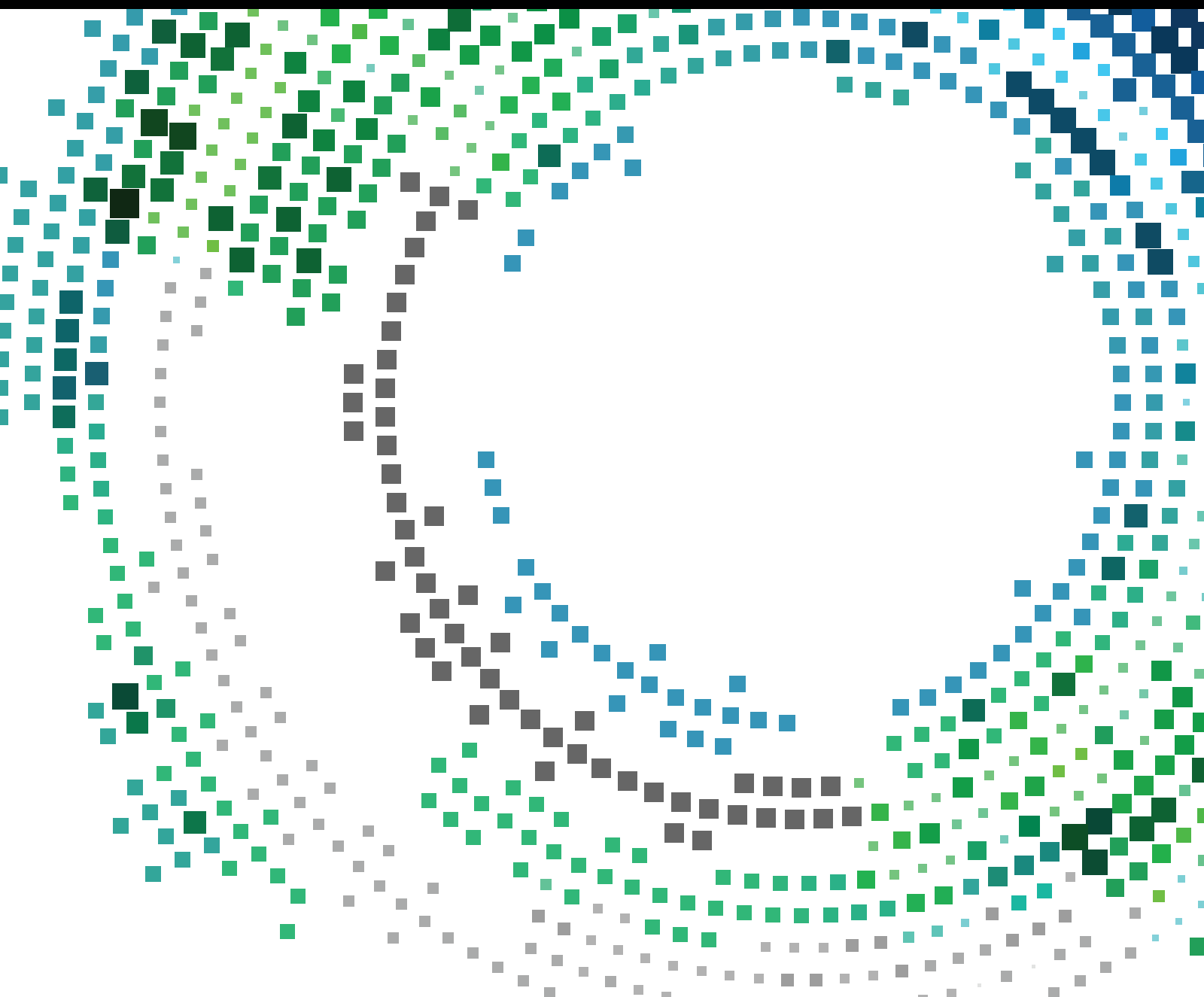
Mobile Information Systems

Advances in Personalized Mobile Services

Lead Guest Editor: Fabio Gasparetti
Guest Editors: Federica Cena, Damianos Gavalas, Shuk Y. Ho, Bin Liu, and Dingqi Yang



Advances in Personalized Mobile Services

Mobile Information Systems

Advances in Personalized Mobile Services

Lead Guest Editor: Fabio Gasparetti
Guest Editors: Federica Cena, Damianos Gavalas, Shuk Y. Ho, Bin Liu, and Dingqi Yang

Editorial Board

Contents

Hindawi
Mobile Information Systems
Volume 2018, Article ID 6047696, 2 pages
https://doi.org/10.1155/2018/6047696

Editorial
Advances in Personalized Mobile Services

Federica Cena,[1] Fabio Gasparetti,[2] Damianos Gavalas,[3] Shuk Y. Ho,[4] Bin Liu,[5] and Dingqi Yang[6]

[1]*University of Turin, 10124 Turin, Italy*
[2]*University of Roma Tre, 00154 Rome, Italy*
[3]*University of the Aegean, Syros, Greece*
[4]*The Australian National University, Canberra, ACT 0200, Australia*
[5]*IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA*
[6]*University of Fribourg, Fribourg, Switzerland*

Correspondence should be addressed to Fabio Gasparetti; gaspare@dia.uniroma3.it

Received 17 May 2018; Accepted 20 May 2018; Published 26 June 2018

A large variety of user interaction with products, informational sources, or more general interactive media happens through mobile devices connected to Internet. However, desktop is still important for daytime at work audiences, and smart devices (such as tablets and smartphones) dominate the time that people spend connected to the web.

Personalization is a desired functionality for applications within mobile environments. It provides means of fulfilling users' needs more effectively and efficiently and consequently increasing users' satisfaction, overtaking the traditional one-size-fits-all paradigm. It is considered a fundamental feature for both users who are receiving services and for service providers who want to target their services to each individual, but different conceptual and technical challenges must be faced to achieve adequate adaptivity.

For personalization to be successful, information about the specific users is necessary to understand real user needs. Three principal sources of data about the user are usually considered: contextual information (e.g., geolocation and presence of other people), representations of user specific attributes, such as interests and needs, and social signals.

The motivation behind this special issue is to solicit cutting-edge research relevant to personalized mobile services, with significant advances, carry out innovative explorations, and establish foundations for further research. The special issue has attracted 12 submissions. Following a rigorous review process (including a second review round), 5 outstanding papers (acceptance rate 41%) have been finally selected for inclusion in the special issue. The accepted papers cover a wide range of research subjects in the broader area of mobile services, such as recommender systems, routing protocols, task classification, user authentication, and notification systems.

The paper "Mobile Personalized Service Recommender Model Based on Sentiment Analysis and Privacy Concern" by Liang Xiao, Feipeng Guo, and Qibei Lu proposes a recommender model for mobile services based on the sentiment analysis, taking into account privacy considerations. The article, firstly, introduces a sentiment analysis algorithm based on sentiment vocabulary ontology and then clusters the users based on sentiment tendency. Then, a measurement algorithm is proposed, which integrates personality traits with privacy preference intensity, and then clusters the users based on personality traits. Last, hybrid collaborative filtering recommendation is performed by combining sentiment analysis with privacy concerns. The effectiveness of the proposed method is validated through a series of experiments using practical application datasets.

The paper "Automatic Task Classification via Support Vector Machine and Crowdsourcing" by Hyungsik Shin and Jeongyeup Paek investigates a task classification problem in personal assistant systems and proposes an automatic task classification approach. More precisely, by collecting a dataset of task commands using crowdsourcing via Amazon Mechanical Turk, they use SVM to classify a task command into one of the 32 predefined task types. In

addition, an architecture of integrating the proposed approach with personal assistant systems is also suggested.

The paper "EpSoc: Social-Based Epidemic-Based Routing Protocol in Opportunistic Mobile Social Network" by Halikul Lenando and Mohamad Alrfaay introduces an epidemic routing technique that takes advantage of specific characteristics extracted from social networks. Epidemic routing is a flooding-based approach in which nodes continuously replicate and transmit messages to known contacts that do not already receive a copy of the message. This novel approach shows advantages in terms of the delivery ratio and latency on a real-world dataset, which includes traces of Bluetooth sightings by groups of users carrying mobile devices.

Sandeep Gupta, Attaullah Buriro, and Bruno Crispo in their paper "Demystifying Authentication Concepts in Smartphones: Ways and Types to Secure Access" provide an extensive descriptive survey on security and privacy threats to the user's personal data stored in mobile devices, with particular relevance to user authentication. The survey covers also important usability issues related to the different techniques in the literature, which may influence the decision of the adoption of one specific approach in large-scale scenarios.

Push notifications provide convenience and value to mobile app users. For instance, users can receive sports scores and news right on their lock screen and utility messages like traffic, weather, and ski snow reports. But notifications may also be considered a big distraction to users during their everyday activities. Jemin Lee et al. in their paper "Reducing Smartwatch Users' Distraction with Convolutional Neural Network" propose an AI-based notification management system for smartphone and smartwatch users based on Convolutional Neural Networks. By analyzing of a large real-world corpus, the authors prove the efficacy of the approach in the scenario of a binary classification of each notification into wanted/unwanted classes.

We do hope that this special issue will be of considerable interest to the MIS audience, highlighting state-of-the-art trends, methodologies, and applications in personalized mobile services.

Acknowledgments

We would like to sincerely thank the authors of all the submitted papers for considering our special issue and the MIS as a potential publication venue for their research results. We would also like to especially thank the authors of the accepted papers for their effort in revising and improving their work—occasionally, several times—in response to reviewers' comments. In addition, we would like to thank the anonymous reviewers for doing an excellent job in reviewing the submitted papers and making this special issue possible. Last but not least, we take this opportunity for thanking the journal EiC and the Editorial Board for giving us the opportunity to organize this special issue, which we sincerely believe provides a fresh, relevant, and useful overview of ongoing research in the multifaceted area of smart cities.

Federica Cena
Fabio Gasparetti
Damianos Gavalas
Shuk Y. Ho
Bin Liu
Dingqi Yang

Hindawi
Mobile Information Systems
Volume 2018, Article ID 8071251, 13 pages
https://doi.org/10.1155/2018/8071251

Research Article

Mobile Personalized Service Recommender Model Based on Sentiment Analysis and Privacy Concern

Liang Xiao,[1,2] Fei-Peng Guo ,[1,2] and Qi-Bei Lu[3]

[1]*School of Management and E-Business, Zhejiang Gongshang University, Hangzhou 310018, China*
[2]*Modern Business Research Center, Zhejiang Gongshang University, Hangzhou 310018, China*
[3]*School of Science and Technology, Zhejiang International Studies University, Hangzhou 310023, China*

Correspondence should be addressed to Fei-Peng Guo; guofp@hotmail.com

Received 7 December 2017; Revised 23 February 2018; Accepted 19 April 2018; Published 3 June 2018

Academic Editor: Damianos Gavalas

The existing mobile personalized service (MPS) gives little consideration to users' privacy. In order to address this issue and some other shortcomings, the paper proposes a MPS recommender model for item recommendation based on sentiment analysis and privacy concern. First, the paper puts forward sentiment analysis algorithm based on sentiment vocabulary ontology and then clusters the users based on sentiment tendency. Second, the paper proposes a measurement algorithm, which integrates personality traits with privacy preference intensity, and then clusters the users based on personality traits. Third, this paper achieves a hybrid collaborative filtering recommendation by combining sentiment analysis with privacy concern. Experiments show that this model can effectively solve the problem of MPS data sparseness and cold start. More importantly, a combination of subjective privacy concern and objective recommendation technology can reduce the influence of users' privacy concerns on their acceptance of MPS.

