes have different influences on the degree of user recognition, it is necessary to calculate the weight of the attribute items. Figure 1 shows the performance of a single attribute in user identification. It can be clearly seen from Figure 1 that the URL and user name have different distributions of similarity between the same user and different users when user matching is performed. As these attribute items are highly distinguishable, the weight allocation should be relatively large. When users match in terms of their interests, the similarity distribution between the users is small, meaning that the weight distribution should also be small. Again, as each attribute has different effects on user identification, it is therefore necessary to assign corresponding weights to each attribute.

4.3. Weight Allocation Algorithm. After the user data is preprocessed, multiple user attributes are determined. When determining the weighting coefficient of the similarity judgment of each attribute in the user data, the traditional expert subjective weighting method encounters the problem of poor robustness, while the objective weighting method relies too much on the existence of a large amount of sample data, which is poor in universality. Therefore, this article proposes the posterior probability-based information entropy weight allocation algorithm.

Input: Source network account user data information vector F_A, user data vector $\{F_j\}_{j=1}^{k}$ for all accounts in the target network, user data vector F_B to be matched account in the target network
Output: The final similarity $V_{final} = W_i(x)V(F_A, F_B)$ between the two accounts F_A and F_B
1: foreach F_j in $\{F_j\}_{j=1}^{k}$
2: for i=1 to n
3: Calculate the similarity $V(F_A, F_B) = (v_1^{AB}, v_2^{AB}, ..., v_n^{AB})$ of account A and B by using formula (6) (7) (8) (11) (12) (13)
4: end
5: for i=1 to n
6: The attribute weights of the user data are assigned using equation (15)
7: end
8: Calculate the final similarity $V_{final} = W_i(x)V(F_A, F_B)$ between the two accounts F_A and F_B
9: return V_{final}

Algorithm 1: User data similarity calculation.

In information theory, the entropy value reflects the degree of information disorder. The smaller its value is, the more orderly the information is, and the more valuable this attribute is; on the contrary, the more disordered the information is, the lower the value of this attribute is. Therefore, information entropy can be used to evaluate the effectiveness of the attributes used. According to the definition of information entropy, for any random variable, the formula is as follows:

$$E(x) = -\sum_{x \in X} P(x) \log P(x) \tag{14}$$

where $p(x)$ is the possible value probability for the attribute.

In order to make the probability description of attributes more precise, more effective weights are assigned to each attribute. On the basis of information entropy, the posterior probability of user attributes is further calculated, which aids in improving the precision of user identification. By combining the posterior probability and information entropy, the attribute weight of the user account is $W(x)$, such that

$$W(x) = -p(y_s \mid s) \sum_{x \in X} p(x) \log(p(x)) \tag{15}$$

where $p(y_s \mid s)$ is the posterior probability of the attribute.

The user data information contains n attribute items. The data information vector is $F_j = (a_1^j, a_2^j, ..., a_n^j)$, where a_i^j $(i = (1, 2, ..., n))$ represents the ith attribute information of account j. The user similarity vector is defined as $V(F_A, F_B) = (v_1^{AB}, v_2^{AB}, ..., v_n^{AB})$, where v_i^{AB} represents the similarity between the ith attribute of user F_A and user F_B's attribute information. If the similarity exceeds the threshold, output "1"; otherwise, output "0". Accordingly, the user similarity vector is a vector composed of "1" and "0". Therefore, the final user similarity vector is $V_{final} = W_i(x)V(F_A, F_B)$. The process is summarized in Algorithm 1.

4.4. Random Forest Confirmation Algorithm Based on Stable Marriage Matching

4.4.1. Similarity Score. In order to improve the efficiency of the similarity calculation between users, this study adopts a stable marriage matching algorithm to perform the many-to-many matching calculation. The overall similarity of users is evaluated by means of the similarity score of user matching. The relevant formula is as follows:

$$Score = \sum_{i=1}^{20} w_i v_i^{AB} \tag{16}$$

where Score denotes the final Score of the match, w_i denotes the weight of the ith attribute of the user, and v_i^{AB} denotes the similarity of the ith attribute of user A and user B. The higher the Score is, the more likely it is the same user.

4.4.2. Stable Marriage Matching Algorithm. The scoring formula can evaluate the overall similarity of two users based on user data information. The higher the score, the more likely it is that the two users are the same user. The stable marriage matching algorithm uses the similarity score between users to select candidate matching pairs. If we calculate user data for all accounts, then the computational complexity will be high. Therefore, it is necessary to obtain v_s by filtering the target account in another network according to the condition C: filter accounts by username. The specific steps involved are as follows.

