Human Computer Interaction Activity Based User Identification

Tongda Zhang, Xiao Sun, Yueting Chai, and Hamid Aghajan

***Abstract*—In this paper, an intensive study is performed on the dataset that contains people's activity records of computer usage, and a multi-feature based user identification algorithm is developed. It is shown that features generated from user's program/process start up history and website access records are extremely effective to differentiate people from each other. The proposed algorithm exploits this fact for identifying user by matching user's current behavior features with the history feature library. The experimental results show that the proposed multi-feature based algorithm can achieve 91.2% user identification accuracy.**

***Index Terms*—User identification, computer using behavior, feature matching, feature combination.**

I. Introduction

A. Background

As the online data becomes cheaper to collect and more readily accessible in recent years, Internet based data, especially social network data, has been thoroughly studied to understand online individual users and community [1], [2]. However, because of the difficulty of data collection, people's behavior and activities of computer usage has rarely been recorded and researched.

Although the Internet user's behavior on a website can be used to predict future online conduct or build a recommendation system, it only contains one aspect of the user's behavior that he or she choose to show on a specific website. The link prediction method [3] and the edge sign prediction algorithm [4] are both examples of such limitation. Unlike the data from a social network, the behavior records collected on a user's computer (keep track of the program opened history, the website visit history etc.) may contain more comprehensive information about the user.

It has already been shown that some of people's behaviors are repetitive and following some basic patterns [5], [6]. With rich information contained in users' behavior records of computer operating, we can expect to know more about the uniqueness of each user, and can use that information to identify users from each other.

B. Dataset

The datasets we used in this paper are log files of people's human computer interaction activities. To be more specific, a daemon runs in the background of the volunteer's computer and keeps records of every user interaction event with the computer. Those events include booting the computer, opening a new process, changing the current focused window, visiting a website and so on. For each event, the daemon traces the start and end timestamps, the name of the process if any process is involved, the URL if it is a website visiting event and so forth.

The dataset is provided by China Internet Network Information Center (CNNIC) [7] and is collected by client software [8]. It has 7 days' observation of 775 volunteers with over 2 million event records in total. To protect the privacy of the volunteers, the name of each user is replaced by its hashed value. So the actual identity of each volunteer cannot be retrieved.

The event record in the dataset is well organized in a format that the time stamp, event name, and other detail information are separated with special characters:

Time stamp => event marker => [detail field name] value

And each event takes a separate line. So no extra data clean work is needed to extract the event information.

C. Our Work

To begin with, we generate five different features as descriptors of user's daily computer using behaviors. These features are used to capture the unique and repetitive patterns of each people.

In order to compare the importance of each feature, we then build an evaluation scheme. It quantifies the features' effectiveness of distinguishing people from each other. Experiments are performed on the dataset with each feature. It turns out that three features are extremely effective while the other two are almost useless.

After that, as an approach of identifying user, the single feature based user identification algorithm is proposed and tested on the dataset. It is followed by a further analysis of the algorithm's performance with tweaking its output selection mechanism.

Finally, we improve the user identification accuracy by using multi-feature based algorithm, which combines the information from different features. The final identification precision we achieved is far better than the base line method and single-feature based algorithm.

D. Paper Structure

The rest of this paper is organized as follows. Section II Part A describes the five features we developed in this paper, and elaborates their practical meaning and the intuition behind them; Part B and C define the similarity functions for

Manuscript received February 18, 2014; revised May 12, 2014.

Tongda Zhang is with the Electrical Engineering Department, Stanford University, Stanford, CA 94305 USA (e-mail: tdzhang@stanford.edu).

Xiao Sun is with National Engineering Laboratory for E-Commerce Technology, Tsinghua University, Beijing, China; DNSLAB, China Internet Network Information Center, Beijing 100190, China (e-mail: sunx11@mails.tsinghua.edu.cn).

Yueting Chai is with the National Engineering Laboratory for E-Commerce Technology, Tsinghua University, Beijing, China (e-mail: chaiyt@mail.tsinghua.edu.cn).

Hamid Aghajan is with the Department of Electrical Engineering, Stanford University, and the Department of Telecommunication and Informatics in Gent University, Belgium (e-mail: hamid@icdsc.org).

DOI: 10.7763/IJMLC.2014.V4.436

each feature and describe the evaluation experiment we used; Part D performs some basic analyses on each feature. Section III develops the single feature based and multi-feature based user identification algorithms. It also compares the result of each method. In the last part, some conclusions are drawn based on experimental results, and several future directions are pointed out.