1. Introduction

With the constant development of personalized recommendation technology and its wide use in mobile commerce, mobile recommender system crops up [1, 2]. It provides users with accurate and real-time mobile personalized service (hereafter abbreviated as MPS) [3, 4]. At the same time, users from online platforms like Weibo and Twitter generate an abundance of content (UGC). Platforms have become the main way for users to express their feelings and share information. Consequently, more and more online reviews of online purchases and transaction services emerge [5]. Yenter and Verma [6] said that the opinions given by online users on products and services contain user's emotional tendency. Mobile personalized recommender system can mine implicit interest of users in online reviews, which can fully provide user's preferences and sentiment polarity over attributes, functions, and experiences of products and services [7]. Since the information is stored in the form of behavior logs, tracks, and transaction data in network, opinion mining technology is required to extract and analyze user's sentiment tendency and business knowledge hidden in reviews, drawing many mobile commerce service providers' attention [8, 9]. Therefore, it is a hot topic to extract information about user's sentiment tendency and implicit preference on the basis of text opinion mining to assist personalized recommender system and provide quality mobile personalized service.

Pang and Lee [10] defined opinion mining as a process of analyzing, disposing, concluding, and reasoning on subjective texts with sentiment. However, some of the users' privacy information, such as location and preference, is often exposed through online reviews mining. The privacy leak in MPS is a big issue, jeopardizing the private life of users [11]. Users' privacy awareness and attention paid to the risk of privacy disclosure are gradually increasing [12]. Although many commercial websites implement online privacy policy mechanism, problems exist as follows: (1) the MPS providers unilaterally set up privacy policies but ignore privacy preferences of individuals. Users are only given two choices, either refusing to use the service or passively accepting all privacy policies in order to obtain the service.

They cannot subjectively choose the disclosed type of privacy information [13]. (2) Some researches start to use history preference data of mobile users and privacy protection technology to improve the quality of recommendation. However, they ignore the information about users' sentiment tendency and personality traits [14], hence reducing recommendation accuracy. The traditional researches show that personality traits of individuals have an important influence on how users shop online, their psychological preferences, and privacy awareness [15]. Therefore, how to integrate information, such as sentiment tendency, privacy concerns, and personality traits, into MPS is a research puzzle in this field.

To sum up, this paper analyzes influence factors of privacy concerns and online reviews and puts forward mobile personalized service recommender model based on sentiment analysis and privacy concerns. Firstly, it studies the sentiment analysis method at the level of opinion target and quantifies the sentiment tendency by opinion mining technology. Secondly, it integrates the quantitative influence of sentiment in recommend mechanism. Thirdly, it proposes a measurement method of personality traits integrated with privacy preference, mining personality traits, privacy preferences, and user preferences to obtain user groups with similar interests. At last, a novel hybrid collaborative filtering recommend method combining sentiment tendency and user's personality traits is put forward to achieve MPS.

This paper is organized as follows. After the introduction part, the paper discusses related work in the second part. In the third part, it proposes a mobile personalized service recommender model based on sentiment analysis and privacy concerns. In the fourth part, it evaluates the performance of the proposed model, and finally in the fifth part, the paper concludes with future work.

2. Related Work

2.1. Sentiment Analysis Based on Opinion Mining. Sentiment analysis of reviews on mobile commerce platform includes extraction, classification, retrieval, and induction of sentiment information [10]. There have machine-learning and knowledge-based approaches. The latter uses sentiment dictionary and syntactical rules for sentiment analysis. Hu and Liu [16] proposed a novel sentiment analysis method emphasizing reviews of product features by using traditional sentiment words library. Then, this method integrated context information with sentiment reviews to predict sentiment polarity of a product, increasing prediction accuracy. Wang et al. [17] analyzed the characteristics of ontology and Chinese online reviews and proposed a text mining model based on sentiment vocabulary ontology to build match of opinion target. This model attracts attention to mobile commerce enterprises. Similarly, Somprasertsri and Lalitrojwong [18] analyzed the sentiment tendency in the dependencies of commodity and opinion based on ontology model. This method integrated syntax with semantic information so as to quantify sentiment value, which had great business value in application. Ma et al. [19] constructed syntactic rules by adopting noun pruning and frequency filtering technology to extract review corpus. Then, they used syntactic rules to calculate the sentiment tendency of each sentence and boiled it down to four kinds of sentiment type.

Sentiment mining method based on machine learning, which is very different from the knowledge-based, requires more training time, and the model is too complex [20]. Therefore, Mi et al. improved the traditional sentiment mining method and built a text sentiment opinion mining model on the basis of binary language model and gray theory to achieve quality sentiment-oriented mining [21]. Additionally, considering the complexity of the machine-learning method, Somprasertsri and Lalitrojwong [22] extracted commodity features offline according to the maximum entropy and trained the model using corpus-tagged library. Finally, they extracted product features of online opinions by using traditional auxiliary commodity information. The model reduced learning time and achieved good prejudging results.

2.2. Mobile Personalized Services Recommendation Based on Sentiment Analysis. Some researchers have found that similar preferences of music can be matched by analyzing sentiment features. On one hand, Yang et al. [23] focused on sentiment characteristics of users who had common musical interests when calculating similarity of users. Similarly, Kuo et al. [24] found sentiment characteristics among different users by analyzing features of theme music in the movie and built a user interest of music model based on sentiment tendency to achieve accurate recommendations. On the other hand, Cai et al. [25] proposed cross-domain recommendation to predict users' sentiment tendency in music by analyzing opinions from Weibo in real time. Han et al. [26] presented a context-aware music recommendation system. They modeled and classified users' sentiment based on ontology language and solved the problem of recommendation data sparsity with context by using the nonnegative matrix factorization technique. Additionally, Mudambi and Schuff [27] integrated analysis of online reviews with users' behavior preference mining. They can get better recommendation results through analysis of users' sentiment tendency toward good opinions.

The research of sentiment mining in collaborative filtering recommendation also becomes a hotspot recently. Traditional collaborative filtering recommendation method relies on a matrix of "user-review" to calculate user similarity or item similarity. But, it covers limited information and often leads to user interest bias due to many factors such as context. She and Chen [28] introduced the sentiment analysis method based on topic model into collaborative filtering recommendation, which increased accuracy by using rich-text review information. Winoto and Tang [29] studied sentiment analysis in the film recommendation system. They found user groups of similar interests through sentiment tendency submitted actively by users and then proposed a sentiment-aware collaborative filtering method. Shi et al. [30] proposed a method based on decomposition matrix when calculating the films similarity on the basis of sentiment. They integrated sentiment analysis results in the process of collaborative filtering recommendation, collecting users' ratings data of IMDB in three stages, that is, before

a movie, during a movie, and after a movie. Then, this method utilized users' sentiment reviews published on Twitter for improving the algorithm of collaborative filtering recommendation and accurately forecasting the box office.

2.3. Mobile Personalized Services Recommendation Based on Privacy Concerns. The term "privacy concerns" is used to measure consumers' worriedness, perceptions, and controls of information privacy [31]. Culnan and Armstrong [32] claimed that mobile internet activities often involved personal privacy information, such as payment information, motion, and geographical location. Obtaining user's personal information is crucial to the survival and development of MPS providers [31], which can be used to accurately recommend users' information, and to stimulate their continuous acceptance of the service by satisfying their needs. Thus, their satisfaction and loyalty are improved [33, 34]. However, the problem of privacy leak is increasingly serious, and more and more researchers propose new ways to solve privacy issues [35]. For example, some people use data encryption and anonymous protection in mobile personalized services to generate recommended content for reducing privacy worriedness. Li and Sarkar [36] studied individual privacy policy mechanism based on cluster and analyzed problems of privacy concerns in the process of user-based collaborative filtering recommendation. They presented a novel method of encrypting and anonymizing for individual privacy information to protect user privacy. McSherry and Mironov [37] put differential privacy technology into collaborative filtering recommendation system. They carried out differential privacy process on item-to-item covariance matrix. Through experiments in learning and forecasting stage, they found that introducing differential privacy protection technology in recommender system with high accuracy was feasible. Duckham and Kulik [38] proposed a privacy protection model of location based on noise model for encrypting the MPS in mobile environment.

Some scholars study privacy concerns and strategies from the subjective perspectives of users. Wang and Duan [39] synthesized the influence from individual and system privacy factors of online users to quantify these factors. Then, they designed a privacy quantification model based on universal vectors from the subjective perspective of user perception. Similarly, Sutanto et al. [40] took advertising application of personalized mobile as the example and designed a technology solution called Personalized, Privacy-Safe Application, trying to solve the paradox between personalization and privacy. This method was verified by field experiments. Chellappa and Shivendu [41] designed user's privacy protection model based on economics. They calculated users' utility and facilitated users' use by weighing the costs of not having a MPS service and being supervised by a service provider. In addition, Korzaan and Boswell [42] studied the influence of the five personality traits, which include extraversion, agreeableness, conscientiousness, neuroticism, and openness on consumer privacy, and found that agreeableness has an effect on privacy concerns. According to Junglas and Spitzmuller [43], openness and agreeableness had a negative influence on privacy concerns, and person with different personalities have different psychological tendency of privacy concerns. Choi and Choi [44] also showed that privacy concerns were different among users with different characteristics and thus affected their behaviors. For example, people who are open are more willing to experience and accept MPS.

3. Mobile Personalized Service Recommender Model Based on Sentiment Analysis and Privacy Concern

Since most sentiment analysis algorithms excessively rely on the size of data at present, it is hard to meet user requirements for fast response of MPS. Therefore, this paper adopts the text analysis method based on sentiment dictionary. But the traditional methods have a poor effect on emotion analysis, which is not included in sentiment dictionary. So, our model proposes a sentiment analysis algorithm based on sentiment vocabulary ontology to accurately obtain user interests. Firstly, users with similar sentiment tendency or privacy preference may have the same interest preference [13, 15]. Secondly, the key step is to find the k users who have similar characteristics with the target users in collaborative filtering recommendation method [45]. This article uses "user-sentiment" vector matrix to find the k users who have similar sentiment tendency. Considering the differences of users' privacy preference [13], this paper introduces the concept of "personality traits integrating with privacy preference." It uses "user-personality traits" vector matrix to find the most k similar users who have similar privacy preference. Thirdly, as shown in Figure 1, our model provides recommendation based on sentiment analysis and privacy concern.

3.1. Sentiment Tendency Analysis Algorithm Based on Sentiment Vocabulary Ontology

3.1.1. Building the Sentiment Vocabulary Ontology Library. Users usually give "positive" or "negative" opinions to online service. Firstly, users express opinions with words like "service quality is very good, value for money" after electric business transactions. Secondly, users use emotional words containing emotional polarity to describe the sentiments, such as "this shopping experience is very happy, I like it." Thirdly, the opinion of "I am not very happy today" shows a "negative" emotion tendency. However, the opinion of "I feel very happy" shows "active" emotion tendency. The example above shows that negative words can make emotion polarity of sentiments reverse. Adversative phrase and excessive adverbs of degree will also change user's emotion tendency, such as "mobile phone is very cool, but flashy" and "mobile phone keys are too small." Therefore, this section constructs three vocabulary ontology library of comment, emotion, and inversion for different descriptions of online opinions. Finally, this paper sums up users' sentiment tendency as negative, positive, and neutral in MPS. The negative is subdivided into evil, anger, fear, and grief. The positive is subdivided into good and happy. The neutral is surprise.

