Your Username Can Give You Away: Matching Turkish OSN Users with Usernames

Önder Çoban[1], Ali İnan[2], Selma Ayşe Özel[1]

[1]Çukurova University, Computer Engineering Department, Adana, Turkey.
[2]Adana Alparslan Türkeş Science and Technology University, Computer Engineering Department, Adana, Turkey.
e-mail: onder.cbn@gmail.com, ainan@atu.edu.tr, saozel@cu.edu.tr

Research Paper Received: 01.01.2021 Revised: 26.02.2021 Accepted: 26.02.2021

Abstract—User profile matching (i.e., user cross-referencing, user identification) aims to find accounts that belong to the same users over different websites or online social networks (OSNs). Solving this problem can be useful for many operations and functionalities such as friend recommendation and link prediction across different OSNs. Additionally, identifying users across different OSNs may enable an adversary to aggregate incomplete information of users. Hereby, an adversary can extract and use online footprint of users to violate their privacy and security via putting them into threats such as identity theft, online stalking, and blackmailing among many others. Usernames are indispensable elements of all websites that require user registration. Even though usernames are generally short strings, they potentially reflect users' characteristics and habits such as the political sense of belonging, hometown, and so on. In this study, we make an effort to match users of distinct OSNs relying only on their usernames. We use two different approaches based on machine learning and vector-based username similarity to build our learning function. We also explore different feature spaces from the literature and further investigate which approach produces better results. We conducted our experiments on a real-world username data set that is extracted from the OSN accounts of Turkish users we crawled in our previous work. Our results show that building learning function by binary classification outperforms the similarity approach and it achieves the best F-score of 0.921 without feature selection and extension.

Keywords—User profile matching, username similarity, online social networks, Facebook.

1. Introduction

Online social networks (OSNs) are so extremely popular communication tools that such services (e.g., Facebook, Twitter, Instagram, etc.) have hundreds of millions of users and they are even said to be able to reflect real-life [1]. OSNs allow users to connect and reveal their personal information depending on the chosen OSN's focus, services, and functionalities [2]. Today, people tend to create multiple OSN accounts for different purposes such as finding new friends, discussing opinions, playing online games, and so on [3]. For instance, users use LinkedIn for professional purposes. Facebook, on the other hand, is preferred by users who want to connect with their family

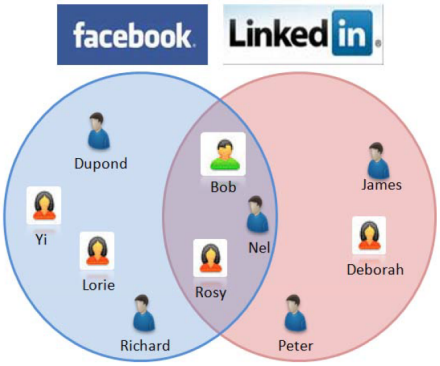


Fig. 1: Matching users across OSNs. Bob, Nel, and Rosy have accounts both in the Facebook and LinkedIn [10].

members and friends [4]–[6]. Each OSN has its solution for storing and displaying the information of its users [7] and users disclose different aspects of their lives on different OSNs [8] which is often incomplete [4], [9].

Matching users which is exemplified in Figure 1, therefore, can benefit inter-OSN operations and functionalities such as aggregating user information, friend recommendation, searching for a user and so on [7], [8], [10]. However, it can also be employed to violate the security and privacy of users since a successful identification may give way to collect user information from different OSNs. Additionally, users' personal information may be disclosed to advertisers, prospective matches, and data aggregators who use the data for malicious or unrelated purposes [11]. User identification can also be employed to collect public users' data from other OSN so as to compromise privacy by using the collected data as auxiliary information [12]. These privacy threats also bring together security risks. Primary causes of concern are identity theft, fraud, and (cyber) stalking [13].

All of these cases show that matching (or identifying) users across different OSNs is an important problem in the OSN research field. Therefore, identifying users across different sites is an attractive topic for researchers [6], [14], [15]. User profile matching, however, is not a straightforward task because there is no common identifier that allows linking accounts of the same individual across different OSNs. Profile names (i.e., display names) do not suffice because they may repeat. On the other hand, usernames are unique but only within the context of the same OSN. Also bear in mind that users are free to select usernames they want instead of their real identities and OSNs rarely link their users' accounts with other sites or services [9]. User identification studies that use public data can be grouped into three types based on the attributes that are used [6], [16]–[18].

These three types of techniques utilize (i) profile (e.g., display name, age, etc.), (ii) content (e.g., wall activities, etc.), and (iii) network structure (e.g., friends, followers, etc.) based attributes. There are many studies that employ profile attributes [4], [5], [7]–[10], [15], [19]–[21], content attributes [3], [14], [22]–[24], and network structure attributes [5], [8], [17]. However, user matching studies, which especially use content or network structure attributes, tend to fail as publicly shared data is often unavailable, incomplete, or unreliable due to the privacy settings or some other specific purposes [15].

On the other hand, a username may not always be available and be a numeric string that is automatically assigned by OSNs in some cases [4]. In such a case username does not have enough information to help user identification, which makes the user matching process quite fragile [15]. Leaving out such situations, nevertheless, screen names (i.e., display names) and usernames are the most discriminative information for user identifi-

cation [21] as they are fundamental elements of all websites and can reflect a user's characteristics and habits. For instance, users often select their new usernames by changing their previous usernames. They often use a combination of the following simple modifications to create a new username [9]:

- adding prefixes or suffixes (e.g., jack.sparrow → jack.sparrow01),
- making abbreviation (e.g., jack.sparrow → jsparrow), and
- changing or adding some characters (e.g., jack.sparrow → jackkk.sprrw).