Step 1. Each user on social network M and the user on social network N are matched by scoring formula.

Step 2. The user on M is matched with the user on N who ranks first according to the score. If the user on N has already matched other users, the user will compare the user who has already matched himself with the user who is requesting matching with himself. Finally, the user with the highest score will be selected as the other half of the matching pair.

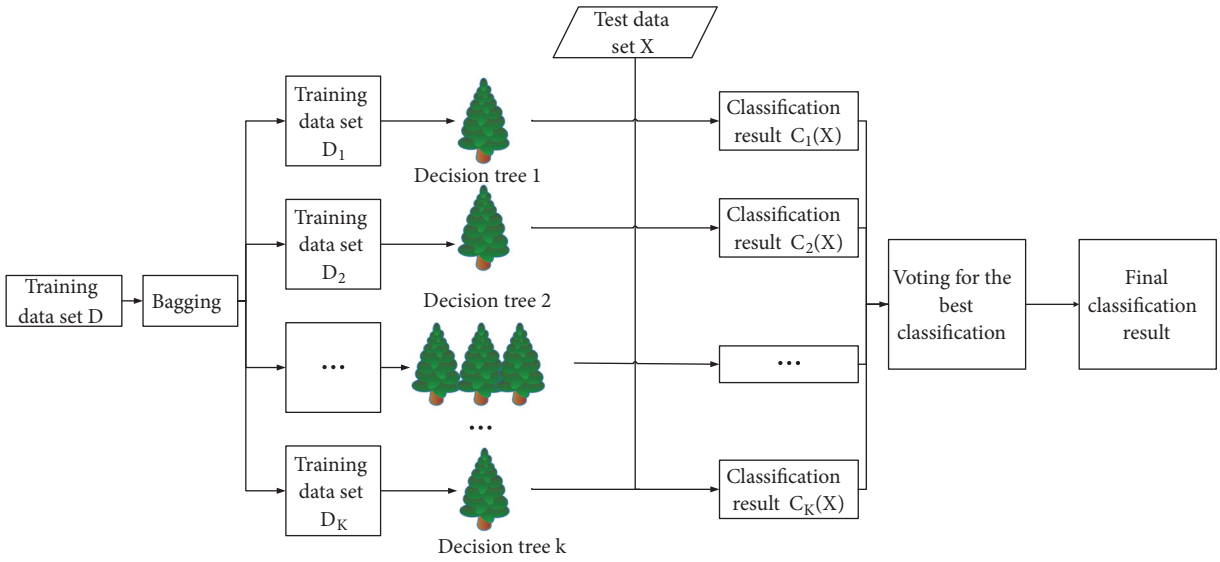


Figure 2: Random forest model construction process.

Step 3. After *Step 2* is complete, some users will still fail to be matched successfully. A user who does not match will be matched with the highest ranked user among all users who have not rejected themselves, after which *Step 2* will be repeated until all users match.

4.4.3. Random Forest Algorithm. Users through the stable marriage matching algorithm output is a matching pair. However, such results cannot be used directly, as it would be easy to obtain poor matches if this was the case. In order to solve this problem, a second confirmation of random forest is established in order to eliminate the negative influence of wrong matching results on the final results as far as possible.

There are many algorithms based on supervised learning, including logistic regression, SVM, Adaboost, etc. Random forest is selected as the final quadratic confirmation algorithm in this study for the following reasons.

(1) There are 20 data dimensions used in this study, among which there may be linear correlation dimensions. The dimension of linear correlation not only plays no positive role in the training of the supervised learning model, but also impacts the effect of other nonlinear correlation dimensions. In general machine learning model training, data dimensionality reduction will be processed, and data dimensionality reduction is a tedious process. However, random forests are not sensitive to multicollinearity, and the results are robust to missing and unbalanced data.

(2) While overfitting is always discussed in machine learning, it is not easy for the random forest model to produce the overfitting phenomenon owing to the randomness involved.

Figure 2 provides an overview of the construction process of the random forest validation model. The specific steps are as follows.

Step 1. Acquire the training set.

(a) The original input training data set D comprises 20 prediction attributes and a classification label Y. The 20 prediction attributes are the user similarity vector $V(F_A, F_B) = (v_1^{AB}, v_2^{AB}, ..., v_n^{AB})$ obtained in the above, while the classification label is $Y = (1 \parallel 0)$. A class label of "1" means that the users are the same, while "0" means that they are not the same.