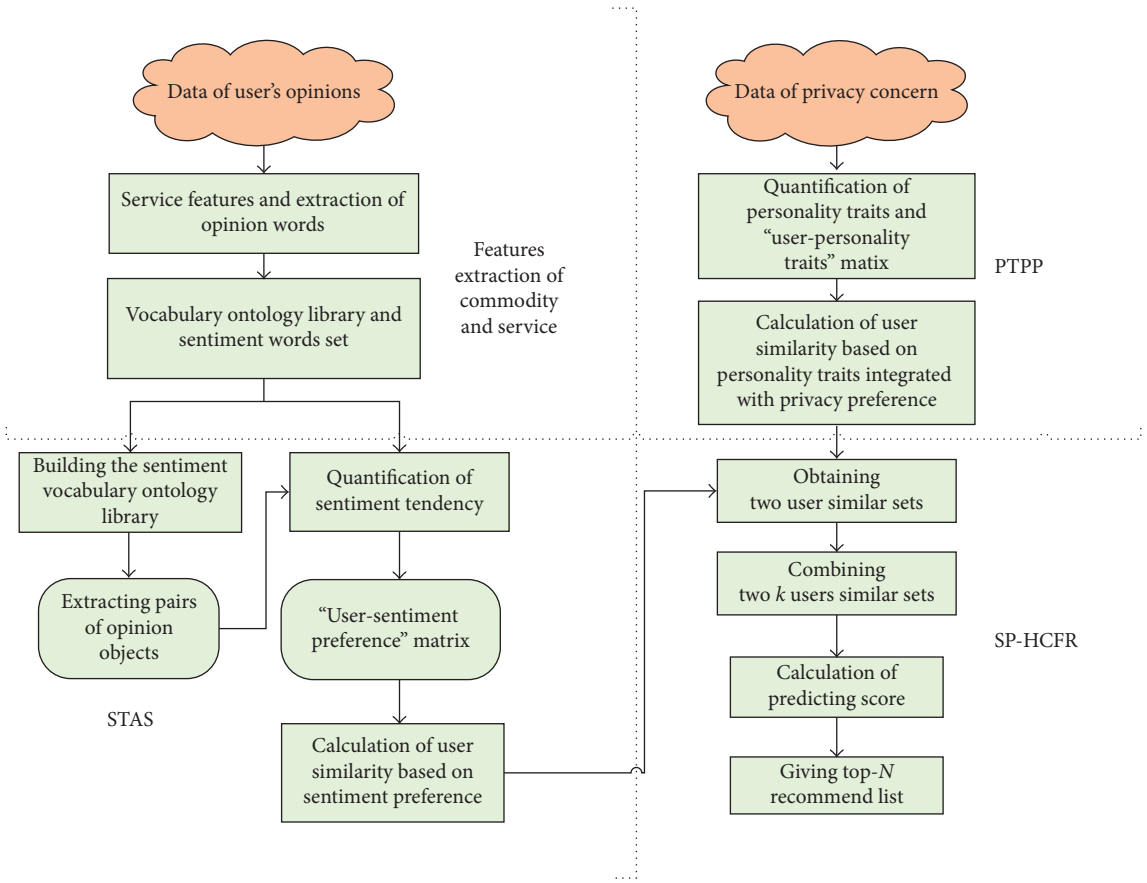


Figure 1: The model architecture.

(1) Building the Comment Vocabulary Ontology Library. Users use comments to directly express their preferences for goods and services. They reflect users' satisfied or unsatisfied attitude. Opinion mining methods calculate the sentiment type according to the comments. Considering the inconsistency and nonstandard use of the users' comments on the Internet, this paper puts forward comment vocabulary set based on ontology and uses triple ET = (B, R, E) to describe the sentiment transform. The B means the basic information of original words. The R means the synonymous relations. The E is the two-tuple of sentiment type and membership degree.

(2) Building the Emotion Vocabulary Ontology Library. Users use emotion words to directly express their emotion for goods or services. They reflect users' "like" or "dislike" attitude. Opinion mining methods calculate the sentiment type according to the emotion words. Considering the differences of emotion vocabulary classification, this paper studies eight kinds of emotion expression in social relationships [46] and twelve kinds of Chinese vocabulary classification [47]. On this basis, we select seven emotion types, which are happy, good, anger, sorrow, fear, evil, and surprise. Then, we use similarity algorithm to quantify the value of sentiment tendency.

(3) Building the Inversion Vocabulary Ontology Library. This paper complements the sentiment vocabulary ontology library of DUITR [48], which collects escape words to construct inversion vocabulary ontology library, including overdone degree adverbs, twist words, and negative words. But it does not consider general modifiers. Considering the complexity of the Chinese, this paper calculates the distance length in one statement between sentiment and inversion words (range [−4, +4]) to adjust the polarity of sentiment words.

3.1.2. Sentiment Tendency Analysis Algorithm. Sentiment tendency analysis algorithm based on sentiment vocabulary ontology (STAS) (Algorithm 1) divides opinion statements into pairs of words based on condition random fields (CRFs) [49] and domain ontology. Then, STAS extracts pairs of opinion targets based on CRFs model, including opinion target, opinion word, and phrase for sentiment tendency analysis, and improves the efficiency of the opinion mining.

Definition 1. Sentiment vocabulary library $S = \langle s_1, s_2, \ldots, s_m \rangle$; s_i is the sentiment word, $i = \{1, 2, \ldots, m\}$.

Definition 2. Opinion text set $D = \langle d_1, d_2, \ldots, d_n \rangle$; d_i is the opinion text, $i = \{1, 2, \ldots, n\}$. $d_i = \langle w_1, w_2, \ldots, w_k \rangle$; w_i is the feature item of d_i, $i = \{1, 2, \ldots, k\}$.

Definition 3. Sentiment text set $D^s = \langle d_1^s, d_2^s, \ldots, d_t^s \rangle$ is the text collection containing sentiment words; d_i^s is the sentiment text, $i = \{1, 2, \ldots, t\}$.

Input: Sentiment Opinions Object Set, Comment Vocabulary Ontology Library, Emotion Vocabulary Ontology Library, and Inversion Vocabulary Ontology Library.

Output: Triple<Opinions Service, Sentiment Type, Sentiment Tendency Value>.

(1) It extracts opinion objects and opinion words based on CRF model [50] and judges whether the opinion sentence has sentiment words. If there are no sentiment words, the STAS ends directly. Otherwise, it jumps to the next step.

(2) It matches each opinion text phrase through sentiment vocabulary ontology library and constructs the relationship between opinion objects and opinion phrases.

(3) It traverses opinion phrases to match sentiment words in opinion objects by using sentiment vocabulary ontology library. If sentiment word exists, STAS changes it to sentiment type, calculates the sentiment polarity according to inversion words, and is stored as Triple<Opinions Service, Sentiment Type, Sentiment Tendency Value>. If it does not exist, it outputs as Triple<Opinions Service, "neutral," 0>.

(4) STAS repeats step 1 to step 3, until it judges all the opinion objects.

The overall sentiment tendency of opinion objects is calculated by weight values of comment words, emotion words, and reverse words. STAS quantifies the sentiment type of opinion words based on artificial tagging and fuzzy set theory [50] and adopts the membership degree to predict its value. Besides, it calculates the similarity between s and s' based on Levenstein edit distance and predicts the sentiment tendency value of sentiment words. Thirdly, it calculates the sentiment tendency value of reverse words by the Triple<opinions object, reverse words, sentiment polarity> and PMI-IR algorithm [51]:

$$\mathrm{sim}(s, s') = \max(0, (\min(|s_i|, |s_i'| - \mathrm{ed}(s_i, s_i'))/\min(|s_i|, |s_i'|)),$$

where $\mathrm{ed}(s_i, s_i')$ is the Levenstein edit distance between s_i and s_i'.

$$\mathrm{PMI}(\mathrm{word}) = \sum_{p\,\mathrm{word} \in P_{\mathrm{set}}} \mathrm{PMI}(\mathrm{word}, p\,\mathrm{word}) - \sum_{n\,\mathrm{word} \in N_{\mathrm{set}}} \mathrm{PMI}(\mathrm{word}, n\,\mathrm{word}),$$

where *word* is the target word whose sentiment type is unknown, and P_{set} and N_{set} are the positive sentiment vocabulary set and negative sentiment vocabulary set, respectively, in basic sentiment words.

Algorithm 1: Sentiment tendency analysis based on sentiment vocabulary ontology.

The recognition of opinion target can be viewed as a sequence labeling problem. It labels the opinion corpora in sequence based on CRF model. It inputs a word string $w_1, w_2, \ldots, w_k$ and defines $D = D_1, D_2, \ldots, D_n$ as observed sequence, outputs a labeling sequence with the highest probability D^s, and then defines $D^s = D_1^s, D_2^s, \ldots, D_n^s$ as the observation sequence. The chain conditional probability distribution of D^s is shown in the following formula:

$$P(D^s|D) = \frac{1}{Z(D)} \exp \sum_i \sum_k \lambda_k f_k(d_{i-1}^s, d_i^s, d, i) + \sum_i \sum_k \mu_k g_k(d_i^s, d)), \quad (1)$$

$$Z(D) = \sum_d \exp\Big(\sum_i \sum_k \lambda_k f_k(d_{i-1}^s, d_i^s, d, i) \Big), \quad (2)$$

where f_k is the transfer characteristic function from location i to $i-1$, g_k is the state characteristic function on location i, λ_k and μ_k are weight values in the process of training, and Z_X is a normalizing factor relaying on D. CRF model uses Viterbi method to find a tag sequence named D^{s*} for getting the maximum $P(D^s|D)$, when the training uses the iterative algorithm based on maximum likelihood.

To the end, we calculate the sentiment tendency to judge users' sentiment preference of commodities or services by using sentiment vocabulary knowledge ontology library. The algorithm of STAS is shown as follows.

3.2. Measurement Method of Personality Traits Integrated with Privacy Preference

3.2.1. Measurement of User's Privacy Preference. Social network websites generally use privacy settings like "how to find me" (email or phone), "who are allowed to comment on me" (all, only fans, and only the persons who I care about), "recommend friends in your phone's address book," "who are allowed to give direct messages to me" (all, only fans, only the persons who I care about), "who are allowed to @ me" (only the persons who I care about, all), "binding with other accounts," "who are allowed to get my location" (all, only fans, only the persons who I care about), and so on. Henson et al. [52] found that users' active and personality traits have close relevance with their privacy behaviors. Mobile users' active in social network can be described with numbers of blogs delivered, photos uploaded, and opinions given. It also has significant correlation with "who are allowed to comment on me" (AC), "who are allowed to give direct messages to me" (AM), and "who are allowed to get my location" (AG). Thus, this paper chooses the above three items as a measurement index of privacy preference. Taking Weibo as an example, this paper uses multiple linear regression model and (3) to calculate the intensity of privacy preference. The three evaluation indexes are shown as follows, all setting half a year as the time period: (1) "who are allowed to comment on me" (AC_u), (2) "who are allowed to give direct messages to me" (AM_u), and (3) "who are allowed to get my location" (AG_u). The privacy preference vectors of users are named "P_u":

$$P_u = (\mathrm{AM}_u, \mathrm{AC}_u, \mathrm{AG}_u), \quad (3)$$

Input: Mobile user u, recommend service set *Service(R)*, and score matrix of "user-sentiment."
Output: $\mathrm{sim}(u,v)_{\text{sentiment-pearson}}$.

(1) To calculate the average user preference for a certain sentiment type:
$r_{u,s'} = \left(1/|S_{u,s'}|\right)\sum_{s\in S_{u,s'}} r_{u,s,s'}$,
$S_{u,s'} = \left\{s | s \in S, r_{u,s} \neq \text{null}, ss' = 1\right\}$, where $|S_{u,s'}|$ is the number of items in $S_{u,s'}$ and $r_{u,s,s'}$ is the user preference for service s with sentiment s', and to construct a two-dimensional "user-sentiment" preference matrix.

(2) It proposes an improved user's similarity calculation method based on preference matrix of "user-sentiment":
$$\mathrm{sim}(u_i,u_j)_{\text{sentiment-pearson}} = \left(\sum_{s'\in S'}\left(r_{u_i,s'} - \overline{r_{u_i}^{s'}}\right)\left(r_{u_j,s'} - \overline{r_{u_j}^{s'}}\right)\right) \Big/ \left(\sqrt{\sum_{s'\in S'}\left(r_{u_i,s'} - \overline{r}_{u_i}\right)^2 \sum_{s'\in S'}\left(r_{u_j,s'} - \overline{r}_{u_j}\right)^2}\right),$$
where $\overline{r_{u_i}^{s'}}$ is the average sentiment preference of u_i in all service-relative sentiment. It can select the k users similar set of u_i based on $\mathrm{sim}(u_i,u_j)_{\text{sentiment-pearson}}$.

