Evaluating the impact of friends in predicting user's availability in Online Social Networks

Andrea De Salve[1,2], Paolo Mori[1], and Laura Ricci[2]

[1] Institute of Informatics and Telematics - National Research Council, Pisa, Italy
[2] University of Pisa - Department of Computer Science, Pisa, Italy.
e-mail: desalve@di.unipi.it - paolo.mori@iit.cnr.it - ricci@di.unipi.it

Abstract. In recent years, Online Social Networks (OSNs) have changed the way people connect and interact with each other. Indeed, most people have registered an account on some popular OSNs (such as Facebook, or Google+) which is used to access the system at different times of the days, depending on their life and habits. In this context, understanding how users connect to the OSNs is of paramount importance for both the protection of their privacy and the OSN's provider (or third-party applications) that want to exploit this information. In this paper, we study the task of predicting the availability status (online/offline) of the OSNs' users by exploiting the availability information of their friends. The basic idea is to evaluate how the knowledge about availability status of friends can help in predicting the availability status of the center-users. For this purpose, we exploit several learning algorithms to find interesting relationships between the availability status of the users and those of their friends. The extensive validation of the results, by using a real Facebook dataset, indicates that the availability status of the users' friends can help in predicting whether the central user is online or offline.

Keywords: Personal behavior, Availability prediction, Online Social Networks

1 Introduction

Online Social Networks (OSNs) have attracted millions of users, that connect everyday to the OSNs in order to share information with their friends or to directly interact with those who are online. The daily activities performed by each user produce a huge amount of private data that can be retrieved and exploited for different purposes, such as to predict the user's behavior. As for instance, the most part of current OSNs provide tools (such as the Chat Status) that can be used to obtain availability status (online/offline) of their users. In this scenario, a user u, who is friend of z, could access the availability status of the friends in common with z. Supposing that z wants to protect his availability status from being disclosed to u, the availability information that u could collect about z' friends may be considered as a threat to the privacy of z, because u can exploit such knowledge to infer the availability status of z. Understanding how

this information impact the predictability of the user's behavior is essential to protect the privacy of the OSNs' users. Unfortunately, the problem of predicting the availability status of a user z by exploiting the availability status of the friends of z has not been investigated for the privacy protection in a user-centric scenario (see Sec. 2). In addition, the problem of predicting the availability status has many real-world applications, like deciding what is the best time to send instant notification to a OSN's user and managing important problems in a distributed scenario (such as data availability and information diffusion [10]). The aim of this paper is to investigate the task of predicting availability status of OSN's users by exploiting a large real Facebook dataset, containing availability chat status of a set of users for 32 consecutive days. The data obtained from Facebook are used to train an extensive set of learning algorithms and to validate their ability in solving the task of predicting the availability status of the users. Using this sample of Facebook's users, we also evaluate the performance of each algorithm and we highlight some important properties and issues.

The rest of the paper is structured as follows: Sec. 2 provides an overview of the related works studying the temporal users' behavior in OSN while Sec. 3 presents the Facebook dataset used for the evaluation of our results, the general characteristics of the dataset (see Sec. 3.1), and the preparation of such dataset for the task of classification (see Sec. 3.2). Sec. 4 describes the classification algorithms used in our experiments while Sec. 5 validates their results by exploiting several quality measures. Finally, Sec. 6 reports the conclusions and discusses the future works.

2 Related works

The study of temporal properties of OSNs' users have gained attention from researchers and several works have been proposed. In [17], authors propose to predict the availability status of a user's peer by combining results of different predictors. The first predictor labelled a users' peers based on their uptime status (namely, strongly/ weakly offline, and strongly/ weakly online). The second and the third predictors exploits Bruijn graph to represent the recent availability patterns of the users and to reduce the noise of variation on these patterns. Finally, the authors consider linear predictor that exploits the availability status of users in the last k days.

Authors of [4] demonstrated the existence of regular patterns in users' behavior. In addition, they implement a linear predictor which exploits the last 7 days of history to predict their online periods for the next week. However, the proposed approach is evaluated through an epidemic distributed protocol which chooses good partners by using the linear predictor.

Authors of [5] exploit a real dataset derived from MySpace to show that user's availability is correlated to both the time of the day and the presence of their friends on the platform.