The basic idea behind this intuition is that display names and usernames which belong to the same user often have redundant information [9], [15]. Some of the studies, therefore, try to identify users based on the only display name, only username, or both [4], [15], [19], [20].

In this paper, we perform user matching across different OSNs for Turkish users. For this purpose, we created a dataset by taking the usernames of users who disclose their connections to other OSNs in their Facebook profiles. We build a learning function to determine whether two accounts match (i.e., belong to the same user) and perform this matching with usernames only. To build the learning function, we use similarity comparison (SC) and binary classification (BC) approaches from which the former makes its decision by comparing the similarity score between two vectorized usernames against a predefined threshold value. The latter, on the other hand, makes its decision with the help of a machine learning classifier. We additionally employ a combined model that uses extended feature space to train and test a binary classifier.

We would like to note that username-based matching respects users' privacy since it reduces the use of attributes as well as the degree of computational complexity [25]. To the best of our knowledge, this study is the first one for matching Turkish users' accounts across different OSNs, which is conducted on a real-world dataset.

The remainder of this paper is organized as follows: In Section 2, we review the previous user identification studies across different OSNs. In Section 3, we explain our methods used in this paper. In Section 4, we present a quantitative description of our data set. We then give our experimental results in Section 5, and finally, we present our discussion along with conclusions of the findings and give our future research directions in Section 6.

2. Literature Review

As mentioned in Section 1, user matching across different sites can be useful for many purposes. It can also be used to violate the privacy and security of OSN users. As such, it has attracted the interest of the research community. In this section, we present our review of the literature with a focus on studies performing user identification/matching across different OSNs. We would like to note that these studies were compiled through careful analysis of prominent journals and conference proceedings with the keywords "online social networks", "user profile matching", "user identification", "user identity linkage", and "user account matching".

The published works on this topic are as follows: user identification across three OSNs is performed by using solely display names of users in [19]. In this study, authors obtained a 0.9 AUC (Area under the ROC Curve) value on their dataset and they found that more than 45% of users prefer to use the same display name on different

OSNs. User matching based on the only username is performed in [20]. In this work, the authors use a similarity score that is obtained on self-information vectors to decide whether two usernames belong to the same user or not.

Online digital footprints (i.e., aggregated information from different OSN accounts that belong to the same user) are extracted across OSNs by using public profile attributes in [21]. In this study, the authors discovered that username and display name are the most discriminative information for user identification. A set of properties, namely, ACID (Availability, Consistency, non-Impersonality, and Discriminability) [6] is defined to evaluate matching tasks based on profile attributes in a reliable way [8]. In this study, it is claimed that accuracies reported in previous studies are lower than in practice.

A novel approach is proposed in [26] to develop a search engine that helps to identify users across multiple OSNs. The authors demonstrated that their approach provides a significant improvement in terms of performance accuracy. The information redundancies in k-hop ($k > 1$) neighbors and their contributions to user identification have been addressed in [27]. The authors utilized information redundancies obtained by analyzing similarities of k-hop neighbors of users. They also used friendship-based information redundancies jointly with the display-name-based information redundancies in the user matching task. Based on their results, the authors showed that friendship-based information redundancies especially for the 1-hop neighbors are very useful for user identification, and jointly applying redundancies provides better performance [27].

A profile comparison tool is developed in [7] to identify users across OSNs. This tool decides to match two accounts by calculating the similarity of their vectorized profile attributes. The authors evaluated their tool on a dataset that contains a small number of positive pairs and measured the performance (i.e., 83%) as the number of correct predictions. A user matching framework that employs profile attributes is presented in [10]. This framework makes it possible to assign different similarity measures for attributes. The authors of this work evaluated this framework on randomly generated OSN profiles and found that this approach outperforms classical methods.

An extended stable matching method based on profile and content attributes is proposed in [24]. In this study, the authors showed that their method identifies up to 70% of user accounts. An algorithm, namely, FRUI is proposed in [17] to calculate the degree of matching for all candidate pairs. In this study, the authors consider top-ranked pairs as identical and demonstrate that their algorithm outperforms other network structure-based methods. In [4], the matching task is considered as a binary classification task and the authors obtained the best F1 score as a value of 0.962 by using attributes extracted only from display names of users. A co-training method is proposed in [28] to speed-up the user matching process. In this approach, both profile and network structure information are used and it is found that co-training performs better than profile and relation information models.

Another machine learning-based framework, namely, MUSIC is proposed in [23]. In this study, the authors use both word2vec and doc2vec models to represent the account owner's contents such as post, message, etc. This framework achieves an F1 score of 0.893 just by using feature vectors extracted from user-generated contents. An iterative user identification algorithm is proposed in [5] that uses both public profile and network

structure information to iteratively match profiles. This algorithm obtains a high precision of 0.94 on a quite large dataset. In [3], profile and network structure-based features are used to match user profiles with the help of binary classification. In this study, the authors obtained a 0.98 AUC value on a dataset that includes 158 positive and 306 negative pairs from two different OSNs.

Different features extracted from usernames are employed for mapping (i.e., matching or identifying users) users in [9]. In this study, the proposed methodology, namely, MOBIUS achieved an accuracy of 0.938 with the help of binary classification. A user identification framework is also proposed in [15]. This framework achieves an F1 score over 0.9 by using features extracted from both display names and usernames of users. SocialMatching++ that uses events and biographies for user matching is proposed in [22] to enhance attribute approaches.

Besides the matching efforts summarized above, there also exist other studies which are aiming at privacy-preserving user identification. Such types of studies mainly focus on performing user identification over OSNs and databases by using encrypted (or anonymized) data or some statistics like histograms instead of the actual information of users [11], [12]. This is because user identification is not just used for nefarious intentions but is also needed for some other good purposes like friendship recommendation, advertising, and so on.