Algorithm 2: User similarity calculation based on sentiment analysis.

P_u can be quantified by the following formula to show the value of privacy preference intensity:

$$P_u = \beta_1(\mathrm{AM}_u) + \beta_2(\mathrm{AC}_u) + \beta_3(\mathrm{AG}_u). \tag{4}$$

The probability value of AM_u, AC_u, and AG_u can be 0 or 1, respectively, meaning "allowed" or "not allowed." The measurement of P_u means specific preference weighted value in privacy settings of mobile users. P_u with lower value belongs to the group of low awareness in privacy concern, whereas the reverse is the group of high awareness.

3.2.2. Measurement of User's Personality Traits. Traditional user's personality traits are generally quantified with discrete values but yield no ideal effect. This paper adopts the "big-five personality" questionnaire and users' self-rated scores to get continuous personality traits data. Selfhout et al. [53] found that not all five dimensions of "big-five personality traits" could affect users' behaviors and social relations in network. Therefore, we select openness, extraversion, and agreeableness as the research objects and quantify the personality traits based on multivariate linear regression model. This model selects 15 independent variables, which are friends, visitors, blogs, photos, photo albums, direct messages, opinions, and the forwarded number of opinions, photos, hot topics, videos, music, and so on. This model extracts 400 users' behavior character data as a sample set from the questionnaire survey, Web crawler-like tools, and characterizes them in social network behavior data:

$$S_u = (O_u, C_u, E_u, A_u, N_u). \tag{5}$$

Among them, the vector S_u is defined as user's personality traits, O_u is the score of openness calculated by the number of friends FN_u, forwarded hot topic TC_u, comments CN_u, and uploaded photos PN_u. C_u is the score of responsible. E_u is the score of extrovert calculated by the number of friends FN_u, blogs delivered DB_u, and comments CN_u. A_u is the score of agreeableness calculated by the number of friends FN_u, forwarded hot topic TC_u, comments CN_u, and uploaded photos PN_u. N_u is the score of neurological. This paper calculates the value of personality traits preference by quantifying the S_u. Therefore, the calculation method of user's personality traits integrated into privacy preference (PTPP) is shown as follows:

$$\begin{aligned} O_u &= \beta_1(\mathrm{FN}_u) + \beta_2(\mathrm{TC}_u) + \beta_3(\mathrm{CN}_u) + \beta_4(\mathrm{PN}_u), \\ E_u &= \beta_1(\mathrm{FN}_u) + \beta_2(\mathrm{DB}_u) + \beta_3(\mathrm{CN}_u), \\ A_u &= \beta_1(\mathrm{FN}_u) + \beta_2(\mathrm{TC}_u) + \beta_3(\mathrm{CN}_u) + \beta_4(\mathrm{PN}_u), \\ P_u &= \beta_1(\mathrm{AM}_u) + \beta_2(\mathrm{AC}_u) + \beta_3(\mathrm{AG}_u), \end{aligned} \tag{6}$$

where $S_{u,p} = (O_u, E_u, A_u, P_u)$, and O_u, E_u, A_u, and P_u are, respectively, calculated by multivariate linear regression model.

3.3. Hybrid Collaborative Filtering Recommendation Method Based on Sentiment Analysis and Privacy Concerns

3.3.1. User Similarity Calculation Based on Sentiment Analysis. We change the score matrix of "user-service" to preference matrix of "user-sentiment" to calculate the similarity of users. The construct of preference matrix of "user-sentiment" defined as $R_{u,s}$ relies on the score matrix of "user-service" and the correlation matrix of "service-sentiment." Each low in $R_{u,s}$ is the sentiment preference vector: $R_{u,s} = \left\{r_{u,s} | u \in U, s \in S, r_{u,s} \in [0,100]\right\}$. $r_{u,s}$ is the value of user preference u for specific sentiment s.

3.3.2. User Similarity Calculation Based on Personality Traits Integrated with Privacy Preference. This paper uses matrix of "user-personality traits" $R_{u,p}$ to construct the k users similar set of target user u. The item score in $R_{u,p}$ is calculated by measurement method of PTPP. Each item score in $R_{u,p}$ is the weighted value computed by the score of "openness," "extraversion," "agreeableness," and "privacy preference." Each row of $R_{u,p}$ is the score vector of personality traits integrated with privacy preference, and the paper comprehensively quantifies $r_{u,p}$ in comprehensive quantification of personality traits in four dimensions so as to calculate the similarity among users.

3.3.3. Collaborative Filtering Recommend Method Combining User's Sentiment Tendency with Personality Traits. The core of SP-HCFR is the calculation of hybrid users' similarity that is weighted according to the $\mathrm{sim}(u,v)_{\text{sentiment-pearson}}$

Input: Mobile user u, privacy preference vector P_u, personality trait vector S_u, and score matrix of "user-personality traits" $R_{u,p}$: $R_{u,p} = \{r_{u,p} | p \in P, u \in U, r_{u,p} \in [0,1]\}$

Output: $\text{sim}(u,v)_{\text{privacy-preference}}$.

(1) It obtains users' basic data of personality traits through the questionnaire of "personality assessment of mobile user" S_u: $S_u = (O_u, C_u, E_u, A_u, N_u)$

(2) It calculates the comprehensive value of users' personality traits under privacy concern through data of mobile network behavior $S_u = (O_u, E_u, A_u, P_u)$:
$O_u = \beta_1(\text{FN}_u) + \beta_2(\text{TC}_u) + \beta_3(\text{CN}_u) + \beta_4(\text{PN}_u)$,
$E_u = \beta_1(\text{FN}_u) + \beta_2(\text{DB}_u) + \beta_3(\text{CN}_u)$,
$A_u = \beta_1(\text{FN}_u) + \beta_2(\text{TC}_u) + \beta_3(\text{CN}_u) + \beta_4(\text{PN}_u)$,
$P_u = \beta_1(\text{AM}_u) + \beta_2(\text{AC}_u) + \beta_3(\text{AG}_u)$.

(3) Then, it uses the following equation to compute user similarity based on $R_{u,p}$. Among them, user $\vec{u}$ and $\vec{v}$ are the personality trait vectors integrating with privacy concern:

$$\text{sim}(u,v)_{\text{privacy-preference}} = (\vec{u} \cdot \vec{v}) / (\|\vec{u}\| \|\vec{v}\|).$$

Algorithm 3: User similarity calculation based on personality traits integrating with privacy preference.

Input: Mobile user u, recommend service set *Service(R)*, and score matrix of "user-sentiment" and "user-personality traits."
Output: Top-N recommend services and its score.

(1) The calculation of user similarity is based on sentiment analysis.

(2) The calculation of user similarity is based on personality traits integrated with privacy preference.

(3) It searches the k users similar set of target use by using the composite user similarity, which is calculated by the following equation:

$$\text{sim}(u,v) = \alpha \times \text{sim}(u,v)_{\text{privacy-preference}} + (1-\alpha) \times \text{sim}(u,v)_{\text{sentiment-pearson}}$$

It uses fifty percent of cross-validation method to determine the parameters $\alpha \in [0,1]$. When $\alpha = 0$, $\text{sim}(u,v) = \text{sim}(u,v)_{\text{privacy-preference}}$, and when $\alpha = 1$, $\text{sim}(u,v) = \text{sim}(u,v)_{\text{sentiment-pearson}}$.

(4) It predicts the users' preference and sorts in Top-N to give recommendation:

$$\begin{aligned} P'_{u,i} &= \overline{P_u} + \alpha \times \left(\left(\sum_{v \in V} \text{sim}(u,v)_{\text{privacy-preference}} \times \left(P_{u,i} - \overline{P_u} \right) \right) / \left(\sum_{v \in V} \text{sim}(u,v) \right) \right) \\ &\quad + (1-\alpha) \times \left(\left(\sum_{t \in T} \text{sim}(u,t)_{\text{sentiment-pearson}} \times \left(P_{t,i} - \overline{P_u} \right) \right) / \left(\sum_{t \in T} \text{sim}(u,t) \right) \right), \quad V \neq \varnothing \,\&\, T \neq \varnothing \\ &= \overline{P_u} + \left(\left(\sum_{v \in V} \text{sim}(u,v)_{\text{privacy-preference}} \times \left(P_{v,i} - \overline{P_u} \right) \right) / \left(\sum_{v \in V} \text{sim}(u,v) \right) \right), \quad V \neq \varnothing \,\&\, T = \varnothing \\ &= \overline{P_u} + \left(\left(\sum_{t \in T} \text{sim}(u,t)_{\text{sentiment-pearson}} \times \left(P_{t,i} - \overline{P_u} \right) \right) / \left(\sum_{t \in T} p\text{sim}(u,t) \right) \right), \quad V = \varnothing \,\&\, T \neq \varnothing. \end{aligned}$$

Algorithm 4: Collaborative filtering recommend method combining sentiment tendency with user's personality traits.

calculated by Algorithm 2 and $\text{sim}(u,v)_{\text{privacy-preference}}$ calculated by Algorithm 3.

4. Performance Evaluation

4.1. Experimental Data and Evaluation Standards

4.1.1. Data Sets of Practical Application. Firstly, we tag the text opinion corpus and select group of opinion match, including opinion words and opinion targets. Experiments fetch 800 opinion corpus from Weibo and sorts out 500 users, 305 opinion targets, 541 opinion words, and two-tuple of sentiment opinions unit 416. This paper uses <opinion target, opinion words> to construct these sentiment opinion unit tagged by experts. Secondly, we use the sentiment opinion unit as a test data set and classify each unit into different sentiment types with artificial methods. Then, STAS uses these artificial results as contrast standards. On this basis, we obtain real recommended data sets with 500 users, 7 sentiment types, 800 score records, and 305 items.

Table 1: The data sets of recommend.

Data sets	Score records	User	Item	Sentiment type
Data sets of practical application	800	500	305	7
Training data sets	4,544,409	105,137	25,058	16
Testing data sets	19,506	160	3,396	16

4.1.2. Standard Data Sets of Recommend. It selects Moviepilot-mp.mood, which contains the information of users, movies, sentiment types, locations, opinion time, and so on. It divides the Moviepilot-mp.mood into training data set and testing data set. The training data set includes 4,544,409 score records, which is scored by 105,137 users in 25,058 movies under 16 sentiment types. The testing data set includes 19,506 score records, which is scored by 160 users in 3,396 movies under 16 sentiment types. It adopts five marks as a step size when users give scores in the range of 0–100. The data sets of recommend are listed in Table 1.

Table 2: The experimental result of STAS.

Sentiment type	Precision	Recall	$F1$
Sorrow	0.62	0.65	0.63
Happy	0.64	0.66	0.65
Anger	0.66	0.67	0.66
Fear	0.68	0.71	0.69
Surprise	0.69	0.73	0.71
Good	0.74	0.77	0.75
Evil	0.75	0.79	0.77

Table 3: Influence of different conditions of STAS.

Different condition combination	The results of sentiment analysis		
	Recall	Precision	$F1$
Comment words, emotion words, and inversion words	0.66	0.64	0.65
Comment words and inversion words	0.61	0.56	0.58
Emotion words and inversion words	0.53	0.49	0.51
Emotion words	0.34	0.28	0.31

4.1.3. Evaluation of standard of experiment. It uses precision rate p, recall rate r, and F-measure F_β for the effect analysis of STAS:

$$\begin{aligned}\text{Precision} &= \frac{\text{the number of opinion units really belongs to correct sentiment type}}{\text{the sum of opinion unit numbers belongs to correct and incorrect sentiment type}},\\ \text{Recall} &= \frac{\text{the number of opinion units really belongs to correct sentiment type}}{\text{the sum of opinion unit number judged to correct and incorrect sentiment type}},\\ F_\beta &= \frac{\left(\beta^2+1\right)\times p\times r}{\beta^2\times p\times r}.\end{aligned} \tag{7}$$

It uses mean average precision (MAP) for the effective analysis of SP-HCFR (Algorithm 4). The MAP measures the sort accuracy of Top-N. The higher the MAP, the higher the accuracy of the recommendation.