II. Feature Selection

In this section, we characterize user's daily behavior and activities on the computer into several features. And some analyses regarding the effectiveness and uniqueness of each proposed feature are performed.

A. Features and Intuition

Intuitively, many of people's daily behaviors are repetitive and usually follow similar routines. For examples, college students may maintain their weekly schedule during a semester according to the courses they enrolled and the activities of clubs and organizations they participated in; employees may commute between the company and their homes follow the same route every day.

Those routines and customs that people follow, on the other hand, may be different from person to person. Because different people may have different life styles, dissimilar social circles and their unique living habits. In other words, different daily routines may include information that can be used to distinguish people from each other.

Similar intuition goes for people's behaviors of using a computer. For instances, the time when a person turn on the computer and the total time the user spent on computer in a day reflect his or her daily schedule; different programs opened by a user indicate the thing that person (entertaining, programming, working and so on); the visited websites during a day show us the user's internet behavior custom. All these activities can be different from person to person.

Therefore, we have chosen five features that can be extracted from the dataset as candidates to represent each person's unique computer-using behavior. The details of each feature are described in table I. Where u_{ij} represents the behavior data of user i in jth observing day, $i = 1,2,\dots,775$ is the index of different volunteers, and $j = 1,2,\dots,7$ is the index of day number. For example, $Start(u_{ij})$ is the time when user i turned on the computer on jth observing day.

For the simplicity of expression, we will refer feature $Start(u_{ij})$, $Duration(u_{ij})$, $PSetEarly(u_{ij})$, $PSetStable(u_{ij})$ or $WebSet(u_{ij})$ as feature 1, feature 2, feature 3, feature 4 and feature 5 respectively throughout the rest of the paper. And $Start(\cdot)$, $Duration(\cdot)$, $PSetEarly(\cdot)$, $PSetStable(\cdot)$, $WebSet(\cdot)$ are feature functions used for generating each feature.

TABLE I: Feature Descriptions

Name	Description
$Start(u_{ij})$	The time user opens the computer.
$Duration(u_{ij})$	Total time that user spent on computer.
$PSetEarly(u_{ij})$	The set of program that user opened in the first 10 minutes.
$PSetStable(u_{ij})$	The set of programs user opened/visited after 10 minutes.
$WebSet(u_{ij})$	The set of website URLs user visited.

B. Definition of Similarity

Similarity score functions need to be defined in order to evaluate the effectiveness and uniqueness of each feature defined in the previous part. Similarity score functions are used to quantify two features values similarity. It can compare a feature of a user in one day with his or her own feature in the other day, or with another user's feature value. The value that similarity function returned should be higher if the compared feature values are similar, lower if they are very different. Here are the two similarity score functions we are using:

$$SimilarValue(f_1, f_2) = \frac{1}{|f_1 - f_2| + \varepsilon} \tag{1}$$

$$SimilarSet(f_1, f_2) = \frac{|f_1 \cap f_2| + \varepsilon}{|f_1 \cup f_2| + \varepsilon} \tag{2}$$

Eq. (1) is used for calculating similarity score of feature 1 or feature 2. When we compare the time that user i opening the computer on day j with the time that user m opening the computer on day n, namely $Start(u_{ij})$ and $Start(u_{mn})$, $|Start(u_{ij}) - Start(u_{mn})|$ gives us the absolute time difference between them. And Eq. (1) is the reciprocal of such difference, which measures the extent of similarity between the feature values. The same principal holds true for feature 2.

Eq. (2) is the equation that calculate Jaccard similarity coefficient of set f_1 and f_2, with a small positive number ϵ added on both the numerator and denominator. It is used for feature 3, 4 and 5. Because these last three features are sets of items, and the Jaccard similarity [9] coefficient is known as a statistic measure of the similarity of sets. The small positive number is used to avoid the "divide by zero error" that will happen if both f_1 and f_2 are empty sets.

C. Test Set and Feature Library

With the definition of above five features and similarity score functions, here we define our effectiveness and uniqueness evaluation experiment to be "identifying a user using his or her behavior feature in one day after observing all users' behavior data in the past".