(b) The original training data set D is sampled by random sampling with K playback times via the Bagging method, and a new training subset U of K is obtained.

Step 2. Generate the decision tree.

(a) The number of prediction attributes in the training sample is 20, and $F = \sqrt{20} \approx 5$ attributes are randomly selected from the 20 prediction attributes to form a random feature subspace X_i, which is the split attribute set of the current node of the decision tree. During the generation of the random forest model, F remains unchanged.

(b) According to the decision tree generation algorithm, each node is split by selecting the optimal split attribute from the random feature subspace X_i.

(c) Each tree grows completely without pruning. Finally, according to each training set D_i, the corresponding decision tree is generated as $h_i(D_i)$.

(d) Combine all the generated decision trees together to generate a random forest model $\{h_1(D_1), h_2(D_2), ..., h_i(D_i)\}$. Each test tree $h_i(D_i)$ is tested using the test set sample X to obtain a corresponding classification result $\{C_1(X), C_2(X), \ldots, C_K(X)\}$.

(e) Using the plurality voting method, according to the classification result output by the K-tree decision tree, the classification result with a large number of decision trees is used as the final classification result corresponding to sample X of the test set.

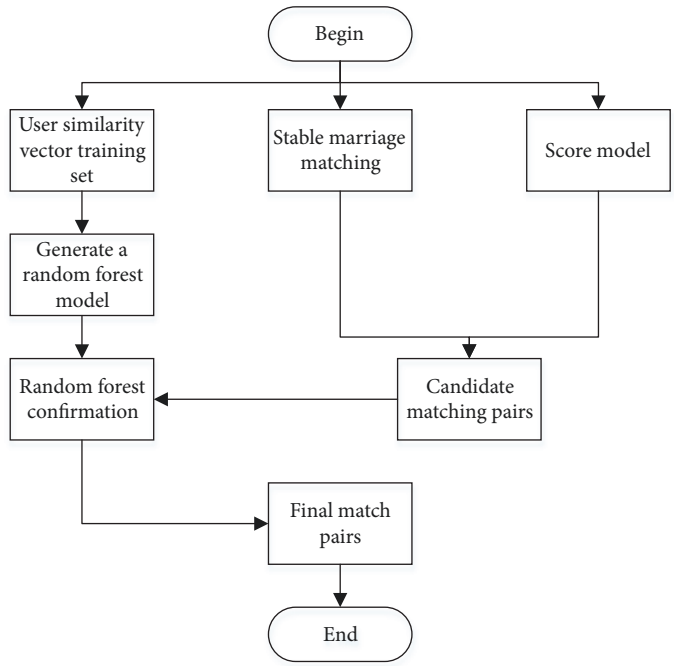


Figure 3: Random forest confirmation based on stable marriage matching.

Figure 3 describes the process of RFCA-SMM. The input data of the algorithm is the user attribute data in the IoT. By using the stable marriage matching algorithm in combination with the scoring formula, the overall similarity between users can be obtained in order to select the candidate matching pairs with the user similarity vector training set as input. The random forest model is constructed, and the candidate matches obtained are used as input data to confirm and identify in the random forest. If the identification result for the candidate matching pair is not the same user, the candidate matching pair is marked as "unmatched"; by contrast, if the candidate match pair contains two instances of the same user in the random forest confirmation, the final match result is generated. The algorithm flow of RFCA-SMM is represented by Algorithm 2.

Input: To be matched account v_s, Candidate matching account v_{match}, the set of accounts V_M that have not been matched in the social network M, the set of accounts V_N that have not been matched in the social network N
Output: The final match pairs set R
1: R=ϕ
2: Initialize unmatched queue
3: while $V_M \neq \phi$ and $V_N \neq \phi$ do
4: if v_s=NULL then
5: v_s=UserSelect(V_M,V_N)
6: v_{match}= UserMatch(v_s,V_M,V_N)
7: end if
8: (v_s,v_{match})= Secondary confirmation (v_s,v_{match},R)
9: end while
10: pruning process
11: return R

Algorithm 2: RFCA-SMM.

5. Analysis of Experimental Results

In order to verify the effectiveness of the proposed algorithm, [35] provides five open datasets of foreign mainstream social networks.