Authors of [11] exploit probabilistic linear regression that is trained, by using different datasets, to minimize the Mean Squared Error. However, the authors

Table 1. General characteristics of the Facebook dataset

Name	Value
Start date	3 April 2015 - 10:05:00
End date	9 April 2015 - 12:55:00
Number of registered users	204
Total number of monitored users	66880
Number of time instants in a day T	288
Sampling frequency Δ	5min

do not use OSN dataset to train the model, but three different datasets derived from instant messaging and peer-to-peer applications.
Authors of [9] use a sample of Facebook users to investigate the relationships between the ego network structure of a user and the availability patterns of the friends in the ego network. In particular, the authors identified strong similarity (or temporal homophily) between the availability patterns of the users and their ego network's friends.
Authors of [2] analyzed four distinct OSNs derived from Orkut, MySpace, Hi5, and LinkedIn. In particular, they investigate the most recurrent actions performed by the users of the datasets and the types of contents shared with friends. Finally, another interesting analysis related to messages exchanged by the college students on Facebook is reported on [12]. The authors showed the presence of a weekly temporal pattern where users are clustered by similar patterns. They noticed the presence of a seasonal pattern in students' activities and showed that the weekend heavily impacts the typical users' patterns.

To the best of our knowledge, none of the previous works investigated the problem of predicting the availability status of the users by exploiting information derived from friends. In addition, the most part of related works adopt very simple models (such as linear predictor) to predict availability status of a user and they assume to know only the past availability status of the same user.

3 The Facebook dataset

We focus on Facebook OSN because it enables its user to infer temporal information about the behaviour of their friends by exploiting the Facebook Chat Status. In particular, we developed a Facebook Application[3] which exploits the Facebook API to monitor the users registered to the application, by sampling periodically the chat status of such users and their friends. The Facebook Chat API allows to check if a user is logged on Facebook, regardless of the actions he is doing during the session (such as browse contents or send messages). The temporal information collected from Facebook allow us to build session traces of users which indicate the presence or not of Facebook's users, at different time instants. Table 1 describes the characteristics of the dataset retrieved by our application.

[3] https://www.facebook.com/SocialCircles-244719909045196/

The Facebook application had been running for 32 days, i.e., from 9 March to 10 April 2015. During this time interval, we sampled the Facebook Chat status of all the registered users and their friends every 5 minutes. We decided to fix the sampling frequency (Δ) to 5 minutes because a shorter granularity was not possible for technical reasons related to the Facebook Graph API. In particular, our application was able to retrieve the following sets of information:

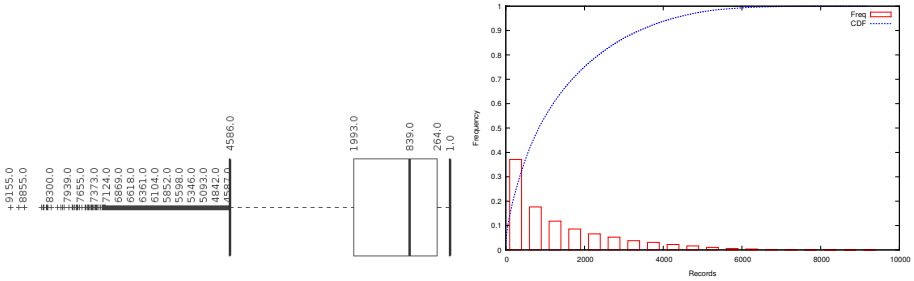
Friendship the set of friends of each registered user, and the friendship relations existing between them.

Online presence the availability status of the Facebook Chat of the registered users and their friends. The availability status can assume a limited set of value: 0 if user is offline, 1 if user is online.

Using this methodology we were able to access the availability status of 204 registered users and of their friends (for a total of 66.880 users). The set of registered users has the advantage of representing a very heterogeneous population: 114 males and 90 females, with age range of 20–79 with different education and background. In addition, the Since the majority of the registered users have geographical location in Italy or in central Europe. For the sake of clarity, we assume a discrete time model, where the time t_i of a day d is represented by positive integers $i = 1, 2, \ldots, T$ and the number of time instants T in a day is equal to 288 as it depends on the sampling frequency Δ, which is equal to 5 minutes.