4.2. Experimental Results of STAS. In order to verify the superiority, STAS matches 416 two-tuple<sentiment opinion unit, sentiment type> with ontology of sentiment vocabulary library, including the ontology library of comment words, emotion words, and reverse words. The paper uses STAS to judge the sentiment type of each opinion unit in testing data set. The results of STAS experiment are compared with artificial tag standard, which are shown in Table 2.

It can be seen that STAS has a high rate of precision, recall, and $F1$ in seven sentiment types for the classification of sentiment. It also suggests that STAS can obtain accurate user preferences. At the same time, it divides text corpus into several sentiment opinion units, which are mined as 2-tuple<sentiment opinion unit, sentiment type> for more accurate prediction of sentiment analysis. The experimental results also show that three kinds of sentiment vocabulary ontology library have more accurate sentiment analysis than comments vocabulary. STAS can make up these problems, such as diversity of sentiment, deviation of sentiment similarity, and incorrect sentiment judgement.

To verify the influence of sentiment analysis based on comment words, emotion words, and inversion words, this paper does four experiments, respectively, by selecting comment words, emotion words, and inversion words, by selecting comment words and inversion words, by selecting emotion words and inversion words, and by selecting emotion words. It compares overall sentiment analysis through the indexes of precision rate, recall rate, and $F1$. The experimental results are shown in Table 3.

It can be seen that four kinds of words have a significant influence on sentiment analysis and improve the calculation accuracy of user preferences in MPS. Therefore, the STAS can semantically express the relationship among complex opinion targets by the ontology. It also improves the effectiveness of traditional opinion mining by combining with multisource information.

4.3. Experimental Results of PTPP. This paper adopts the stepwise multivariable linear regression model (SMLRM) to measure PTPP. It selects variables in SMLRM whose F-probability is less than 0.05. PTPP keeps variable with the highest significant level of coefficient, eliminates the non significant variables, and obtains the final significant regression equation of coefficient through several times of selection and elimination.

4.3.1. To Quantify Openness Dimension of Personality Traits. This paper finds the results in Table 4 through SMLRM that the dependent variable (openness) has linear regression relationship with the independent variable, such as the number of comments CN_u (regression coefficient is −0.007), friends FN_u (regression coefficient is 0.126), uploaded photos PN_u (regression coefficient is 0.595), and forwarded hot topics TC_u (regression coefficient is 0.088). At the same time, the openness has a positive linear relationship with the number of friends, uploaded photos, and forwarded hot

Table 4: Regression model in openness dimension.

Variable	Regression coefficient b	Standard coefficient r	T test	Significant degree p
FN_u	0.126	0.807	11.105	0.000
TC_u	0.088	0.122	2.491	0.015
CN_u	−0.007	−0.145	−2.458	0.014
PN_u	0.595	0.919	2.163	0.031

Table 5: Regression model in extraversion dimension.

Variable	Regression coefficient b	Standard coefficient r	T test	Significant degree p
FN_u	0.105	0.655	11.942	0.000
DB_u	0.126	0.167	3.748	0.000
PN_u	0.877	0.171	3.399	0.001

Table 6: Regression model in agreeableness dimension.

Variable	Regression coefficient b	Standard coefficient r	T test	Significant degree p
FN_u	0.146	0.778	10.572	0.000
TC_u	0.088	0.099	2.004	0.046
CN_u	−0.009	−0.171	−2.878	0.004
PN_u	1.162	0.194	3.472	0.001

Table 7: Regression model in privacy preference dimension.

Variable	Regression coefficient b	Standard coefficient r	T test	Significant degree p
AM_u	−0.821	−0.654	−11.941	0.000
AC_u	−0.139	−0.166	−3.747	0.031
AG_u	−0.137	−0.170	−3.399	0.022

topics. The number of uploaded photos exerts most influence to openness. The composite correlation coefficient $R = 0.832$, and the decision correlation coefficient $R^2 = 0.767$. It means that regression model in openness dimension has positive correlation with these factors and fits testing data set well.

Regression equation:

$$O_u = 0.126(FN_u) + 0.088(TC_u) - 0.007(CN_u) + 0.595(PN_u). \quad (8)$$

4.3.2. To Quantify Extraversion Dimension of Personality Traits. It finds the results in Table 5 through SMLRM that the dependent variable (extraversion) has linear regression relationship with the independent variable, such as the number of friends FN_u (regression coefficient is 0.105), uploaded photos PN_u (regression coefficient is 0.877), and blogs delivered DB_u (regression coefficient is 0.126). At the same time, the extraversion has a positive linear relationship with the number of friends, uploaded photos, and blogs delivered. The composite correlation coefficient $R = 0.780$, and the decision correlation coefficient $R^2 = 0.821$. It means that regression model in extraversion dimension has positive correlation with these factors and has a good fitting degree in testing data set.

Regression equation:

$$E_u = 0.105(FN_u) + 0.126(DB_u) + 0.877(CN_u). \quad (9)$$

4.3.3. To Quantify Agreeableness Dimension of Personality Traits. It finds the results in Table 6 through SMLRM that the dependent variable (agreeableness) has linear regression relationship with the independent variable, such as the number of friends FN_u (regression coefficient is 0.146), forwarded hot topics TC_u (regression coefficient is 0.088), comments CN_u (regression coefficient is −0.009), and uploaded photos PN_u (regression coefficient is 1.162). At the same time, the agreeableness has a positive linear relationship with the number of friends, forwarded hot topics, comments, and uploaded photos. Finally, the composite correlation coefficient $R = 0.886$, and the decision correlation coefficient $R^2 = 0.771$. It means that regression model in agreeableness dimension has positive correlation with these factors and fits testing data set well.

Regression equation:

$$A_u = 0.146(FN_u) + 0.088(SC_u) - 0.009(CN_u) + 1.162(PN_u). \quad (10)$$

4.3.4. To Quantify Privacy Preference Dimension of Personality Traits. The person who gives low scores in this dimension prefers to ignore privacy and security and shares the privacy to others. He or she does not care about tagging his or her place and social network information but begins to care about the privacy setting. Instead, he or she prefers to protect his or her privacy, rejects new service, and tends to protect his or her own autonomy. This paper finds the results in Table 7 through SMLRM that the dependent variable (privacy preference) has a linear regression relationship with the independent variables, such as "who are allowed to comment on me" AC_u (regression coefficient is −0.821), "who are allowed to give direct messages to me" AM_u (regression coefficient is −0.139), and "who are allowed to get my location" AG_u (regression coefficient is −0.137). The regression coefficients of these three factors are less than 0, which means privacy preference is of negative correlation with the three factors, and the "who are allowed to give direct messages to me" has the most powerful impact on users' privacy preference. Finally, the composite correlation coefficient $R = 0.850$, and the decision correlation coefficient $R^2 = 0.765$. It means that regression model in privacy preference dimension has positive correlation with these factors and fits testing data set well.

Regression equation:

$$P_u = -0.821(AM_u) - 0.139(AC_u) - 0.137(AG_u). \quad (11)$$

From the experimental results, it shows that personality traits with privacy preference, openness, extraversion, and agreeableness can better reflect and quantify online behaviors

Table 8: Comparison of SP-HCFR under the influence of different α ($P@R$).

SP-HCFR	$P@5$ ($k=10, 20, 30, 50$)				$P@10$ ($k=10, 20, 30, 50$)			
	10	20	30	50	10	20	30	50
0.0	0.49	0.54	0.56	0.59	0.45	0.50	0.54	0.55
0.2	0.51	0.57	0.58	0.60	0.50	0.53	0.55	0.57
0.4	0.53	0.58	0.59	0.60	0.52	0.55	0.56	0.58
0.6 (cutoff points)	0.55	0.59	0.60	**0.61**	0.53	0.56	0.57	**0.59**
0.8	0.54	0.57	0.59	0.60	0.51	0.55	0.56	0.57
1.0	0.50	0.54	0.56	0.59	0.49	0.53	0.54	0.56

Table 9: Comparison of SP-HCFR under the influence of different α (MAP and DOA).

SP-HCFR	MAP ($k=10, 20, 30, 50$)				DOA ($n=80\%$–$20\%, 70\%$–$30\%, 60\%$–$40\%, 50\%$–50%)			
	10	20	30	50	80%–20%	70%–30%	60%–40%	50%–50%
0.0	0.53	0.57	0.60	0.62	0.79	0.82	0.84	0.85
0.2	0.56	0.59	0.62	0.63	0.81	0.84	0.86	0.86
0.4	0.57	0.60	0.63	0.64	0.82	0.85	0.87	0.87
0.6 (cutoff points)	0.58	0.61	0.64	**0.65**	0.83	0.85	0.87	**0.88**
0.8	0.56	0.60	0.62	0.64	0.82	0.84	0.86	0.87
1.0	0.54	0.58	0.61	0.63	0.80	0.83	0.85	0.87

of mobile users. PTPP obtains more accurate and objective scores of personality traits, improving the accuracy of calculation of user similarity in follow-up collaborative filtering recommendation.

4.4. Experimental Results of SP-HCFR

4.4.1. Comparison of Hybrid Collaborative Filtering Methods Based on User Similarity under the Influence of Different α. Different values of α, which is set to $\alpha = 0, 0.2, 0.4, 0.6, 0.8, 1.0$, mean different weight influence of sentiment characteristic and privacy concern in collaborative recommend. SP-HCFR combines the method of user collaborative filtering algorithm based on personality traits (PP-UCF, $\alpha = 0.0$) with the method of user collaborative filtering algorithm based on sentiment analysis (SA-UCF, $\alpha = 1.0$). This paper has some comparison experiments to the influence in the weighted coefficient α. The results are shown in the evaluating indexes of MAP, DOA, $P@10$, and $P@5$ and are listed in Tables 8 and 9. The α and k are set to $\alpha = 0.2, 0.4, 0.6, 0.8$ and $k = 10, 20, 30, 50$. Firstly, several experiment results show that SP-HCFR has a higher accuracy than others, and it reaches the highest accuracy when $\alpha = 0.6$. Secondly, the value of α is of nonlinear relationship with the sorting accuracy. Thirdly, the results show that SP-HCFR can obtain more accuracy in k similar users by using the combination of user similarity calculation and also improves the accuracy of recommendation.

4.4.2. Performance Comparison of Different Collective Filtering Algorithms. In order to analyze the influence of privacy preference, personality traits, and sentiment characteristics on MPS, this paper does some comparisons among different algorithms. The results are shown in Figure 2. Firstly, SP-HCFR is ahead of traditional collaborative

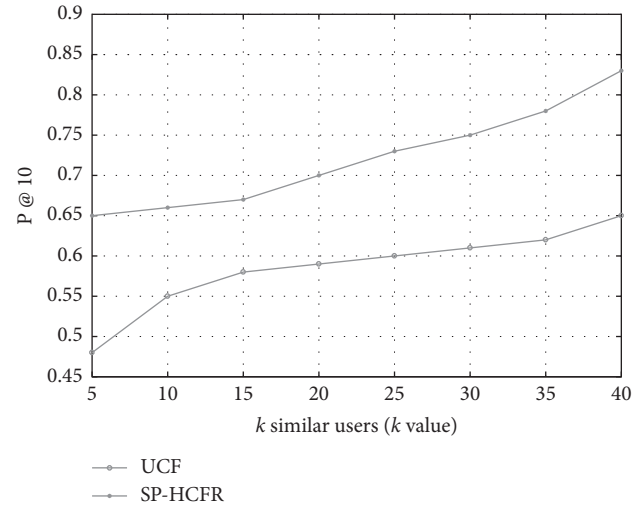


Figure 2: Performance comparison between SP-HCFR and UCF.

filtering algorithm based on user (UCF) in the indexes of $P@10$, MAP while the importance weight is $\alpha = 0.6$. Secondly, although actual score data are sparse, SP-HCFR solves this problem by using the information of privacy preference and sentiment characteristics for calculation of user similarity. Thirdly, the results show that it has great significance for introducing privacy concern, personality traits, and sentimental characteristics in recommendation.