To be specific, the first six days' data is regarded as our observation and the data in the last day is treated as our test set. Then, the experiment can be restated as follows:

Knowing observed feature values in the first 6 days, which is

$$\{User\ i: Feature(u_{i1}),\ Feature(u_{i2}), \dots, Feature(u_{i7})\ \}_{i=1,2,\dots,775}$$

And given a test feature value $v_f^{(k)}$ in the 7th day, how accurate can we identify that it is the user k who generated that feature value, in other works, $v_f^{(k)} = Feature(u_{k7})$. where $Feature(\cdot)$ is the feature function refers to either $Start(\cdot)$, $Duration(\cdot)$, $PSetEarly(\cdot)$, $PSetStable(\cdot)$ or $WebSet(\cdot)$, according to which feature we are studying.

For a specific feature function $Feature(\cdot)$ and a user i, we define the user i's feature's library as:

$$\{User\ i: Feature(u_{i1}),\ Feature(u_{i2}), \dots, Feature(u_{i6})\ \} \tag{3}$$

For a certain feature function $Feature(\cdot)$, the test set for that corresponding feature is defined as:

$$\left\{v_f^{(i)} = Feature(u_{i7})\right\}_{i=1,2,\dots,775} \qquad (4)$$

D. Feature Analysis

Before getting into the evaluation experiment for each feature, the way of measuring feature's effectiveness needs to be defined.

We propose to use the effectiveness rank as the measure criteria. For a specific feature function Feature (·) and a certain test feature value $v_f^{(k)}$, the effectiveness rank is defined as:

$$Effect\left(Feature(\cdot), v_f^{(k)}\right) = \sum_{i=1}^{775} \mathbf{1}\{bestMatch(L_i, v_f^{(k)}) \geq bestMatch(L_k, v_f^{(k)})\} \quad (5)$$

where $\mathbf{1}\{\cdot\}$ is indicator function stated as:

$$\mathbf{1}\{\text{statement}\} = \begin{cases} 1 & if\ statement\ is\ true \\ 0 & if\ statement\ is\ false \end{cases} \qquad (6)$$

And L_i $(i = 1,\ 2,\ \dots,\ 775)$, which is defined in Eq.(3), is the feature library of user i ,. Function $bestMatch(\cdot,\cdot)$, which is described in Eq.(7), is the highest score among all the similarity scores between the test feature $v_f^{(k)}$ and each feature value in user i's feature library.

$$bestMatch(L_i, v_f^{(k)}) = \begin{cases} \max\limits_{v \in L_i} \left\{SimilarValue(v, v_f^{(k)})\right\} & \text{for feature 1, 2} \\ \max\limits_{v \in L_i} \left\{SimilarSet(v, v_f^{(k)})\right\} & \text{for feature 3,4,5} \end{cases} \qquad (7)$$

Therefore, the effectiveness rank for user k's test feature $v_f^{(k)}$ defined in Eq.(5) is the number of users whose feature library is more similar with test feature value $v_f^{(k)}$ rather than the user k's own feature library.

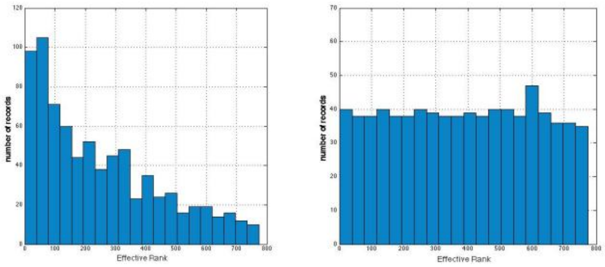
The definition would be more intuitive if we think the above process as Internet searching. The feature $v_f^{(k)}$ is the searching keyword. And the search result returned is a list of candidate users, which is in a descending order of their similarity score. Under this setting, the effectiveness rank is the minimum number of users that the search result need to return in order to include the actual result---user k. Therefore, effectiveness rank can be used to measure identification accuracy. The smaller the effectiveness rank is, the more precise the identification will be.

For feature 1, we go through the whole test set Eq. (4) with feature function $Feature(\cdot) = Start(\cdot)$, calculate the effectiveness rank of each test feature using Eq. (5). The result is an effectiveness rank set:

$$E_{Start} = \left\{e_f^{(k)}\right\}_{i=1,2,\dots,775} \qquad (8)$$

where $e_f^{(k)} = Effect\left(Feature(\cdot), v_f^{(k)}\right)$. The histogram of the rank set E_{Start} is shown in Fig. 1 (a).