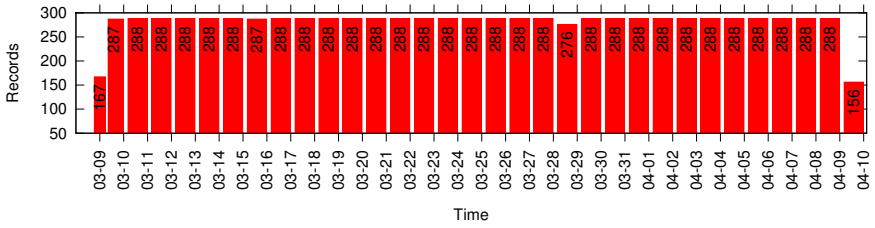
3.1 Data understanding

In this section we describe the general characteristics of our dataset. Due to technical reasons related to the memory space required by our application, the original dataset is a collection of records consisting of a pair $< t_i, u >$, where t_i is the time instant and u is the identity of a user that was found online at the corresponding time instant t_i. As a result, the dataset contains only temporal information about the users who are online for a total of $88,875,873$ records. The boxplot of Fig. 1(a) provides a quick overview of the extreme characteristics of our dataset by showing the number of records collected for each user. The box corresponds to the interquartile range (IQR), covering the middle 50% of the data, and it goes from the lower quartile (Q1=25%) to the upper quartile (Q3=75%). We have collected more than 839 records for more than 50% of the users and the IQR ranges in the interval [264,1993]. The dotted line of the box indicates the $1.5 \cdot IQR$ and the data points outside this limit are drawn separately.
In addition, the plot indicates the presence of a higher number of user with only one record (i.e., 1.35% of the users are online only for 5 minutes during the whole monitored period). The upper extremity of the box plot identifies the users who were online for the most part of the crawling period, having number of records in the range [4586,9155]. The box plot of our sample came from the statistical distribution shown in Fig. 1(b), which depicts both the frequency distribution



(a) Box plot

(b) Cumulative and frequency distribution



(c) Number of execution of the crawler per day (mm-dd)

Fig. 1. The box plot (a) depicts statistical parameters of our dataset while graph (b) displays statistical distribution of the values. Finally, graph (c) shows the number of times in a day that our crawler has been successfully executed.

(Freq) and the Cumulative Distribution Function (CDF) of the number of records of the monitored users. The plot indicates that 70% of the users spend online short periods of time because we have collected less than 4000 records per user during the entire monitored period, while a small fraction of the users (about 10%) exposes more than 4000 records. In addition, we observed that there are no users who have been online for the entire monitoring period, i.e., having a number of records equals to 9250.

To achieve a deeper understanding of the collected data, we investigated the number of online/offline users over time. Fig. 2 shows the total number of online/offline users, as well as, the fraction of online/offline users. The plot indicates the presence of a cyclic day/night pattern, which is confirmed also by results in [2, 9, 10, 12]. In particular, most of the users are connected during lunchtime and in the evening. In addition, we noticed that users who are offline, i.e., having availability status $S = 0$, are more than the users who are online (i.e., $S = 1$).

3.2 Data preparation

In this section, we investigate the quality of our dataset and we describe how we transform the data in order to prepare them for the prediction task. Fig. 1(c) shows, for each day of the monitored period, the number of times that our