In this study, we perform username-based profile matching for Turkish OSN users. We conduct our experiments on a real-world dataset that has been obtained from a crawled public Facebook data [13]. Even there exist many studies in this field, it seems that this is the first study that aims to match different OSN accounts for Turkish users. As our method only relies on usernames, it reduces the use of attributes as well as the degree of computational complexity. Username-based matching has an additional advantage in that it is often highly accessible and respects the personal privacy of users.

3. Methods

User matching, (see Figure 1), is often solved by building a learning function. Let u_1 and u_2 be arbitrary usernames in two different OSNs, respectively. Then, a learning function f can be formulated as follows [9], [20]:

$$f(u_1, u_2) = \begin{cases} 1 & \text{if } u_1 \text{ and } u_2 \text{ belong to the same user} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The function f is often learned in two different ways: using (i) vector-based comparison (i.e., the similarity between usernames), and (ii) machine learning [14], [15]. Firstly, various similarity and matching methods are used to build a learning function [7], [10], [20], [22], while supervised binary classification is employed in the second way [3], [4], [8], [9], [19], [21], [23], [28]. The matching task is performed on the data instances that include positive and negative pairs in both of the two approaches. Positive instances are comprised of two different usernames that belong to the same user. On the other hand, negative instances include the usernames of two different users.

3.1. Feature Extraction

We borrowed features that have been used in the previous studies for the purpose of username-based user identification. We used two different feature sets from which the first set includes 1,373

TABLE 1: List of content and pattern features that we borrowed from [20].

Feature Set	Definition	# of features
Content Features	Possible bigram patterns of letters and digits (i.e., “ab”, “00”, “cd”, “09”, “mn”, etc.)	1,296
Pattern Features	Letter digit gram patterns (e.g., LLLDDL) with length of 6	64
	Letter digit permutations (i.e., “only letters”, “only digits”, “letters + digits”, “digits + letters”, “letters + digits + letters” and “digits + letters + digits”)	6
	Date patterns (i.e., “year + month + day”, “month + day + year”, “day + month + year” and “month + day”)	4
	Keystroke patterns (i.e., “each two consecutive characters are adjacent but not the same row”, “all characters in the same row”, “each character is the same or adjacent with the previous one”)	3

TABLE 2: List of similarity and distance functions that we used to extract features.

Feature	Definition
Length Difference	$\Delta\text{Len} = \text{abs}(\text{len}(u_1) - \text{len}(u_2))$
Length Ratio	$\text{Ratio} = \min(\text{len}(u_1), \text{len}(u_2))/\max(\text{len}(u_1), \text{len}(u_2))$
Length of LCSeq Comparing Minimum Length	$\text{Sim}_{\text{LCSeq}} = \text{len}(\text{LCSeq}(u_1, u_2))/\min(\text{len}(u_1), \text{len}(u_2))$
Length of LCS Comparing Minimum Length	$\text{Sim}_{\text{LCS}} = \text{len}(\text{LCS}(u_1, u_2))/\min(\text{len}(u_1), \text{len}(u_2))$
Edit Distance Comparing Maximum Length	$\text{Sim}_{\text{Edit}} = \text{edit}(u_1, u_2)/\max(\text{len}(u_1), \text{len}(u_2))$
Jensen-Shannon Similarity of Alphabet Distribution	$\text{Sim}_{\text{JSD}} = 1 - 1/2(\text{KL}(P \parallel M) + \text{KL}(Q \parallel M))$ where $M = 1/2(P + Q)$, and $\text{KL}(P \parallel Q) = \sum_{i=1}^{\vert} P\vert P_i.\log\frac{P_i}{Q_i}$
Cosine Similartiy of Letter Frequencies	$\text{Sim}_{\text{Cosine}} = 1 - \cos(P \parallel Q)$ where $\cos(P \parallel Q) = \frac{\sum_{i=1}^{n}(P_i \times Q_i)}{\sqrt{\sum_{i=1}^{n}(P_i)^2} \times \sqrt{\sum_{i=1}^{n}(Q_i)^2}}$
Jaccard Similarity of Letters	$\text{Sim}_{\text{Jaccard}} = \text{len}(\text{letters}(u_1) \cap \text{letters}(u_2))/\text{len}(\text{letters}(u_1) \cup \text{letters}(u_2))$
Jaro-Winkler Distance	$\text{Dist}_{\text{Jaro}} = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}(\frac{m}{\vert u_1\vert} + \frac{m}{\vert u_2\vert} + \frac{m-t}{m}) & \text{otherwise} \end{cases}$ where, m→ # of matching characters, t → # of transpositions
Bigram and Trigram Similarities	$\text{Sim}_{\text{N-gram}}$: # of common/# of n − grams in shorter username
VMN similarity	We advise the reader to refer to [3] for details.
Dynamic Time Warping Distance	We advise the reader to refer to [9] for details.

content and pattern features. On the other hand, our second feature set includes features that have been obtained by employing 12 different similarity or distance functions. Table 1 presents the list of our pattern and content features, while Table 2 summarizes the distance or similarity functions that we used to extract features in the second set.

As seen from Table 1, the first set of features generally depends on some patterns or permutations. On the contrary, the second set of features depends on different similarity or distance functions that have been mostly employed both in the username and display name based matching studies [3], [4], [8]–[10], [15], [21]. Notice that in Table 2, we advise the reader to read [3] and [9] for more details about VMN similarity and Dynamic Time Warping Distance methods respectively, so as to reduce the complexity of this paper.

3.2. Classification

To build the learning function f, we use different approaches. We first try to match users

by using a simple model that gives its decision based on the similarity between two usernames. Secondly, we employ machine learning classifiers to build a binary classification model. Finally, we build a combined model that uses machine learning classifiers on a combined (i.e., extended) list of features.