Using similar approaches, we can also get the rank sets $E_{Duration}$, $E_{PSetEarly}$, $E_{PSetStable}$ and E_{WebSet} for the other 4 features. The histograms of them are presented in Fig. 1(b), Fig. 2(a), Fig. 2(b) and Fig. 2(c), respectively.

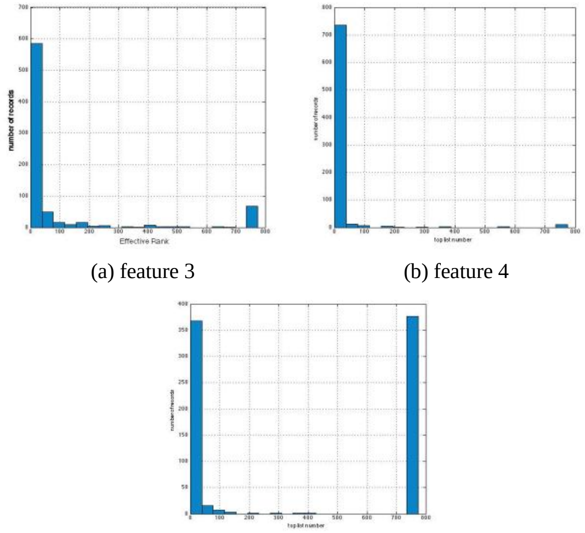


(a) feature 1

(b) feature 2

Fig. 1. Histogram of effectiveness rank set.

The distribution of the effective ranks for feature 1 can be seen from Fig. 1(a). The number of records is almost monotone decreasing when the rank's value becomes bigger. It means that a user's feature 1 value is more similar with his (or her) own feature library rather than other users'. However, almost half of the test feature values fall into the bins whose effective rank value is bigger than 150. Using our Internet search metaphor, it means that more than half of the searches need to return more than 150 matching users as result in order to include the target user who actually generated that test feature value. Therefore, feature 1 contains information useful to differentiate users, but it is not very effective and unique.



(a) feature 3

(b) feature 4

(c) feature 5

Fig. 2. Histogram of effectiveness rank set.

As Seen in Fig. 1(b), the distribution of the effectiveness ranks for feature 2 is very similar to uniform distribution. A feature value of one user is equally possible to be similar with feature values of other users. Therefore, feature 2 can hardly be used to distinguish people from each other.

In Fig. 2 (a) and (b), we can see that the feature 3 and feature 4 have very similar distributions of the their effectiveness ranks. In both histograms, most of the test feature values fall into the first bin, which means the identification result is really accurate. Thus, both features are very effective.

As we can see, the distribution of the effective ranks for feature 5, which is represented in Fig. 2(c), is so bipolarized that almost half of the test feature values fall into the first bin and another half fall into the last bin. In other words, feature 5 either can distinguish different users accurately or is unable to tell different people at all. Further analysis shows this is caused by some empty test feature values that $v_f^{(k)} = \phi$. Because an empty set will make all similarity scores equal to zeros, which makes all the users look the same in term of feature 5.

III. User Identification

With the set of features in hand, we develop single feature based user identification method and make comparisons of the identification result by using different features. Then, in order to achieve higher user identification accuracy, we improve the identification method by combining the information from different features.

A. Single Feature Based Method

The workflow for the single feature based method is described in Fig. 3. For each test feature value from the test set, we compare it with every user's feature library using $bestMatch(\cdot,\cdot)$ function. Then we sort the list of users in the descending order of their similarity score and pick the top one (the best match user) as our identification result.

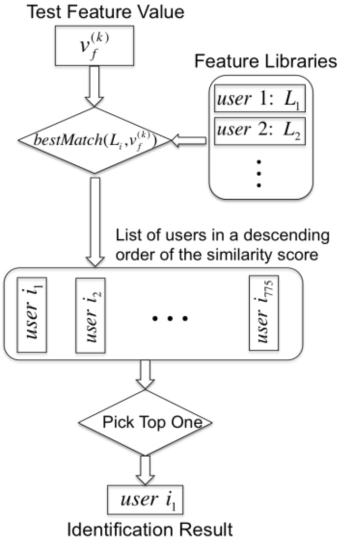


Fig. 3. Single Feature based algorithm.