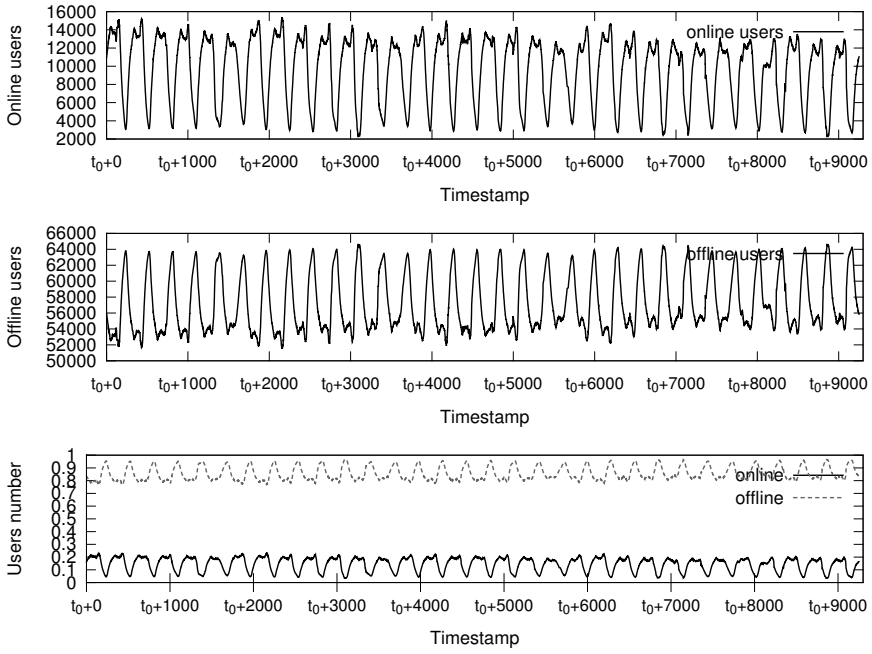


Fig. 2. The total number of online/offline users during the monitored period.

crawler has been executed without failure. Indeed, sometimes the Facebook API had blocked the execution of our application due to the excessive number of requests. We expected a total number of 288 executions per day (except for the first and the last day of the monitored period). The plot in Fig. 1(c) indicates that the execution of our crawler has failed once both on 10 and on 16 of April, while we have 12 failures on 29 of April. Since these missing values can lead to wrong data analysis results, we replaced these missing values with the availability status of users in the same time instants of the previous day (in the case of 16 and 29 of April) or of the consecutive day (in the case of 10 of April).

Since our dataset is a collection of records $< t_i, u >$ which indicate only when user u is online, we perform a transformation phase on the original dataset and we construct new attributes that are necessary for the prediction task. Given a specific time instant t_i, we want to predict the availability status S of a user by exploiting the number of online user's friends and the number of offline user's friends at time t_i. The availability status of the user is a nominal target attribute S, whose value is equal to 1 if the user is online or 0 if the user is offline in the corresponding time instant. For these reasons, we create for each time instant t_i and for each user u registered to our application a record $< onFriends, offFriends, S >$ which indicates the number of online/offline friends of u (onFriends/offFriends) along with the availability status of u at time instant t_i. Table 2 summarizes the set of attributes considered in each instance

Table 2. Description of the attributed obtained from the dataset

Name	Type	Values/Format	Description
onFriends	numeric	integer	number of online friends
offFriends	numeric	integer	number of offline friends
S	nominal	1=online, 0=offline	availability status

and the availability prediction problem consists in predicting the availability status S by exploiting such attributes: $onFriends, offFriends => S(0/1)$.

4 Training and Classification algorithms

We consider several supervised learning algorithms for the classification task and we split the Facebook dataset into two disjoint sets: the training data and test data. In our experiments the training data consist of the first 80% of the data while the remaining 20% is used for testing the learning algorithms. The two sets have empty intersection, i.e., they do not have common records. Tab. 3 summarizes the set of learning algorithms we considered in our experiments, as well as the input parameters. In the following, we briefly summarize these algorithms.

C4.5 Decision Tree Learner [18] is one of the most used methods for classification decision tree. The algorithm built a hierarchical decision tree which is used for classifying the availability status of a user, based on a number of attributes (i.e., the number of online/offline friends, in our case). The target attribute to predict are not used by the DT while the other attributes are used by the tree to routes an unclassified instance towards a leaf node, based on their values. In particular, each intermediate node checks if some condition is met on the value of the attributes and makes decisions about the next child node which must be considered, until a leaf node is reached. Each leaf node has a label that is used to classify the input instance as online or offline. The conditions on internal nodes of the DT are generated by splitting the domain of attributes in two partitions (i.e., using a binary split) and the Gini index is used as a quality measure to calculate such splitting point.

Rep Tree Learner algorithm is provided by Weka Mining Software [20] and it built a regular decision tree with reduced-error pruning by maximizing entropy values.