3.2..1 Similarity comparison (SC)

In this approach, the f function uses an inner similarity or matching method to calculate a score on feature vectors of which each one corresponds to a single username. Afterward, the f function returns 1 or 0 depending on the score calculated by the inner method. The f function, that gives its decision by comparing the computed similarity score with a predefined threshold value, is shown as follows [20]:

$$f(u_1, u_2) = \begin{cases} 1 & \text{if } \mathrm{sim}(u_1,\ u_2) \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\mathrm{sim}(u_1,\ u_2)$ is the inner similarity method and τ is the predefined threshold value. To compute the inner similarity between two usernames, we represent usernames by using both binary and self-information vector models from which the latter is introduced in [20]. Let a and u be a feature (i.e., attribute) and a username, respectively. Then a feature indicator function is defined as follows [20]:

$$I_a(u) = \begin{cases} 1 & \text{if } u \text{ satisfies the feature } a \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Given m features $\{a_1,\ a_2, \ldots, a_m\}$, a username is represented by both binary and self-information vector models as follows [20]:

$$\mathrm{V_{Binary}} = \langle \mathrm{I_{a_1}(u)}, \ldots, \mathrm{I_{a_m}(u)} \rangle$$

$$\mathrm{V_{Self-info}} = \langle \mathrm{I_{a_1}(u).W(a_1)}, \ldots, \mathrm{I_{a_m}(u).W(a_m)} \rangle$$

where $W(a)$ is estimated self-information of feature a and can be obtained on username set U as follows:

$$W(a) = \frac{|\{u \in U | I_a(u) = 1\}|}{|U|} \tag{4}$$

In the similarity comparison (SC) approach, we use the cosine similarity between two vectorized usernames to decide whether they belong to the same user or not.

3.2..2 Binary classification (BC)

In this approach, a dataset (i.e., a training set that contains both positive and negative instances) is firstly converted into a classification-ready structure. In this structure, each instance is represented as a feature vector that each dimension stands for a value of each of the extracted features. Afterward, a binary classifier is trained on this dataset to build the learning function f.

Upon completion of the training, the binary classifier is expected to output whether a new pair of usernames constitute a negative or positive instance match or not. In this study, we employ binary classification with the help of Weka [29]'s well-known machine learning algorithms including Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), SMO (Sequential Minimal Optimization), Random Forest (RF), Decision Trees (J48), and Support Vector Machines (SVM).

3.2..3 Combined Model

This model uses the binary classification approach on extended feature space. As seen from Table 1, content and pattern features are extracted for each username. On the contrary,

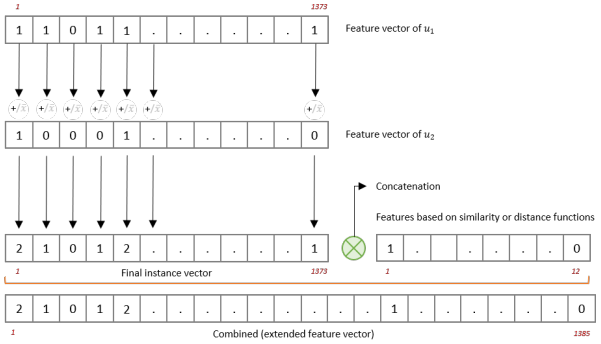


Fig. 2: Feature extension in the combined model.

as seen from Table 2, features depending on similarity or distance functions are extracted for each pair of usernames. This means that we have two feature vectors when we use features listed in Table 1, while we have one instance vector once we use features listed in Table 2.

As machine learning classifiers require one feature vector per instance, we have to convert two feature vectors into a single instance vector when we employ features of Table 1. In order to do so, we take the average ($\overline{x}$) or sum ($+$) of each feature's values and created our final instance vector. We then combine this final instance vector with the features of Table 2 to obtain our extended feature space. This simple process is depicted in Figure 2. Notice that we represent features of Table 1 by using binary or self-information vector models and create a final instance vector by taking the sum or average of feature vectors so as to explore effects on classification performance. Our combined feature vector now consists of 1,385 features.

3.3. Measuring feature importance

To measure feature importance, we use the mutual information (MI) method which is also widely used for the purpose of feature selection. Given to random variables x and y, their MI is defined in terms of their probabilistic density functions $p(x), p(y)$, and $p(x, y)$ [30]:

$$MI(x;y) = \iint p(x,y) log \frac{p(x,y)}{p(x)p(y)} dxdy. \quad (5)$$

MI is a measure between two random variables which quantifies the amount of information obtained about one random variable, through the other random variable. In this paper, we use MI to measure dependence between features and matching status by using its implementation in yellowbrick [31] Python package.

TABLE 3: A quantitative description of crawled public Facebook data [13].

Property (# of)	Count
users (nodes) whose entire profile has been crawled	20,000
friendship links (edges) among crawled users	402,300
users whose direct friends have been crawled	261
total friendship links	3,980,270
unique accounts discovered	2,350,454

3.4. Evaluation

We measure the performance of the learning function f by using the F1 score for both SC and BC approaches. Let us define the following values depending on the output of the function f:

- TP: # of correctly predicted positive instances,
- TN: # of correctly predicted negative instances,
- FN: # of wrongly rejected positive instances,
- FP: # of wrongly rejected negative instances.

The F1 score can be calculated as follows:

$$F1 = (2 \times P \times R)/(P + R) \quad (6)$$

where P (Precision) $= \frac{\text{TP}}{\text{TP+FP}}$ and R (Recall) $= \frac{\text{TP}}{\text{TP+FN}}$.