TABLE II: Single Feature Based Identification Precision

	Total	Empty Cases Excluded
$\boldsymbol{Start(u_{ij})}$	0.52%	0.52%
$\boldsymbol{Duration(u_{ij})}$	0.52%	0.52%
$\boldsymbol{PSetEarly(u_{ij})}$	36.1%	39.6%
$\boldsymbol{PSetStable(u_{ij})}$	74.3%	75.3%
$\boldsymbol{WebSet(u_{ij})}$	30.8%	60.1%

We perform this method on the whole test set using each feature. The identification accuracy result is shown in the "Total" column of Table II.

We mentioned earlier in Section II Part D that for feature 3,4 and 5, some empty test feature values (where $v_f^{(k)} = \phi$) would cause the feature lost its ability to identify people. Under such cases, the proposed single feature based method will degenerate into random guess method. Therefore, we exclude these pathological test cases altogether and redo the user identification experiments. The results are shown in the "Empty Casas Excluded" column of Table II.

As seen in Table II, the precision using feature 1 or feature 2 is only slightly better than using the random guess (0.13%). This result is consist with our earlier feature analysis in Section II Part D.

For feature 3, 4 and 5, the precision of user identification is much higher than random guess. This is especially true for feature 4, whose identification precision reaches 75.3%.

Considering the significant accuracy difference between the result of using feature 1, 2 and the result of using feature 3,4 and 5, we will only consider using feature 3, 4, 5 in the following part of this paper.

B. Further Analysis

Before we jump into the multi-feature based method. Let's take a further look at the last 3 features.

At the last step of single feature based algorithm in Fig. 3, instead of picking top one as the identification result, we use either top 1, top 2, top 4 or top 8 best matching candidate users as our result set. For each cases, the probability that the result set includes the target user is shown is Table III.

TABLE III: Precision of Returning Multiple Candidates

	Top1	Top2	Top4	Top8
$\boldsymbol{PSetEarly(u_{ij})}$	39.6%	45.9%	54.5%	63.8%
$\boldsymbol{PSetStable(u_{ij})}$	75.3%	81.2%	85.6%	89.2%
$\boldsymbol{WebSet(u_{ij})}$	60.1%	66.6%	73.6	80.7%

The more candidate users it return, the more likely that result set will cover the actual user. Table III shows that by using only feature 4, we can narrow down the target user into 8 candidates from all 775 users with almost 90% accuracy.

C. Multi-Feature Based Method

In order to achieve better accuracy, a reasonable improvement can be made on top of the single feature based method is to combine different features together.

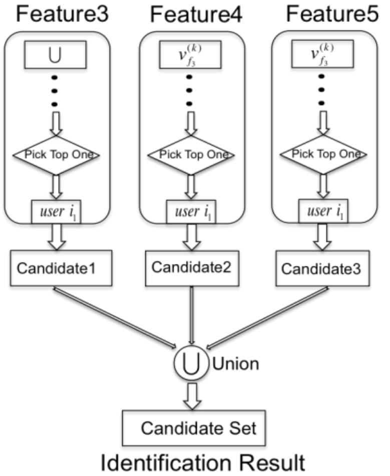


Fig. 4. Multi-feature based algorithm.

Fig. 4 shows the workflow of our proposed multi-feature based user identification method, which uses the last 3 features together. We first perform single feature based algorithm to find the top 1 candidate using feature 3, 4 and 5 separately. Then we take a union of the 3 candidates as our result set. If all three candidates agree with each other, the result set will only include one user; if only two candidates agree with each other, the result becomes a 2 element result set; otherwise, this method will return a set with 3 element. Table IV shows the identification accuracy results of all these three cases separately. And the “Total” column gives the total precision combining all three cases.

TABLE IV: MULTI-FEATURE BASED IDENTIFICATION PRECISION

	Single return	2 element return	3 element return	Total
Multi-Feature method	96.3%	100%	72.7%	91.2%

From Table IV, we can see that if all three features agree with each other or at least 2 of them agree with each other, the identification accuracy can achieve more than 96%. Even when the three features give 3 different candidates, we still can have 72.7% confidence that the result set includes the target user. Therefore, by using this multi-feature based method, we end up with 91.2% accuracy to narrow down the target user within 3 candidates. This is much higher than random guess and the single feature based method. The comparison of all these user identification methods is shown in Fig. 5.