In this study, precision rate, recall rate, F-measure (F1), and AUC are used as evaluation criteria. The relevant formulae are as follows:

$$precision = \frac{tp}{tp + fp} \tag{17}$$

$$recall = \frac{tp}{tp + fn} \tag{18}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{19}$$

AUC: the area under the Receiver Operating Characteristic (ROC) curve is directly calculated. The ROC curve is defined as the X-axis by the False Positive Rate (FPR), while the True Positive Rate (TPR) is defined as the Y-axis. The formulae for these two values are as follows:

$$TPR = \frac{tp}{tp + fn} \tag{20}$$

$$FPR = \frac{fp}{fp + tn} \tag{21}$$

where tp denotes the number of the same users that are correctly matched, tn denotes the number of users that are

Table 1: Comparison of several types of supervised learning.

Algorithms	Precision	Recall	F1	AUC
Random Forest	0. 961	0. 881	0. 919	0. 961
SVM	0. 950	0. 860	0. 903	0. 945
Logistic	0. 935	0. 900	0. 917	0. 965
Adaboost	0. 910	0. 870	0. 890	0. 900

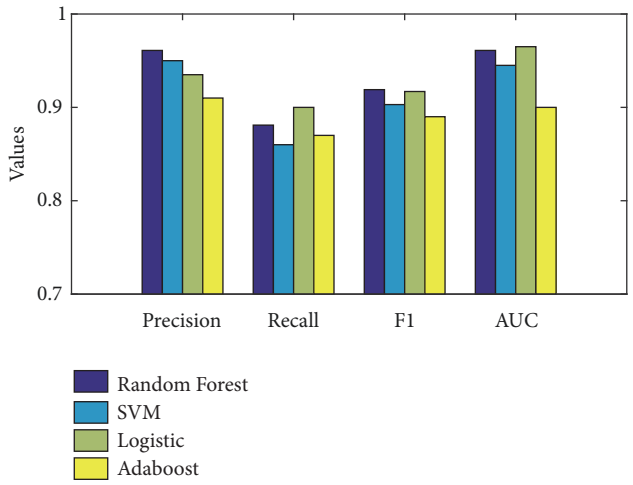


Figure 4: Comparison of several types of supervised learning.

unmatched and not the same, fp denotes the number of users that are matched but are not the same, and fn denotes the number of users that are not matched but are the same users.

5.1. Comparison of Random Forest Model and Other Supervised Learning Models. This article uses the random forest supervised learning model for the confirmation of candidate matching pairs. In order to verify the effectiveness of the proposed algorithm in obtaining the matching results, the random forest model and other supervised learning models are analyzed with reference to the evaluation indicators outlined above. The ratio of training set data to test set data is 3:1 and the number of users is 1000 pairs. The results are presented in Table 1 and Figure 4.

It can be seen from Figure 4 that these supervised learning algorithms have relatively good results; among them, the best performing algorithms are Random Forest and Logistic. Random Forest performs slightly better in precision rate and F1, while Logistic performs slightly better in recall rate and AUC. However, considering the modeling scenarios of these two supervised learning algorithms, Random Forest has an advantage over Logistic. Therefore, the effectiveness of the random forest model is also proven.

5.2. Comparison of RFCA-SMM and RCM Algorithm Results. This section presents a comparative analysis of the RFCA-SMM and Ranking-based cross-matching (RCM) algorithms. The purpose of the RCM algorithm is to accurately find more matching pairs, which decompose the seed user's identification into a step-by-step iterative process. In the iteration process of each step, the calculation process of the algorithm is divided into three substeps: account selection, account matching, and cross matching. Accumulate the results of each iteration to form a result set. The advantage of this algorithm is to compare the results obtained each time and select the user account with a high score as the final result. However, the precision of the RCM algorithm is largely affected by the number of seed users (that is, the number of users who are known to match pairs). If it is not possible to know in advance which accounts are the same user (that is, there is untagged identity match), then the RCM algorithm needs to be improved in terms of precision. Since the algorithm for user identification in this study is untagged, the experimental results of the two algorithms are analyzed using an unmarked data set, as shown in Table 2 and Figure 5.

Table 2: Comparison of RFCA-SMM and RCM results.

Algorithms	Precision	Recall	F1	AUC
RFCA-SMM	0. 962	0. 871	0. 914	0. 961
RCM	0. 934	0. 875	0. 904	0. 912

It is clear from Figure 5 that the proposed RFCA-SMM algorithm is superior to the RCM algorithm in terms of precision, F1, and AUC in this study, although its performance is slightly lower than that of the RCM algorithm in terms of recall rate. The reason is that the proposed algorithm performs user-generated data processing, user attribute weight distribution, and secondary confirmation based on supervised learning compared with the RCM algorithm. It can be seen from Figure 5 that the final user matching results of the proposed algorithm achieve some improvement in the evaluation index compared with the RCM algorithm, mainly because the proposed algorithm (unlike RCM) does not take the social network structure into account [36]. In summary, the algorithm proposed in this study improves the precision of user identification to a great extent.