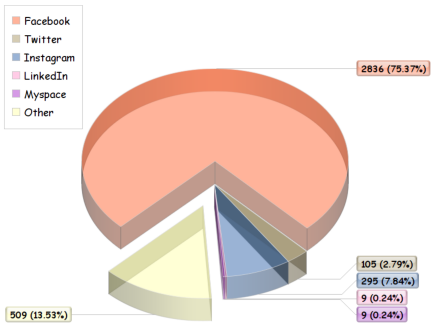


Fig. 3: Distribution of disclosed connections of users in crawled Facebook data.

4. Dataset

The dataset utilized in this study is collected with a web crawler that collects public Facebook data from user accounts. The design choices and implementation details of the crawler along with detailed statistical analyses of the collected OSN data can be found in [13]. This crawler basically outputs a breadth-first traversal of all public Turkish Facebook accounts starting at a seed account. In this study, we work with the largest snapshot of the crawled real-world Facebook data, the basic statistical properties of which are detailed in Table 3.

Facebook OSN allows users to publicize their social media accounts through their profile pages. This feature, called "connections", links a Facebook account to other Facebook accounts of the same user, as well as accounts over distinct other OSNs such as Twitter, Myspace, Instagram, and LinkedIn. A connection lists the platform on which the shared account resides and the username on that platform. For example, "Twitter (jack.sparrow25)" is a connection to a Twitter account with the username "jack.sparrow25".

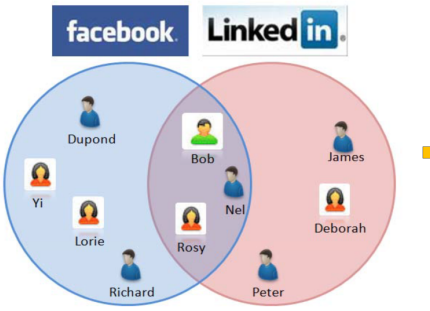
Connection shares are utilized quite commonly. Among the 20K users in our Facebook data set, only 3,7K of users disclose at least one "connection". The distribution of these connections by the service provider is depicted in Figure 3, where connections that do not link to another OSN service were categorized under "other".

TABLE 4: A sample of positive username pairs from our data set.

User No	Username on	
	OSN1	OSN2
1	murat.d****n.19**	mrt_d****n
2	lutfu.b****z	lutfu****z
3	kemal.t****n.1*	t****n.kemal
4	mye.t****n	t****necrin
5	e****.sahin.7*2	e****.sahin.7*2

As seen from Figure 3, quite interestingly, a large majority (≈75%) of users share their Facebook account that they already reside. Only 418 of all 3.7K users share connections to other OSN services. Additionally, we detected that 94 of these 418 connections are completely comprised of a numeric string username (e.g., "12******9") that is possibly assigned by the social site automatically. In this situation, username-based matching methods fail to work properly, due to such usernames contain almost no information redundancies [4], [15].

As such, we excluded these numeric usernames from that of the 418 usernames to make our methods more suitable. This elimination process has resulted in a total of 324 connections that form the base of our training/test data set. As explained before, our matching process relies only on usernames and disregards the underlying graph structure (e.g., friendship links). Consequently, the training/test data set is comprised of pairs of OSN-usernames alongside the binary class label. If the two usernames belong to the same individual, we say that the corresponding instance is positive.



User(s)	Positive instance pairs		Label
Bob	*bobatfacebook*	*bobatlinkedin*	Positive
Nel	*nelfb****	*linkedinnel**	Positive
...	...	...	...
Rosy	*fb_rosy.****	*lin_rosy.****	Positive

User(s)	Negative instance pairs		Label
Dupon-James	*dupon***fb*	*lin**.james*	Negative
Rosy-Peter	*fb_rosy.****	*peter**	Negative
...	...	...	...
Lorie-Nel	*facelorie****	*linkedinnel**	Negative

Fig. 4: An example on the creation of positive and negative instances based on the OSNs of Figure 1. Users have been assumed to have selected the arbitrary usernames given in the tables on right.

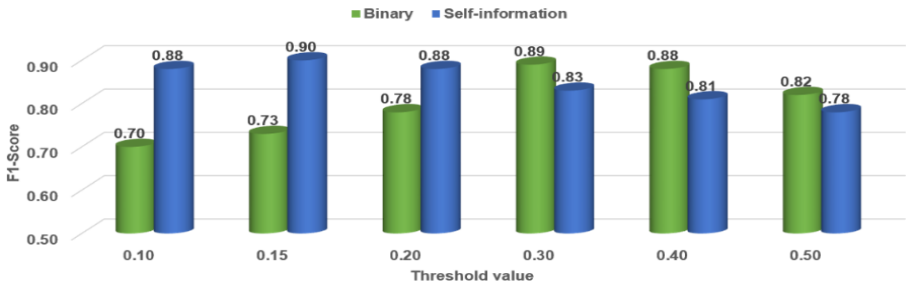


Fig. 5: Results of the SC approach with respect to different vector models and threshold values.

We sampled positive instances using the connection shares in our Facebook data set. Table 4 provides some positive samples for reference purposes. As seen from Table 4, some users prefer to use the same usernames across distinct OSNs, where others (possibly due to unavailability of the same username in the other OSN) make very simple changes in their original usernames such as adding prefixes or suffixes, making abbreviations, and so on. We would like to note that some letters of the usernames listed in Table 4 have been masked with asterisks so as to respect the privacy of corresponding users.

Negative instances were fabricated using a fail-safe strategy: we randomly picked two users of our data set, such that at least one of which shared at least one connection. Let these users be denoted with u and v respectively. We assumed that u and v belong to distinct individuals.

Based on this assumption, we randomly paired a username from u's known OSN accounts (except for Facebook) with a username from v's known OSN account(s). We consider this strategy to be fail-safe because at least one of the owners of accounts u and v use the connections feature of Facebook and neither account lists the other as belonging to the same user if both use the connections feature.