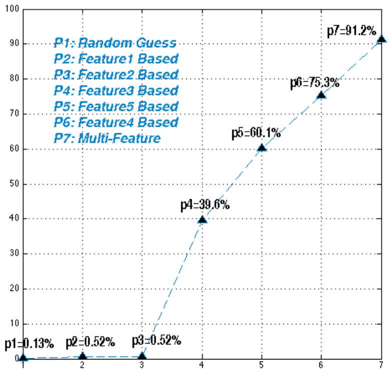


Fig. 5. Precision of different identification methods.

IV. CONCLUSION

In this paper, we generated several features that summarize people’s daily computer using behavior. Then we developed user identification algorithms that use those features to identify users from each other. By eliminating unimportant features and combining the information from the rest features, we successfully improve the performance of our user identification algorithm to 91.2% accuracy.

There are a lot of further directions can be explored starting from this paper. First, more behavior features can be generated from the dataset. Unlike current features that only contain static information, new features may include more dynamic aspects of people’s behavior, such as behavior sequence, state transfer probability and so forth. Second, an online identification algorithm can be developed to deal with long-term observation and identification tasks. For long-term purpose, the feature data could be clustered and summarized when the new data is coming. Another possible future direction is that, instead of identifying user, future activity can be predicted based on user’s history behavior.

ACKNOWLEDGMENTS

We would like to acknowledge the dataset and technical support for our work provided by NELECT (National Engineering Laboratory for E-commerce Technology, Tsinghua Univ.) and DNSLAB of China Internet Network Information Center.

REFERENCES

[1] J. Vosecky, H. Dan, and V. Y. Shen, “User identification across multiple social networks,” *Networked Digital Technologies*, 2009.

[2] L. Garton, C. Haythornthwaite and B. Wellman, “Studying online social networks,” *Journal of Computer-Mediated Communication*, vol. 3, 1997.

[3] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *J. Am. Soc. Inf. Sci.*, vol. 58, pp. 1019–1031, 2007.

[4] T. Zhang, H. Jiang, Z. Bao, and Y. Zhang, “Characterization and edge sign prediction in signed networks,” *Journal of Industrial and Intelligent Information*, vol. 1, no. 1, pp. 19-24, 2013.

[5] J. Froehlich and J. Krumm, “Route prediction from trip, observations,” SAE Technical Paper, 2008-01-0201, 2008

[6] K. Laasonen, “Route prediction from cellular data,” in *Proc. the Workshop on Context Awareness for Proactive Systems*, 2005.

[7] China Internet Network Information Center (CNNIC). [Online]. Available: http://www.cnnic.net.cn

[8] China Internet Network Data Platform Client Software. [Online]. Available: http://www.cnidp.cn/

[9] M. Levandowsky and D. Winter, “Distance between sets,” *Nature*, vol. 234, pp. 34-35, 1971.

Tongda Zhang was born in China, 1989. He graduated with the outstanding graduate award from Tsinghua University in 2011, and earned his bachelor’s degree of automation. In 2012, he received his Master’s degree in electrical engineering from Stanford University. He is currently a Ph.D candidate in electrical engineering at Stanford University with a focus of Human behavior understanding, data mining and machine learning.

Xiao Sun was born in China, 1988. He graduated from Tsinghua University in 2011, and earned his bachelor’s degree of Automation. He is currently a Ph.D. candidate in the Department of Automation at Tsinghua University with an interest in e-commerce and optimization theory.

Yueting Chai was born in China, 1964. He earned his doctor’s degree of automation at Tsinghua University in 1991 and served as deputy chief engineer of National CIMS Engineering Research Center since 1996. Professor Chai is now the director of National Engineering Laboratory for E-Business Technology with a focus on e-commerce and modern logistics.

Hamid Aghajan received his MS and PhD degrees from Stanford University. He is the director of the Ambient Intelligence Research (AIR) Lab at Stanford University. He is the Co-Editor-in-Chief of "Journal of Ambient Intelligence and Smart Environments", and has served as a guest editor for IJCV, IEEE Trans. on Multimedia, CVIU, IEEE JETCAS, and IEEE J-STSP. Hamid was a general chair of ACM/IEEE ICDSC 2008, AMI 2011, ICMI 2012, and program chair of ICDSC 2007. He has organized workshops, special sessions, or tutorials at ECCV, ACM MM, CVPR, ICCV, ICMI, FG, ECAI, EI, and ICASSP. His research topics include behavior modeling for detection of anomaly or lifestyle shift in elderly care, occupancy modeling of smart buildings for resource efficiency, improving office worker's well-being through personalized ergonomic recommendations, analysis of meetings, and avatar-based social interactions.