Random Decision Forests Learner extends the traditional DT Learner by constructing several decision trees [6]. The algorithm selects several random samples from the training data and a decision tree built from each sample. Finally, the resulting decision trees are combined with each other in order to produce a random forest that performs better than the original learners. In particular, the decision trees are joined by using s bootstrap aggregation procedure that assigns weights to average individual decision trees.

Functional Tree Learner [16] combines Logistic Regression [14] and Decision Tree [18] in order to classify instances. In particular, the algorithm built a decision tree where internal nodes are generated by splitting the domain of an attribute in two partitions. Instead, the leaves of the tree embed linear regressions that describe relationships between the number of online/offline friends and the availability status of a user. In particular, the logit function [14] is used to estimate the probability that user is online or offline based on the presence of some characteristics.

Naive Bayes Learner is based on Bayes's theorem [1] which is used to compute the most probable availability status of a user depending on the values of the other attributes. For this purpose, the conditional probability is used and the algorithm assumes that all the attributes are conditionally independent. The training data are used to fit the posterior probability and prior probability distribution while the Laplace corrector is used for estimating the model parameters based on the presence on categorical attribute values. Finally, the generated probabilistic model is used to predict the availability status S of a user by maximizing the conditional probability of S, given the number of online/offline friends of the user.

k-Nearest Neighbor is based on the nearest-neighbor algorithm [8], which classifies the availability status S of a user based on the availability status of the k most similar instances. The underlying algorithm exploits a KD-tree and the Euclidean distance for measuring the similarity between the instances. The prediction is computed by averaging the availability status of the k nearest neighbors. The number of neighbors to be considered is an input parameter and it is fixed to 20. In addition, since K-Nearest Neighbor is affected by the domain of numerical attributes, we decide to scale such domain to similar ranges, i.e., to the [-1,1]).

Probabilistic Neural Network [13] based on the Dynamic Decay Adjustment (DDA) [3] allows to predict availability status of unclassified instances. The network consists of a 4 layers: *i)* the input layer compares the input instances with the training data, *ii)* the pattern layer computes the weighted product of the values and applies a non-linear transformation, *iii)* the summation layer computes the sum of the values by considering both online and offline pattern, finally *iv)* the output layer produces a binary output which correspond to the predicted availability status. The network is trained by exploiting Gaussian function that is tuned by two input parameters, *theta minus* and *theta plus* whose value is fixed by default to 0.2 and 0.4, respectively.

5 Results Validation

We used the test data in order to evaluate the performance of the different learning algorithms. Since the most part of the Facebook's users connect to system only for short periods of time (see Sec. 3) we expect that the number of instances of the dataset having availability status equals to offline are higher than

Table 3. Description of the Learning algorithms used in our experiments

Name	Description
C4.5 Decision Tree Learner (DT)	no pruning, Gini Index
Rep Tree Learner (RDT)	-
Random Decision Forests Learner (RDF)	#Trees=10
Functional Tree Learner (FT)	-
Naive Bayes Learner	Laplace corrector=0
k-Nearest Neighbor (k-NN)	k=20
Probabilistic Neural Network (PNN)	theta minus=0.2, theta plus=0.4

the number of instances with availability status S equals to online. In order to take into account this unbalance we performed an equal size sampling on the test set which allows to under-sample the instance having the most frequent availability status. We plan to investigate more sophisticated techniques for unbalance class distribution (such as [7]) and for the validation of the models (such as cross-validation) as future works.

5.1 Performance measures

In order to compare the accuracy of the learning algorithms used in our experiments we calculated the following quality measures:

- A Confusion Matrix [19] is a table that represents: *i)* the number of instances that have been correctly classified as online (true positive or TP) or as offline (true negative or TN), and *ii)* the number of instances that have been classified as offline when they are online (false positive or FP), and *ii)* the number of instances that have been classified as offline when they are online (false negative or FN).
- The Sensitivity measures the ability of the predictors to correctly classify the availability status of the users (i.e., TP/(TP+FN)).
- The Precision measures the ability of the predictors to correctly identify instances that are online (i.e., TP/(TP+FP)).
- The Specifity measures the ability of the predictors to correctly classify users having availability status equal to offline. It is obtained by calculating TN/(TN+FP).
- The F-measure combines precision and sensitivity by using the harmonic mean and it is used to measure the accuracy of the test.
- The Accuracy is one of the most important measure because it indicates the ability of the predictor to classify instances that are both online or offline. In particular, it obtained by calculating (TP+TN)/(TP+FP+TN+FN).
- The Cohen's kappa is an agreement index that measures the degree of accuracy and reliability of the classification task. Depending on the value of the Cohen's kappa, the index can be interpreted as [15]: *i)* no agreement if $k \in [0, 0.20]$, *ii)* fair agreement if $k \in [0.21, 0.4]$, *iii)* moderate agreement if $k \in [0.41, 0.6]$, *iv)* substantial agreement if $k \in [0.61, 0.8]$, and v) perfect agreement if $k \in [0.81, 1]$.