6. Conclusions

The key features of 4G/5G technology, namely, low energy consumption and low delay, have laid the foundation for the development of the Internet of Things (IoT). Given the various other advantages of this technology, it can effectively promote the rapid development of the IoT industry chain. Since most of the information in the IoT is related to private user data, we propose a random forest confirmation algorithm based on stable marriage matching (RFCA-SMM). The candidate matching pairs are obtained by combining the stable marriage matching algorithm with the scoring model. In order to demonstrate the effectiveness of the random forest model, we analyze the random forest model and several other supervised learning models on the evaluation indicators and finally the second confirmation of candidate matching pairs via the random forest. Moreover, we conduct a comparative analysis of RFCA-SMM and RCM. The experimental results

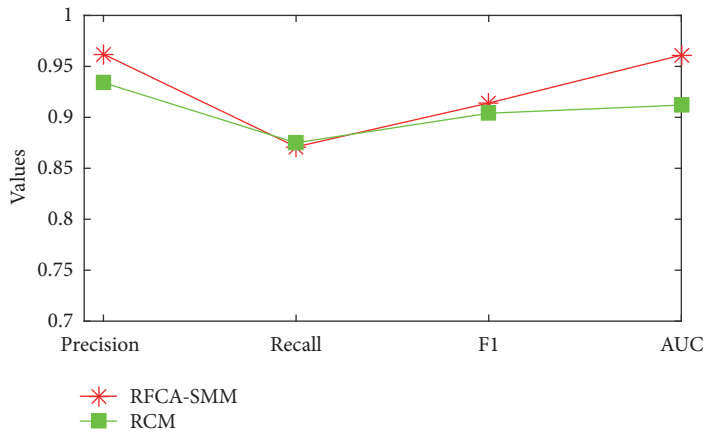


Figure 5: Comparison of RFCA-SMM and RCM results.

show that the proposed algorithm can provide excellent performance with precision rate, F1, and AUC reaching 96.2%, 91.4%, and 96.1%.

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 there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant no. 61771185, Grant no. 61772175, and Grant no. 61801171), Science and Technology Research Project of Henan Province (Grant no. 182102210044 and Grant no. 182102210285), Key Scientific Research Program of Henan Higher Education (Grant no. 18A510009), and Postdoctoral Science Foundation of China under Grant 2018M632772.

References

[1] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "COSNET: connecting heterogeneous social networks with local and global consistency," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1485–1494, 2015.

[2] X. Zhou, X. Liang, H. Zhang, and Y. Ma, "Cross-platform identification of anonymous identical users in multiple social media networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 411–424, 2016.

[3] M. M. Mostafa, "More than words: social networks' text mining for consumer brand sentiments," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4241–4251, 2013.

[4] T. Tuna, E. Akbas, A. Aksoy et al., "User characterization for online social networks," *Social Network Analysis and Mining*, vol. 6, no. 1, p. 104, 2016.

[5] J. Liu, F. Zhang, X. Song, Y. Song, C. Lin, and H. Hon, "What's in a name?: an unsupervised approach to link users across communities," in *Proceedings of the the Sixth ACM International Conference*, pp. 495–504, Rome, Italy, 2013.

[6] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 41–49, USA, 2013.

[7] O. Goga, D. Perito, H. Lei, R. Teixeira, and R. Sommer, "Large-scale correlation of accounts across social networks," Tech-nical report, 2013.

[8] K. Cortis, S. Scerri, I. Rivera, and S. Handschuh, "An ontology-based technique for online profile resolution," in *Social Informatics*, Lecture Notes in Computer Science, pp. 284–298, Springer International Publishing, Berlin, Germany, 2013.

[9] P. Jain, P. Kumaraguru, and A. Joshi, "@ i seek "fb. me": Identifying users across multiple online social networks," in *Proceedings of the the 22nd International Conference on World Wide Web Companion*, pp. 1259–1268, Rio de Janeiro, Brazil, 2013.

[10] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, "Exploiting innocuous activity for correlating users across sites," in *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pp. 447–457, 2013.

[11] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proceedings of the the 22nd ACM international conference (CIKM)*, pp. 179–188, San Francisco, Calif, USA, 2013.