This paper does the performance comparison between collaborative filtering recommend method based on personality traits combined with privacy concerns (PP-UCF) and collaborative filtering recommend method based on sentiment analysis (SA-UCF) and UCF, while $\alpha = 0.6$. The results are shown in Figure 3. Firstly, SA-UCF has not considered the influence of privacy preference on users'

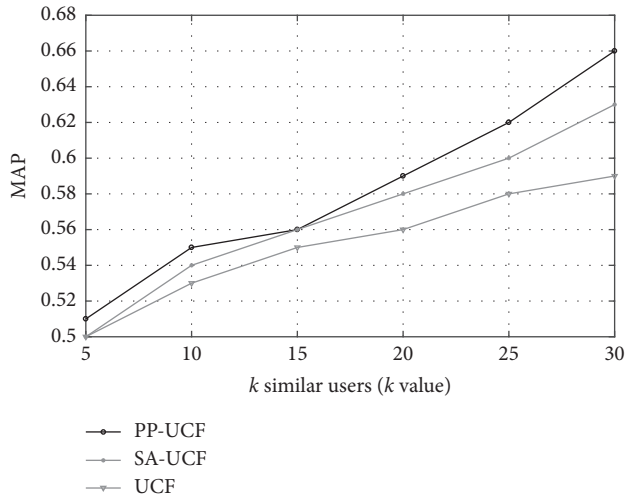


Figure 3: Performance comparison among PP-UCF, SA-UCF, and UCF.

interest, which cannot significantly improve the accuracy of recommendation. Secondly, PP-UCF has a greater influence on the quality of recommendation, which means privacy concern is more important than sentiment characteristics to obtain precision user's interest. Thirdly, PP-UCF and SA-UCF both have better performances than UCF. It also shows that using the information of privacy preference and sentiment characteristics for calculation of user similarity can solve the problem of data sparseness and cold start.

5. Conclusion and Future Work

With the wide use of MPS in mobile commerce, the problems in protecting user's privacy and obtaining user's interest with complicated sentiment are outstanding. A novel personalized recommendation technology should be proposed to address the privacy concern and sentiment analysis. Therefore, this paper proposes a novel recommendation model based on subjective privacy preference and objective recommend technology. The main contributions are as follows: (1) since it can better match user's preference through learning complex sentiment, this paper puts forward STAS to mine the sentiment preference, effectively solves problems of data sparse and cold start, and improves the accuracy of user interests. (2) User's interest is similar with persons who have common privacy concern and personality traits. Therefore, this paper puts forward PTPP to obtain the k similar users. (3) This paper takes full use of both advantages of the above two contributions and puts forward a novel hybrid collaborative filtering recommendation method based on sentiment analysis and privacy concern to protect user's privacy and giving MPS.

A follow-up study may utilize methods of data mining to obtain user's dynamic privacy interest. At the same time, from the perspective of the importance of privacy protection, we will do the research of punctual personalized recommend services by using the control degree of users' privacy disclosure, intensity of privacy concerns, and so on.

Conflicts of Interest

The authors declare that there are no conflicts of interests regarding the publication of this paper.

Acknowledgments

This research was supported by Philosophy and Social Science Planning Project of Zhejiang Province (no. 18NDJC278YB), Natural Science Foundation of Zhejiang Province (no. LY18G020012 and LQ17G020003); Major Project of Social Science Foundation of China (no. 16ZDA053), Major Project of Humanities and Social Sciences in University of Zhejiang Province (no. 2016GH024), and Scientific Research Project of Zhejiang Provincial Education Department of China (no. Y201636584).

References

[1] Z. Lin, K. Y. Goh, and C. S. Heng, "The demand effects of product recommendation networks: an empirical analysis of network diversity and stability," *MIS Quarterly*, vol. 41, no. 2, pp. 397–420, 2017.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, no. 1, pp. 109–132, 2013.

[3] D. Gavalas, K. Charalampos, M. Konstantinos, and P. Grammati, "Mobile recommender systems in tourism," *Journal of Network and Computer Applications*, vol. 39, no. 1, pp. 319–333, 2014.

[4] D. V. Sowmini, V. R. Kagita, A. K. Pujari et al., "Collaborative filtering by PSO-based MMMF," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 569–574, San Diego, CA, USA, 2014.

[5] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.

[6] A. Yenter and A. Verma, "Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis," in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 540–546, New York, NY, USA, October 2017.

[7] P. Kefalas and Y. Manolopoulos, "A time-aware spatio-textual recommender system," *Expert Systems with Applications*, vol. 78, pp. 396–406, 2017.

[8] Z. Li, X. Fang, X. Bai, and O. R. Liu Sheng, "Utility-based link recommendation for online social networks," *Management Science*, vol. 63, no. 6, pp. 1938–1952, 2017.

[9] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013.

[10] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[11] P. Hawking, A. Stein, J. Zeleznikow et al., "Emerging issues in location based tourism systems," in *Proceedings of the International Conference on Mobile Business, IEEE*, pp. 75–81, Sydney, NSW, Australia, July 2005.

[12] J. Y. Tsai, S. Egelman, L. Cranor, and A. Acquisti, "The effect of online privacy information on purchasing behavior: an experimental study," *Information Systems Research*, vol. 22, no. 2, pp. 254–268, 2011.

[13] F. Provost, D. Martens, and A. Murray, "Finding similar mobile consumers with a privacy-friendly geosocial design," *Information Systems Research*, vol. 26, no. 2, pp. 243–265, 2015.

[14] P. Kefalas, P. Symeonidis, and Y. Manolopoulos, "New perspectives for recommendations in location-based social networks: time, privacy and explainability," in *Proceedings of the International Conference on Management of Emergent Digital Ecosystems, ACM*, pp. 1–8, Luxembourg City, Luxembourg, October 2013.

[15] J. Schrammel, J. Koffel, and M. Tscheligi, "Personality traits usage patterns and information disclosure in online communities," in *Proceedings of the 23rd British HCI Group Annual Conference on People and Computes: Celebrating People and Technology, British Computer Society*, pp. 169–174, Swinton, UK, September 2009.

[16] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD 2004)*, pp. 168–177, Seattle, WA, USA, August 2004.

[17] H. W. Wang, P. Yin, J. Yao, and J. N. K. Liu, "Text feature selection for sentiment classification of Chinese online reviews," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 425–439, 2013.

[18] G. Somprasertsri and P. Lalitrojwong, "Mining feature-opinion in online customer reviews for opinion summarization," *Journal of Universal Computer Science*, vol. 16, no. 6, pp. 938–955, 2010.

[19] B. Ma, D. Zhang, Z. Yan, and T. Kim, "An LDA and synonym lexicon based approach to product feature from online consumer product reviews," *Journal of Electronic Commerce Research*, vol. 14, no. 4, pp. 304–314, 2013.

[20] J. Zhao, K. Liu, and G. Wang, "Adding redundant features for CRFs-based sentence sentiment classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 117–126, Honolulu, HI, USA, October 2008.

[21] C. Mi, X. Shan, Y. Qiang, Y. Stephanie, and Y. Chen, "A new method for evaluating tour online review based on grey 2-tuple linguistic," *Kybernetes*, vol. 43, no. 3-4, pp. 601–613, 2014.

[22] G. Somprasertsri and P. Lalitrojwong, "A maximum entropy model for product feature extraction in online customer reviews," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 786–791, Chengdu, China, September 2008.

[23] D. Yang, D. Zhang, Z. Yu et al., "A sentiment-enhanced personalized location recommendation system," in *Proceedings of the ACM Conference on Hypertext and Social Media, ACM*, pp. 119–128, Paris, France, May 2013.

[24] F. F. Kuo, M. F. Chiang, M. K. Shan et al., "Emotion-based music recommendation by association discovery from film music," in *Proceedings of the ACM International Conference on Multimedia*, pp. 7666–7674, Singapore, November 2005.

[25] R. Cai, C. Zhang, C. Wang, L. Zhang, and W. Y. Ma, "MusicSense: contextual music recommendation using emotional allocation modeling," in *Proceedings of the 15th International Conference on Multimedia (MULTIMEDIA'07)*, pp. 553–556, New York, NY, USA, 2007.

[26] B. J. Han, S. Rho, S. Jun, and E. Hwang, "Music emotion classification and context-based music recommendation," *Multimedia Tools and Applications*, vol. 47, no. 3, pp. 433–460, 2010.

[27] S. Mudambi and D. Schuff, "What makes a helpful online review? A study of customer reviews on Amazon.com," *MIS Quarterly*, vol. 34, no. 1, pp. 185–200, 2010.

[28] J. She and L. Chen, "TOMOHA: topic model-based HAshtag recommendation on twitter," in *Proceedings of the International Conference on World Wide Web Companion*, pp. 371-372, Seoul, Republic of Korea, April 2014.

[29] P. Winoto and T. Y. Tang, "The role of user mood in movie recommendations," *Expert Systems with Applications*, vol. 37, no. 8, pp. 6086–6092, 2010.

[30] Y. Shi, M. Larson, and A. Hanjalic, "Mining mood-specific movie similarity with matrix factorization for context-aware recommendation," in *Proceedings of the Workshop on Context-Aware Movie Recommendation, ACM*, pp. 34–40, Barcelona, Spain, September–October 2010.

[31] H. Cavusoglu, T. Q. Phan, H. Cavusoglu, and E. M. Airoldi, "Assessing the impact of granular privacy controls on content sharing and disclosure on Facebook," *Information Systems Research*, vol. 27, no. 4, pp. 848–879, 2016.

[32] M. J. Culnan and P. K. Armstrong, "Information privacy concerns, procedural fairness and impersonal trust: an empirical investigation," *Organization Science*, vol. 10, no. 1, pp. 104–115, 1999.

[33] H. Xu, T. Dinev, H. J. Smith, and P. Hart, "Examining the formation of individual's information privacy concerns: toward an integrative view," in *Proceedings of the 29th Annual International Conference on Information Systems*, pp. 14–17, Paris, France, December 2008.

[34] R. Zhang, J. Q. Chen, and C. J. Lee, "Mobile commerce and consumer privacy concerns," *Journal of Computer Information Systems*, vol. 53, no. 4, pp. 31–38, 2013.

[35] S. Menon and S. Sarkar, "Privacy and big data: scalable approaches to sanitize large transactional databases for sharing," *MIS Quarterly*, vol. 40, no. 4, pp. 963–982, 2016.

[36] X. B. Li and S. Sarkar, "Class restricted clustering and microperturbation for data privacy," *Management Science*, vol. 59, no. 4, pp. 796–812, 2013.

[37] F. McSherry and I. Mironov, "Differential private recommender system: building privacy into Netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 627–636, New York, NY, USA, 2009.

[38] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Lecture Notes in Computer Science*, pp. 152–170, Pervasive, Munich, Germany, 2005.

[39] B. Wang and Y. X. Duan, "Research on information privacy quantization method facing ubiquitous computing environment," *Computer Engineering and Applications*, vol. 47, no. 27, pp. 1–5, 2011.

[40] J. Sutanto, E. Palme, C. H. Tan, and C. W. Phang, "Addressing the personalization privacy paradox: an empirical assessment from a field experiment on smartphone user," *MIS Quarterly*, vol. 37, no. 4, pp. 1142–1164, 2013.

[41] R. K. Chellappa and S. Shivendu, "Mechanism design for "Free" but "No Free Disposal" services: The economics of personalization under privacy concerns," *Management Science*, vol. 56, no. 10, pp. 1766–1780, 2011.