Table 4. Confusion Matrix of the predictors.

	Confusion Matrix			
Name	**TP**	**FP**	**TN**	**FN**
C4.5 Decision Tree Learner	25919	5621	68620	48322
Rep Tree Learner	23792	5004	69237	50449
Random Decision Forests Learner	25765	5638	68603	48476
Functional Tree Learner	23928	5159	69082	50313
Naive Bayes Learner	7710	4212	70029	66531
k-Nearest Neighbor	25575	5694	68547	48666
Probabilistic Neural Network	24328	4966	69275	49913

– The Area Under Curve (AUC) is a measure that is derived from the Receiver Operating Characteristic Curve and it indicates the ability of the classifier in solving the problem of predicting the availability status of the users. The values of AUC belong to the interval $[0.5, 1]$ where 1 indicates perfect classification without errors while 0.5 corresponds to random classification.

5.2 Results evaluation

The Table 4 shows the confusion matrix, indicating the absolute number of instances which have been correctly classified by each algorithm, as well as the number of incorrectly classified instances. From the columns TP and TN we can easily derive the total number of correct predictions made by each algorithm, as well as the total number of incorrect predictions (i.e., the columns FP and FN). The reader can notice how the total number of correct predictions of the C4.5 Decision Tree Learner outperforms the others. In addition, a significant number of correct predictions is also achieved by the Random Decision Forest Learner, the K-Nearest Neighbor, and the Probabilistic Neural Network. Instead, the Naive Bayes Learner is the worst in terms of the number of incorrect predictions. The quality measures used for evaluating the models can be calculated from the confusion matrix and they are shown in Table 5 for the sake of clarity. As we expected, the sensitivity of the classifier in predicting availability status of online is not very high and it does not exceed 0.35. Indeed, the higher number of records having availability status equals to offline heavily affects the class distribution and reduce the number of instances which are useful for the prediction of the online users. For this reason, the algorithms may fail in predicting the availability status of the users when they are online. Indeed, the specificity measure is very higher for almost all the predictors and it clearly indicates the ability of the predictor in identifying the availability status of the offline users.
However, the predictors show to have higher precision, indicating that the most part of users having availability status equals to online are correctly identified by the predictors. The F-measure summarizes the performance of each predictor for the online case and it shows that predictors based on Decision Tree have the best performance. The last step in our analysis consists in evaluating the

Table 5. Performance of the predictors

Name	Sensitivity	Precision	Specifity	F-measure
C4.5 Decision Tree Learner	**0.349**	0.822	0.924	**0.490**
Rep Tree Learner	0.321	0.826	0.932	0.462
Random Decision Forests Learner	0.347	0.821	0.924	0.488
Functional Tree Learner	0.322	0.823	0.931	0.463
Naive Bayes Learner	0.104	0.647	**0.943**	0.180
k-Nearest Neighbor	0.345	0.818	0.923	0.485
Probabilistic Neural Network	0.328	**0.831**	0.933	0.470

Table 6. Accuracy of the predictors

Name	Accuracy	Cohen's kappa	AUC
C4.5 Decision Tree Learner	**0.637**	**0.273**	0.769
Rep Tree Learner	0.627	0.253	0.763
Random Decision Forests Learner	0.636	0.271	**0.803**
Functional Tree Learner	0.626	0.253	0.645
Naive Bayes Learner	0.524	0.047	0.608
k-Nearest Neighbor	0.634	0.268	0.802
Probabilistic Neural Network	0.630	0.261	0.801

accuracy of the predictors. Table 6 reports the Accuracy, the Cohen's kappa and the AUC obtained by each predictor. The most part of the predictors have an Accuracy value that does not exceed 0.65. The agreement index (Cohen's kappa) is also fine because it is higher than 0.20 (except for the predictor based on Naive Bayes Learner). Finally, the AUC value clearly indicates the C4.5 Decision Tree Learner, Random Decision Forests Learner, K-Nearest Neighbor, and Probabilistic Neural Networks are the most promising algorithms in solving the task of availability prediction of the OSNs' users because they enable an attacker to infer the availability status of a user for at least 60% of the time.