This process, which is exemplified in Figure 4, yielded a total of 324 positive real-world instances in our training/test data set. In order to ensure a balanced distribution of positive and negative instances, we also created 324 negative instances, which means that our dataset is comprised of 648 real-world instances in total.

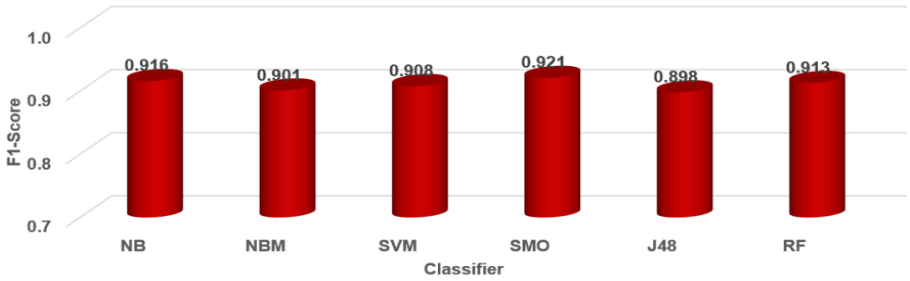
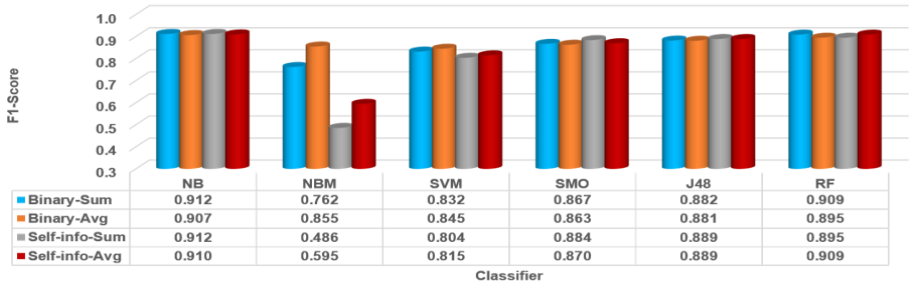


Fig. 6: Results of the BC approach with respect to different classifiers.



	NB	NBM	SVM	SMO	J48	RF
Binary-Sum	0.912	0.762	0.832	0.867	0.882	0.909
Binary-Avg	0.907	0.855	0.845	0.863	0.881	0.895
Self-info-Sum	0.912	0.486	0.804	0.884	0.889	0.895
Self-info-Avg	0.910	0.595	0.815	0.870	0.889	0.909

Fig. 7: Results of the combined model with respect to the classifier, feature representation model, and conversion operator.

5. Experimental Results

To obtain classification results, we first pre-processed all connections in our dataset and only selected OSN account links. We then applied lowercase conversion and punctuation removal on usernames. After the pre-processing, we performed feature extraction to form instance vectors of positive and negative pairs in our dataset.

We then built the learning function f using different approaches. We first applied the similarity comparison (SC) with the feature set given in Table 1. In this approach, the learning function makes its decision by comparing the cosine similarity between usernames across a predefined threshold value. To obtain the results of this approach, we used different threshold values to observe the optimum value. We also represented usernames by using the self-information vector model to compare it against the binary vector model. Figure 5 presents our results obtained with the SC approach. As seen from Figure 5, in the binary vector model, matching success tends to increase as threshold value increases, while the opposite is true for the self-information model. The binary vector model achieves the best F1 score of 0.89 when the threshold is set to 0.3, while the self-information model obtains the best F1 score of 0.90 with a threshold of 0.15. This shows that the results of the two vector models are slightly different, but the self-information model provides a better result than the binary representation of feature vectors. This is because the self-information model considers the popularity of features among all instances in the dataset.

Secondly, we employed the binary classification (BC) approach using the feature set summarized in Table 2. In this step, we trained different classifiers and investigated the effect of the se-

lected classifier on the performance. We used default parameter settings for all of the classifiers which were configured to run with 10-fold cross-validation. Figure 6 presents our results obtained with the BC approach. As seen from Figure 6, the F1 scores of classifiers are slightly different, but the most successful classifier is SMO with the best F1 score of 0.921. This shows that the BC approach outperforms the SC approach even though it uses far fewer features. The reason for this is that the BC approach uses machine learning classifiers that are able to learn and make a generalization from given features.

As the last step of our classification experiments, we employed the combined model that uses the extended feature space obtained by concatenating features (see Figure 2) of Table 1 and Table 2 respectively. The results for different classifiers obtained with 10-fold cross-validation are presented in Figure 7. As seen from Figure 7, combining features of Table 1 and Table 2 does not improve the best F1 score of the binary classification approach. The best F1 score is obtained as a value of 0.912 with the help of the NB classifier, even though classifiers produce slightly different results. Taking the average of vectors of usernames provides slightly different results than the sum operator. On the other hand, the binary representation of username feature vectors seems to be a better choice than using self-information vector representation.

Excluding the MNB, it is clear that classifiers often produce slightly different results. MNB classifier is mostly the worst one as it works well for data that can easily be converted into frequency values, such as word counts in text. Next, to explore the most important features, we use the MI method on the best performant representation of data, where features of Table 1 are represented by the binary vector model were converted to a final instance vector using the sum operator. This final instance vector is then concatenated with the corresponding vector of features that were extracted with functions of Table 2. Using this best representation, we obtained feature dependencies with respect to MI scores. Figure 8 depicts the top 20 most important features.