[42] M. L. Korzaan and K. T. Boswell, "The influence of personality traits and information privacy concerns on behavioral intentions," *Journal of Computer Information Systems*, vol. 48, no. 4, pp. 15–24, 2016.

[43] I. Junglas and C. Spitzmuller, "Personality traits and privacy perceptions: an empirical study in the context of location-based services," in *Proceedings of the International Conference on Mobile Business*, pp. 36–47, Copenhagen, Denmark, June 2006.

[44] S. S. Choi and M. K. Choi, "Consumer's privacy concerns and willingness to provide personal information in location-based services," in *Proceedings of the International Conference on Advanced Communication Technology, IEEE*, pp. 2196–2199, Gangwon-do, Republic of Korea, February 2007.

[45] B. V. Roy and X. Yan, "Manipulation robustness of collaborative filtering," *Management Science*, vol. 56, no. 11, pp. 1911–1929, 2010.

[46] C. Q. Quan and F. J. Ren, "A blog emotion corpus for emotional expression analysis in Chinese," *Computer Speech and Language*, vol. 24, no. 4, pp. 726–749, 2010.

[47] Y. Zhang, Z. M. Li, F. J. Ren et al., "Semi-automatic emotion recognition from textual input based on the constructed emotion thesaurus," in *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, pp. 571–576, Wuhan, China, October 2005.

[48] L. Luo, "DUTIR at the BioCreative V.5.BeCalm tasks: A BLSTM-CRF approach for biomedical entity recognition in patents," in *Proceedings of the Biocreative V.5 Challenge Evaluation Workshop*, Barcelona, Spain, April 2017.

[49] J. D. Lafferty, A. Mccallum, and F. C. N. Pereira, "Condition random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning*, pp. 282–289, Sydney, NSW, Australia, 2002.

[50] W. Shi, H. W. Wang, and S. Y. He, "Study on construction of fuzzy emotion ontology based on Howne," *Journal of the China Society for Scientific and Technical Information*, vol. 31, no. 6, pp. 595–602, 2012.

[51] L. Yue, C. Malu, D. Umeshwar, and Z. ChengXiang, "Automatic construction of a context-aware sentiment lexicon: an optimization approach," in *Proceedings of the 20th International Conference on World Wide Web*, pp. 347–356, Hyderabad, India, March–April 2011.

[52] B. Henson, B. W. Reyns, and B. S. Fisher, "Security in the 21st century: examining the link between online social network activity, privacy, and interpersonal victimization," *Criminal Justice Review*, vol. 36, no. 3, pp. 253–268, 2011.

[53] M. Selfhout, W. Burk, S. Branje, J. Denissen, M. van Aken, and W. Meeus, "Emerging late adolescent friendship networks and big five personality traits: a social network approach," *Journal of Personality*, vol. 78, no. 2, pp. 509–538, 2010.

Hindawi
Mobile Information Systems
Volume 2018, Article ID 6920679, 9 pages
https://doi.org/10.1155/2018/6920679

Research Article

Automatic Task Classification via Support Vector Machine and Crowdsourcing

Hyungsik Shin [1] **and Jeongyeup Paek** [2]

[1]*School of Electronic and Electrical Engineering, Hongik University, Seoul, Republic of Korea*
[2]*School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea*

Correspondence should be addressed to Jeongyeup Paek; jpaek@cau.ac.kr

Received 5 December 2017; Revised 14 March 2018; Accepted 4 April 2018; Published 2 May 2018

Academic Editor: Dingqi Yang

Automatic task classification is a core part of personal assistant systems that are widely used in mobile devices such as smartphones and tablets. Even though many industry leaders are providing their own personal assistant services, their proprietary internals and implementations are not well known to the public. In this work, we show through real implementation and evaluation that automatic task classification can be implemented for mobile devices by using the support vector machine algorithm and crowdsourcing. To train our task classifier, we collected our training data set via crowdsourcing using the Amazon Mechanical Turk platform. Our classifier can classify a short English sentence into one of the thirty-two predefined tasks that are frequently requested while using personal mobile devices. Evaluation results show high prediction accuracy of our classifier ranging from 82% to 99%. By using large amount of crowdsourced data, we also illustrate the relationship between training data size and the prediction accuracy of our task classifier.

1. Introduction

Artificial intelligence and machine learning has received much attention in our information technology era, and we are observing more and more applications in our daily lives than before. In particular, many industry leaders have developed and introduced top-notch applications based on artificial intelligence [1–5]. These applications include personalized content recommendations and personal assistant services [3–7].

Many advanced personal assistant services heavily depend on natural language understanding (NLU) for human-computer interactions [8–10]. There are also many systems that are based on touch-driven interactions [11, 12]. Nowadays, many machines can interact with humans with a certain level of intelligence, and at the core of them are artificial intelligence algorithms and natural language processing.

There are many unsolved problems of natural language understanding, and the problem of automatically classifying a given natural language input into a suitable task or category is one of them. Many researchers and industry leaders have suggested various algorithms and approaches to tackle the problem [8, 9, 13–20]. These research and development activities later resulted in various personal assistant services such as Apple's Siri, Google's Google Now, and Amazon's Alexa.

However, the personal assistant services that are provided by the industry leaders are proprietary, and their internals and implementations are not well known to the public. As they have been continuously updated and improved over the past several years, we believe that their implementations are highly sophisticated and complicated combinations of many different algorithms and the state-of-the-art technologies. Therefore, we asked ourselves the following question: "Is it possible to implement a personal assistant system that is simple enough to be built by applying a well-known machine learning algorithm and personally crowdsourced data?" By answering this question, we hope that our work motivates many researchers and small

industries to build their own intelligent systems in their particular domains.

With the motivation in mind, in this paper, we introduce our own implementation of an automatic task classification system, which is based on a classical machine learning algorithm and crowdsourcing. Many different classification algorithms have been proposed and introduced to the artificial intelligence and machine learning community. To implement our task classification module, we used the support vector machine (SVM), a popular classification algorithm. In particular, we used the LibShortText library [21], which is an extension of Liblinear [22], a library implementing linear support vector machine algorithms.

Using our implementation, we show that the support vector machine algorithm can be successfully used for building personal assistant services, in particular, task classifiers for mobile devices. This task classifier can take a natural language text input and classify the input text into an implied task category among many predefined tasks. Therefore, it can understand humans' natural language command and execute the intended task accordingly on behalf of the user.

Even though Apple, Google, and Amazon are not disclosing the internal architecture or algorithms that were used to implement their own personal assistant services [3], it is believed that they are making use of a large amount of data that they have collected from various sources to implement their systems. In order to train our classification module, we also collected our own training data, and we describe how we collected our data via crowdsourcing.

By using a large amount of collected training data, we investigate and present a relationship between task classification accuracy of our classifier and training data size. We verify that the more training data we use, the better prediction accuracy we can get, but the performance increase rate drops.

This paper is organized as follows. Section 2 introduces a couple of commercial personal assistant systems and the support vector machine algorithm. Then, an open-source library implementing the support vector machine is briefly introduced. Our classifier uses the library to build a task classifier model. In Section 3, we describe our classifier and the library on which the classifier is built. Section 4 describes our crowdsourcing procedure for collecting training data that are used to train our classifier model. Section 5 shows that our classifier can classify short English texts into implied task categories. In particular, precision and recall values are presented for each task. The relationship between prediction accuracy and training data size is also investigated. In Section 6, we propose an overall architecture of a possible implementation of a personal assistant system, which is based on our task classifier. Finally, Section 7 concludes the paper.

2. Background and Prior Work

Before we propose our task classifier, we introduce some prior work on natural language processing and a couple of personal assistant systems. Then, a brief background on the support vector machine algorithm is introduced.

2.1. Natural Language Processing. Natural language processing is a fairly large research area and has a long history in the computer science community. The following are a few prior work in the field.

In 1972, Winograd tried to implement a computer system that can interact with human beings in English [14]. Kuhn and De Mori tried a new data structure, semantic classification tree, to implement a building block for robust matchers for NLU tasks [9]. Manning and Schütze describe statistical natural language processing in their book [8]. Yi et al. presented a sentiment analyzer that extracts sentiment about a subject using natural language processing techniques [15]. Collobert et al. proposed a unified neural network architecture and learning algorithm that can be applied to various natural language processing tasks including part-of-speech tagging, chunking, named entity recognition, and semantic role labeling [16].

Task or category classification has much prior work, too. In 1994, Cavnar and Trenkle proposed a text classification approach based on N-gram [17]. Yang and Pedersen performed a comparative study on feature selection methods in text categorization [18]. Text categorization via SVM was studied by Joachims in 1998 [19]. Yang and Liu performed a study on five different methods for text classification [23]. Sebastiani summarized many different approaches for text classification based on machine learning [24]. Pang et al. performed a study on sentiment analysis (positive or negative) by employing three different machine learning algorithms: naive Bayes, maximum entropy classification, and support vector machine [25]. Tong and Koller proposed support vector machine active learning for text classification applications [20]. Leopold and Kindermann showed that term-frequency transformations have a larger impact on the performance of SVM for text classifications than the kernel functions [26]. Genkin et al. proposed a new approach based on logistic regression that can handle high-dimensional data such as natural language text [27]. Lan et al. proposed a new term weighting method for text classification [28].

2.2. Personal Assistant Systems. One of the innovative and well-known automatic task classification systems for a natural language input sentence would be Siri, a personal assistant system developed by Apple, Inc. [29]. Soon after Siri was introduced, Google also started providing its own similar service, which is called as Google Now [30, 31].

These two systems can understand humans' natural language command, which means that they can classify a given natural language input to a command implied by the input text and perform the predicted task accordingly. For example, these systems understand a voice input command such as "Call John," and on behalf of the user, they perform automatically the user's intended task, which is a "Call" task.

However, Apple and Google have not disclosed how these systems are implemented. Therefore, we designed and implemented our own automatic task classifier based on a widely used classification algorithm, the support vector machine.

2.3. The Support Vector Machine. There are many classification algorithms that are well known to the machine learning community. In particular, Deep Learning [32–34] is a very hot topic these days. If designed and trained carefully, deep learning algorithms usually outperform (in terms of classification accuracy) most of the previously known classification algorithms such as the support vector machine, random forests, and naive Bayes in many domains. However, in order to train a competitive deep learning network, a very large amount of training data is usually required. Furthermore, deep neural networks are often regarded as black boxes because it is hard to understand how the networks classify test instances into correct categories through the deep network layers.

On the contrary, the support vector machine is seen to be more interpretable than deep neural networks, and it had been mostly used in many classification problems. Even though there is no one universal algorithm that outperforms every other classification algorithms in various domains, the support vector machine was widely used due to its relatively powerful performance over many different areas [35]. Considering our goal of this work, we decided to use the support vector machine algorithm rather than using more contemporary but complicated deep learning algorithms.

Support vector machine algorithm was conceptually invented in 1963 by Vapnik and Chervonenkis [36]. In 1992, Boser et al. proposed a way to create nonlinear classifiers via the kernel trick [37]. A couple years later, Cortes and Vapnik introduced the concept of the soft margin [38]. Since its introduction, SVM has seen many applications such as hand-written character recognition.

There are many implementations of SVM with different optimization algorithms. Fan et al. implemented an open-source library for large-scale linear classification, which is named as Liblinear [22]. Liblinear supports logistic regression and linear support vector machines. Another open-source library for short text classification and analysis, called LibShortText, was implemented [21]. LibShortText is an extension of Liblinear, and it can train a classification model with a given training data set consisting of short natural language texts with labels.

3. Automatic Task Classifier

Our main idea is that a practical task classifier, a core part of personal assistant systems, can be implemented to reach a sufficient accuracy by using a classical classification algorithm and basic natural language processing techniques. As we have briefly introduced in Section 2, there exists an open-source library for text classification based on the support vector machine. Therefore, we adopted this library to design our task classifier instead of reinventing the wheels.