[12] J. Haupt, B. Bender, B. Fabian, and S. Lessmann, "Robust identification of email tracking: a machine learning approach," *European Journal of Operational Research*, vol. 271, no. 1, pp. 341–356, 2018.

[13] Y. Li, Z. Zhang, Y. Peng, H. Yin, and Q. Xu, "Matching user accounts based on user generated content across social networks," *Future Generation Computer Systems*, vol. 83, pp. 104–115, 2018.

[14] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pp. 173–187, IEEE, Berkeley, Calif, USA, 2009.

[15] S. Bartunov, A. Korshunov, S. Park, W. Ryu, and H. Lee, "Joint link-attribute user identity resolution in online social networks," in *Proceedings of the 6th SNA-KDD Workshop*, 2012.

[16] Y. Cui, J. Pei, G. Tang, W.-S. Luk, D. Jiang, and M. Hua, "Finding email correspondents in online social networks," *World Wide Web*, vol. 16, no. 2, pp. 195–218, 2013.

[17] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "HYDRA: large-scale social identity linkage via heterogeneous behavior modeling," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 51–62, USA, 2014.

[18] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," in *Proceedings of the VLDB Endowment*, vol. 7, pp. 377–388, 2014.

[19] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen, "Mapping users across networks by manifold alignment on hypergraph," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, vol. 14, pp. 159–165, 2014.

[20] H. Kobayashi and T. Matsui, "Successful manipulation in stable marriage model with complete preference lists," *IEICE Transaction on Information and Systems*, vol. 92, no. 2, pp. 116–119, 2009.

[21] E. Raad, R. Chbeir, and A. Dipanda, "User profile matching in social networks," in *Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS)*, pp. 297–304, 2010.

[22] Y. Na, Z. Yinliang, D. Lili, B. Genqing, E. Liu, and G. J. Clapworthy, "User identification based on multiple attribute decision making in social networks," *China Communications*, vol. 10, no. 12, pp. 37–49, 2013.

[23] F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause, "Cross-system user modeling and personalization on the social web," *User Modeling and User-Adapted Interaction*, vol. 23, no. 2-3, pp. 169–209, 2013.

[24] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao, "Unsupervised clickstream clustering for user behavior analysis," in *Proceedings of the the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 225–236, San Jose, Calif, USA, 2016.

[25] K. A. Alam, R. Ahmad, and K. Ko, "Enabling far-edge analytics: performance profiling of frequent pattern mining algorithms," *IEEE Access*, vol. 5, no. 99, pp. 8236–8249, 2017.

[26] W. Cao, Z. Wu, D. Wang, J. Li, and H. Wu, "Automatic user identification method across heterogeneous mobility data sources," in *Proceedings of the 32nd IEEE International Conference on Data Engineering (ICDE)*, pp. 978–989, IEEE, 2016.

[27] T. Hao, J. Zhou, Y. Cheng, L. Huang, and H. Wu, "User identification in cyber-physical space: a case study on mobile query logs and trajectories," in *Proceedings of the the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–4, Burlingame, Calif, USA, 2016.

[28] X. Han, L. Wang, S. Xu, G. Liu, and D. Zhao, "Linking social network accounts by modeling user spatiotemporal habits," in *Proceedings of the 15th IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 19–24, IEEE, 2017.

[29] X. L Li, Y. L. Han, D. Y. Zhang, and X. G. Xu, "An evaluation algorithm for importance of dynamic nodes in social networks based on three-dimensional grey relational degree," in *Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators*, pp. 201–212, Springer, Singapore, 2018.

[30] M. Almishari and G. Tsudik, "Exploring linkability of user reviews," in *Computer Security – ESORICS 2012*, vol. 7459 of *Lecture Notes in Computer Science*, pp. 307–324, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2012.

[31] G. Kondrak, D. Marcu, and K. Knight, "Cognates can improve statistical translation models," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 46–48, Edmonton, Canada, 2003.

[32] L. Yujian and L. Bo, "A normalized Levenshtein distance metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[33] H. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian Conference on Computer Vision*, vol. 6493 of *Lecture Notes in Computer Science*, pp. 709–720, Springer, Berlin, Germany, 2010.

[34] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 493–502, 2004.

[35] M. Yan, J. Sang, and C. Xu, "Unified youtube video recommendation via cross-network collaboration," in *Proceedings of the the 5th ACM on International Conference on Multimedia Retrieval*, pp. 19–26, New York, NY, USA, 2015.

[36] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–37, 2018.