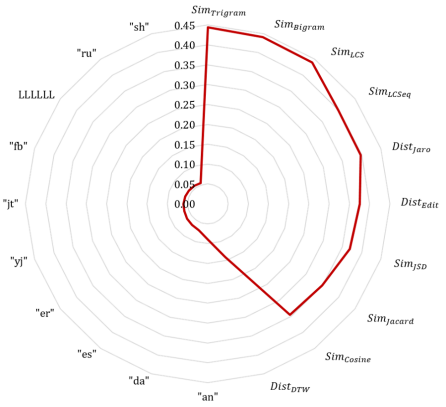


Fig. 8: A spider chart of top 20 features with respect to their MI scores.

As seen from Figure 8, the most important features depend on similarity or distance functions including n-gram similarities, the similarity between longest common substrings (Sim_{LCS}), the similarity between longest common subsequences (Sim_{LCSeq}), Jaro-Winkler distance ($Dist_{Jaro}$), edit distance (Sim_{Edit}), Jensen-Shannon similarity of alphabet distribution (Sim_{JSD}), and so on. Additionally, the LLLLLL feature is the most important one among letter digit pattern features. It is also clear that bigram patterns of letters (e.g., “an”, “da”, “es”, etc.) are among the most important features.

Using these MI scores of the features, we built a simple yet effective experiment scenario that involves incorporating a feature selection process in our classification task. We selected the top-

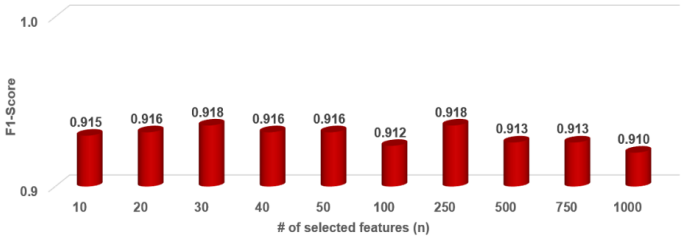


Fig. 9: Results of the NB classifier with respect to the number of selected features.

n features based on their MI scores. We then used the selected features in the classification phase. We used the NB classifier because it was the most successful classifier in the combined model. Figure 9 presents the results of this final experiment where the number of features to select (i.e., n) takes different values in a range between 10 and 1,000.

As seen from Figure 9, feature selection makes a very low contribution to the best results of the combined model. When the number of selected features is 30 or 250, the NB classifier obtains its highest result of 0.918, which is higher than the best F1 score (i.e., 0.912) of the combined model employed without feature selection. On the other hand, the best F1 score obtained with the feature selection falls behind that of the best F1 score of the BC approach employed without feature extension.

6. Discussion and Conclusion

In this paper, we studied the problem of user matching across different OSNs for Turkish users. For this purpose, we first created a dataset by crawling Facebook which is one of the most popular OSNs in the world. As users are able to disclose their social connections on their Facebook accounts, we could collect other OSN connections of such users. Afterward, we applied a simple pre-processing on usernames and created a dataset that includes positive and negative pairs of real-world usernames. We then tried to match users based on their usernames by using different approaches including similarity comparison (SC) and binary classification (BC).

Our experimental results show that using the self-information of features provides a better result than their binary representation in the SC approach. The BC approach, on the other hand, provides better results than the SC approach. In the BC approach, extracting features based on distance and similarity functions has more contribution than content and pattern features into the classification success. We report our best F1 score as a value of 0.921 which is obtained by using features extracted based on well-known similarity and distance functions (see Table 2). This result is quite reasonable when compared to the existing studies' performances that range from 0.7 to 0.96 with respect to different metrics. This is because we perform matching task just based on usernames. Our best F1 score proves that users tend to use the same username or select a similar one for their OSN accounts.

In light of all of these findings, we conclude that usernames are fundamental and inevitable elements of OSNs and may often cause information redundancies. Using only usernames has some prominent advantages in that it respects users' privacy and reduces the use of attributes along with computation complexity. However, username-based profile matching may fail in many cases, since users may use different usernames and, in some cases, the username is not available. Additionally, in some OSNs like Foursquare, the username is a numeric string, which is automatically generated by social site [15]. These reasons make the username-based

matching method quite fragile.

On the contrary, even a matching system uses other information (e.g., profile attributes, content attributes, etc.), cross-referencing users between different OSNs is not an easy task as users may fill their profiles with different (even fake) details. Therefore, we also conclude that username-based matching needs to be studied further and it is more preferable to use it as a part of a complete cross-referencing system.

For users, we suggest not disclose their sensitive and discriminative (e.g., email) information on their OSN accounts. Further, for their newly created OSN accounts, users should select or create a new username that does not have information redundancy and does not reflect their characteristics and habits. This is because their information can be used for different malicious activities where user identification may have an important role.

As future work, we will mine the textual contents of users to learn/infer their other OSN connections. Hereby, we will extend our dataset and use it to study user mapping with the help of previously unused record matching techniques.

References

[1] M. Wani, N. Agarwal, S. Jabin, and S. Hussai. "Design and Implementation of iMacros-based Data Crawler for Behavioral Analysis of Facebook Users", Computer Science: Social and Information Networks, February 2018.

[2] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. "On the evolution of user interaction in facebook", Proceedings of the 2nd ACM workshop on Online social networks, Barcelona, Spain, pp. 37-42, 16-21 August 2009.

[3] O. Peled, M. Fire, L. Rokach, and Y. Elovici. "Entity matching in online social networks", IEEE international conference on social computing (socialcom), Washington, USA, pp. 339-344, 8-14 September 2013.

[4] Y. Li, Y. Peng, W. Ji, Z. Zhang, and Q. Xu. "User identification based on display names across online social networks", IEEE Access, Vol.5, pp. 17342-17353, August 2017.