6 Conclusion and Future works

In this paper, we uncovered a number of interesting results related to the problem of predicting the availability status (online/offline) of OSNs' users. In particular, we showed that the availability status of an individual is partly affected by those of their friends and we found that Decision Tree Learner and Random Decision Forest Learner have the best accuracy in predicting the availability status. In addition, we observed that k-Nearest Neighbor and Probabilistic Neural Network are suitable models for predicting the user's availability.
As future work, we would like to investigate different configuration parameters of the considered learning algorithms. In addition, we plan to deal the class's unbalance problem by exploiting advanced techniques, such as Synthetic Minority Over-sampling [7]. Another interesting aspects is to boost the performance of the predictors by considering other attributes useful for predicting the availability

status of the users, such as the timestamp and sessions length. Finally, we plan to exploit association analysis to identify relationships between the availability status of users and the identities of the users' friends who are online/offline.

References

1. Baron, M.: Probability and statistics for computer scientists. CRC Press (2013)
2. Benevenuto, F., Rodrigues, T., Cha, M., Almeida, V.: Characterizing user behavior in online social networks. In: Proceed. of the 9th ACM SIGCOMM conf. on Internet measurement. pp. 49–62. ACM (2009)
3. Berthold, M.R., Diamond, J.: Constructive training of probabilistic neural networks. Neurocomputing 19(1), 167–183 (1998)
4. Blond, S.L., Fessant, F.L., Merrer, E.L.: Choosing partners based on availability in p2p networks. ACM Trans. on Autonomous and Adaptive Systems (TAAS) 7(2), 25 (2012)
5. Boutet, A., Kermarrec, A.M., Le Merrer, E., Van Kempen, A.: On the impact of users availability in osns. In: Proceed. of the Fifth Workshop on Social Network Systems. p. 4. ACM (2012)
6. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357 (2002)
8. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Tran. on information theory 13(1), 21–27 (1967)
9. De Salve, A., Dondio, M., Guidi, B., Ricci, L.: The impact of user's availability on on-line ego networks: a facebook analysis. Computer Communications 73, 211–218 (2016)
10. De Salve, A., Guidi, B., Mori, P., Ricci, L., Ambriola, V.: Privacy and temporal aware allocation of data in decentralized online social networks. In: International Conf. on Green, Pervasive, and Cloud Computing. pp. 237–251. Springer (2017)
11. Dell'Amico, M., Michiardi, P., Roudier, Y.: Back to the future: On predicting user uptime. CoRR abs/1010.0626 (2010), http://arxiv.org/abs/1010.0626
12. Golder, S.A., Wilkinson, D.M., Huberman, B.A.: Rhythms of social interaction: Messaging within a massive online network. Communities and technologies 2007 pp. 41–66 (2007)
13. Haykin, S.S., Haykin, S.S., Haykin, S.S., Haykin, S.S.: Neural networks and learning machines, vol. 3. Pearson Upper Saddle River, NJ, USA: (2009)
14. Hilbe, J.M.: Logistic regression. In: International Encyclopedia of Statistical Science, pp. 755–758. Springer (2011)
15. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. biometrics pp. 159–174 (1977)
16. Landwehr, N., Hall, M., Frank, E.: Logistic model trees. Machine learning 59(1-2), 161–205 (2005)
17. Mickens, J.W., Noble, B.D.: Exploiting availability prediction in distributed systems. Ann Arbor 1001, 48103 (2006)
18. Quinlan, J.R.: C4. 5: programs for machine learning. Elsevier (2014)
19. Stehman, S.V.: Selecting and interpreting measures of thematic classification accuracy. Remote sensing of Environment 62(1), 77–89 (1997)
20. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2016)