[5] N. Bennacer, C. N. Jipmo, A. Penta, and G. Quercini. "Matching user profiles across social networks", International Conference on Advanced Information Systems Engineering, Thessaloniki, Greece, pp. 424-438, 16-20 June 2014.

[6] O. Goga. "Matching user accounts across online social networks: methods and applications", Université Pierre et Marie Curie, LIP6-Laboratoire d'Informatique de Paris 6, Doctoral dissertation, 151p, Paris, France, May 2014.

[7] J. Vosecky, D. Hong, and V. Y. Shen. "User identification across multiple social networks", IEEE First International Conference on Networked Digital Technologies, Ostrava, Czech Republic, pp. 360-365, 29-31 July 2009.

[8] O. Goga, P. Loiseau, R. Sommer, R. Teixeira, and K. P. Gummadi. "On the reliability of profile matching across large online social networks", Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, pp. 1799-1808, 10-13 August 2015.

[9] R. Zafarani, and H. Liu. "Connecting users across social media sites: a behavioral-modeling approach", Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, Chicago, USA, pp. 41-49, 11-14 August 2013.

[10] E. Raad, R. Chbeir, and A. Dipanda. "User profile matching in social networks", IEEE 13th International Conference on Network-Based Information Systems (NBiS), Takayama, Japan, pp. 297-304, 14-16 September 2010.

[11] X. Yi, E. Bertino, F. Y. Rao, K. Y. Lam, S. Nepal, and A. Bouguettaya. "Privacy-Preserving User Profile Matching in Social Networks", IEEE Transactions on Knowledge and Data Engineering, Vol.32, No.8, pp. 1572-1585, August 2020.

[12] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli. "Where you are is who you are: User identification by matching statistics", IEEE Transactions on Information Forensics and Security, Vol.11, No.2, pp. 358-372, February 2016.

[13] O. Coban, A. Inan, and S. A. Ozel. "Towards the design and implementation of an OSN crawler: a case of Turkish Facebook users", International Journal of Information Security Science, Vol.9, No.2, pp. 76-93, June 2020.

[14] S. I. Bhat, T. Arif, and M. B. Malik. "A Framework for User Identity Resolutions across Social Networks", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol.4, No.1, pp. 307-313, March-April 2018.

[15] Y. Li, Y. Peng, Z. Zhang, H. Yin, and Q. Xu. "Matching user accounts across social networks based on username and display name", World Wide Web, Vol.22, pp. 1095-1097, April 2018.

[16] P. Jain. "Automated methods for identity resolution across online social networks", Indraprastha Institute of Information Technology Delhi, Doctoral dissertation, 137p, New Delhi, India, April 2016.

[17] X. Zhou, X. Liang, H. Zhang, and Y. Ma. 2016. "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, February 2016.

[18] R. Kaushal. "A Systematic Review on User Identity Linkage across Online Social Networks", Indraprastha Institute of Information Technology Delhi, Doctoral dissertation, 50p, New Delhi, India, February 2020.

[19] Y. Li, Y. Peng, Z. Zhang, M. Wu, Q. Xu, and H. Yin. "A deep dive into user display names across social networks", Information Sciences, Vol.447, pp. 186-204, June 2018.

[20] Y. Wang, T. Liu, Q. Tan, J. Shi, and L. Guo. "Identifying users across different sites using usernames", Procedia Computer Science, Vol. 80, pp. 376-385, June 2016.

[21] A. Malhotra, L. Totti, W. Meira Jr, P. Kumaraguru, and V. Almeida. "Studying user footprints in different online social networks", Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), Istanbul, Turkey, pp. 1065-1070, 26-29 August 2012.

[22] H. Hazimeh, E. Mugellini, O. A. Khaled, and P. Cudré-Mauroux. SocialMatching++: "A Novel Approach for Interlinking User Profiles on Social Networks", The 16th International Semantic Web Conference (ISWC), Vienna, Austria, 21-25 October 2017.

[23] Y. Sha, Q. Liang, and K. Zheng. "Matching user accounts across social networks based on users message", Procedia Computer Science, Vol. 80, pp. 2423-2427, June 2016.

[24] Q. Liu, J. Li, Y. Wang, G. Xing, and Y. Ren. "Account matching across heterogeneous networks", IEEE 5th International Conference on Game Theory for Networks (GAMENETS), Beijing, China, pp. 1-5, 25-27 November 2014.

[25] L. Xing, K. Deng, H. Wu, P. Xie, and J. Gao. "Behavioral Habits-Based User Identification Across Social Networks". Symmetry, Vol.11, No.9, pp. 1134, September 2019.

[26] H. Van Pham, and V.T. Nguyen. "A novel approach using context matching algorithm and knowledge inference for user identification in social networks". Proceedings of the 4th International Conference on Machine Learning and Soft Computing, Haiphong City, Viet Nam, pp. 149-153. 17-19 January 2020.

[27] Y. Li, Z. Su, J. Yang, and C. Gao. "Exploiting similarities of user friendship networks across social networks for user identification", Information Sciences, Vol.506, pp. 78-98, January 2020.

[28] Z. Fang, Y. Cao, Y. Liu, J. Tan, L. Guo, and Y. Shang. "A co-training method for identifying the same person across social networks", IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, Canada, pp. 1412-1416, 14-16 November 2017.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. "The WEKA data mining software: an update", ACM SIGKDD explorations newsletter, Vol.11, No.1, pp. 10-18, November 2009.

[30] H. Peng, F. Long, and C. Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy", IEEE Transactions on pattern analysis and machine intelligence, Vol.27, No.8, pp. 1226-1238. August 2005.

[31] B. Bengfort, and R. Bilbro. "Yellowbrick: Visualizing the scikit-learn model selection process", Journal of Open Source Software, Vol.4, No.35, pp. 1075, March 2019.