3.1. Predefined Tasks. We implemented our own automatic task classifier that can classify a given natural language input text to the most appropriate task among the thirty-two predefined tasks. The thirty-two distinct tasks that we have used are shown in Table 1. We picked these tasks based on our observation that they would span most frequently used tasks that a user can command their mobile devices such as smartphones or tablets. However, these thirty-two tasks are not meant to be hardcoded; users can define any task list of their own interest.

Table 1: The predefined thirty-two tasks for mobile devices.

Search		App	Chatting
Transportation	Map	Call	Greeting
Travel	Music	E-mail	Praise
Movie	Photo	Camera	Dispraise
Book	News	Check schedule	Boredom
Game	Apps	Schedule	Love
Restaurant	Recipe	Memo	—
Shopping	Weather	Timer	—
Hospital	—	Alarm	—
Wikipedia	—	Music player	—
Information	—	SNS	—

For example, the thirty-two tasks contain the "Call" task, and our task classifier can classify an input text "Call John." In other words, our classifier will automatically recognize the user's intended task, which is the "Call" task, and the personal assistant system will command the mobile device to search its contacts list to find "John" and finally place a call to him. Even though we defined our own thirty-two tasks targeting mobile devices, different use cases can define their own task lists. For an instance, navigation systems may have totally different tasks such as "Search Location" or "Cancel Navigation."

3.2. Training Task Classifier. In order to implement our automatic task classifier, we exploit LibShortText [21], an open-source library implementing a short-text classifier. This library is very well implemented and provides a capability to change the parameters of the support vector machine algorithm or natural language processing. As the authors of LibShortText claimed, our preliminary experimentation has shown that the default parameters of the library result in very good classification accuracies for our purpose, if not the best, so we mostly followed their recommendations as is.

In addition, in an attempt to enhance the accuracy further, we designed a preprocessing step, which is to replace some words into more general categories. For example, if the given sentence is "I want to have a sushi," then the word "sushi" is replaced with "categoryFood." Therefore, the final sentence in this case becomes "I want to have a categoryFood."

We experimented with the idea of word replacement because replacing more specific words with more general terms may increase classification accuracies by decreasing the dimension of the input feature space (the space of N-grams). In order to implement the preprocessing of word replacement, we first created a dictionary, which maps some words to more general categories. For example, "sushi" and "pizza" are mapped to the "categoryFood" category. After we generated the dictionary, we applied the preprocessing step to the training data set. In other words, all the training data

sentences were transformed to sentences where specific words are replaced with corresponding category titles. Of course, when the task classifier classifies a test input sentence, the same preprocessing step should be performed on the test sentence before the classification process kicks in.

To our disappointment, however, the experiment results were not promising; the classification accuracy with the preprocessing was not higher than the case without that step. We believe there may be many reasons for this. First of all, the classification accuracy of the support vector machine is already very high without the preprocessing, which makes it very hard to increase the accuracy further. Second, the feature space dimension is not reduced enough to affect the classification accuracy.

After we confirmed the experiment results, we chose to stay with the recommended setting of LibShortText without the preprocessing. We also want to mention that the preprocessing takes computation time, which is another reason why we chose not to apply our tested preprocessing.

The LibShortText library requires a training data set to train a support vector machine, so we used a popular crowdsourcing platform to collect our training data set. In order to achieve high classification accuracy for general natural language text inputs, we need a large amount of data. The data collection process is described in the next section.

4. Data Collection via Crowdsourcing

This section describes how we collected our own training data set for our task classifier training.

4.1. Amazon Mechanical Turk. Crowdsourcing has been a powerful way to obtain human intelligent services, ideas, or content by soliciting contributions from a large group of people and especially from online communities [39]. A well-known online survey platform, SurveyMonkey, is a good example of many services that can be used for collecting data via crowdsourcing. Many people are now using SurveyMonkey in small scales for personal, academic, or industrial purpose.

Amazon.com, Inc. is also providing a popular and commercial crowdsourcing platform called Amazon Mechanical Turk (MTurk). MTurk provides an easy-to-use system for collecting a large amount of data sets via crowdsourcing. There have been many research results about the data quality collected by MTurk. Buhrmester et al. described and evaluated the potential contributions of MTurk to psychology and other social sciences [40]. Paolacci et al. addressed potential concerns about the quality of collected data through MTurk by presenting new demographic data about the Mechanical Turk subject population, reviewing the strengths of MTurk relative to other online and offline methods of recruiting subjects and comparing the magnitude of effects obtained using Mechanical Turk and traditional subject pools [41]. Kittur et al. also performed a study on validity of MTurk platform [42]. Many of them indicate that MTurk is a good crowdsourcing platform to collect high-quality data with inexpensive monetary costs.

We collected our own training data set using MTurk to train our task classification engine. MTurk enables crowdsourcing requesters to upload their questionnaires. Once uploaded, MTurk publishes the questionnaires to the MTurk open marketplace so that many MTurk workers can answer the uploaded questionnaires and get paid by the requester.

4.2. Our Data Collection Process. In order to collect English sentences for commanding any of the predefined thirty-two tasks, we created a questionnaire that requests a worker to fill out his/her own example sentence for each different task. For example, for the task of "Restaurant Search", one person can provide an example sentence such as "Find me a good Italian restaurant nearby," and another person can provide a different sentence such as "best Korean food in San Francisco." Once a worker finished filling out an example sentence for each of all the thirty-two tasks, then we compensated the worker for their contribution.

Through the aforementioned MTurk crowdsourcing process, we were able to collect 65,890 sentences for the thirty-two tasks. Each task has at least 2,000 sentences. All these sentences are human generated, so the data set has high quality and variety. For example, a worker provided an example sentence, "I am hungry" for the task of "Restaurant Search," whereas many other workers just provided names of various cuisines such as "Pizza" or "Sushi."

5. Prediction Accuracy

In order to train our classifier and evaluate the classification accuracy, we applied the tenfold cross-validation approach using the 65,890 sentences for the thirty-two tasks collected via crowdsourcing. While applying tenfold cross validation, we measured the precision and recall values for each task.

5.1. Precision and Recall. While performing the tenfold cross validation, the precision and recall values for each task can be computed as follows for each fold.

Suppose that we randomly partition all the collected data set T into ten equal folds. We partitioned T so that each fold has roughly equal number of data points for each task. To compute precision and recall values corresponding to a partition P_i, $i = 1, 2, \ldots, 10$, we form a training data set consisting of all the data except those belonging to P_i. In other words, if we call the training data set for the fold P_i as T_i, we have

$$T_i = \{d \in T \mid d \notin P_i\}. \quad (1)$$

By using T_i as a training data set, we train a task classifier model f_i. We then measure the prediction accuracy of the model f_i using the fold P_i as a validation set.

For each pair of task t_j, $j = 1, 2, \ldots, 32$, and fold P_i, $i = 1, 2, \ldots, 10$, the precision and recall values are computed by the following equations:

Table 2: The precision and recall values of each task as well as the top 3 misclassifications.

Task name	Precision (%)	Recall (%)	Top 3 misclassifications		
Transportation	91.48	94.33	Travel 1.61%	Map 1.46%	Restaurant 0.76%
Map	83.87	82.76	Restaurant 5.53%	Shopping 4.49%	Transportation 3.45%
Travel	93.64	94.44	Transportation 0.85%	Wikipedia 0.81%	Restaurant 0.62%
Music	88.98	86.25	Music player 6.33%	Wikipedia 3.07%	Book 1.56%
Movie	89.74	93.10	Book 1.28%	Information 1.18%	Music 1.09%
Photo	95.86	95.13	Information 0.99%	Camera 0.90%	Wikipedia 0.61%
Book	91.21	91.51	Music 1.52%	Wikipedia 1.47%	Shopping 1.09%
News	93.08	94.14	Wikipedia 1.37%	Information 0.85%	Book 0.61%
Game	88.23	89.07	Apps 6.48%	Music player 0.71%	Movie 0.47%
Apps	90.02	88.92	Game 6.92%	Map 0.47%	Movie 0.43%
Restaurant	86.54	88.35	Map 4.31%	Shopping 2.84%	Transportation 1.14%
Recipe	97.14	98.15	Restaurant 0.43%	Information 0.24%	Map 0.14%
Shopping	86.25	88.75	Map 3.99%	Restaurant 2.71%	Hospital 1.04%
Weather	97.90	99.20	News 0.14%	Information 0.14%	Check schedule 0.14%
Hospital	95.55	94.92	Shopping 1.28%	Map 1.05%	Restaurant 0.81%
Wikipedia	83.05	78.63	Information 7.98%	News 3.32%	Music 1.85%
Information	82.51	83.88	Wikipedia 5.61%	Music 1.24%	Book 1.14%
Call	98.57	98.34	Greeting 0.48%	Love 0.24%	Movie 0.14%
E-mail	99.38	99.19	Call 0.14%	Dispraise 0.14%	Memo 0.10%
Camera	98.16	98.72	Photo 0.24%	Apps 0.19%	Movie 0.14%
Check schedule	90.60	92.97	Schedule 4.66%	Boredom 0.48%	Memo 0.38%
Schedule	93.43	91.35	Check schedule 6.51%	Memo 0.90%	Timer 0.14%
Memo	97.57	97.29	Schedule 0.62%	E-mail 0.29%	Timer 0.24%
Timer	97.59	98.24	Alarm 0.76%	Memo 0.24%	Information 0.14%
Alarm	98.38	98.38	Timer 0.90%	Greeting 0.24%	Movie 0.14%
Music player	91.23	96.28	Music 2.62%	Timer 0.14%	Love 0.14%
SNS	98.69	96.36	News 1.09%	Movie 0.57%	Wikipedia 0.43%
Greeting	93.87	92.14	Dispraise 1.51%	Check schedule 1.11%	Praise 1.00%
Praise	89.12	89.52	Dispraise 2.79%	Love 1.95%	Restaurant 1.06%
Dispraise	84.00	83.48	Boredom 3.29%	Praise 3.07%	Shopping 1.79%

Table 2: Continued.

Task name	Precision (%)	Recall (%)	Top 3 misclassifications		
Boredom	89.76	84.14	Dispraise 4.80%	Game 2.79%	Greeting 1.45%
Love	92.40	88.27	Praise 4.41%	Dispraise 2.12%	Boredom 1.68%

Accuracy of our classifier (in terms of precision and recall) is promisingly high ranging from 82% up to 99%.

$$\text{Precision}(t_j, P_i) = \frac{\left|\left\{d \in P_i \mid f_i(d) = t_j, t(d) = t_j\right\}\right|}{\left|\left\{d \in P_i \mid f_i(d) = t_j\right\}\right|},$$
$$\text{Recall}(t_j, P_i) = \frac{\left|\left\{d \in P_i \mid f_i(d) = t_j, t(d) = t_j\right\}\right|}{\left|\left\{d \in P_i \mid t(d) = t_j\right\}\right|}, \quad (2)$$

where $t(d)$ and $f_i(d)$ are the original and the predicted task labels of the data sentence d, respectively.

5.2. Measured Prediction Accuracy. The tenfold cross-validation results are shown in Table 2. As shown in Table 2, the precision and recall values are very high ranging from 82% up to 99%. Even though there are some misclassifications between similar tasks, the overall prediction accuracy is promising. Therefore, our task classifier can be employed to build practical personal assistant services.

We can also observe that only a few groups of similar tasks have relatively high misclassification ratios. For example, the three tasks of "Map," "Restaurant," and "Shopping" search were mostly confused with each other. The "Music" search and "Music player" app launch tasks were also mostly confused with each other. The "Apps" search and "Game" app launch tasks were mostly confused with each other, and the "Wikipedia" and "Information" search tasks were also confused with each other. Finally, the five chatting tasks were confused among themselves.

In order to increase the accuracy of our task classifier further, we may build a second layer of classification, which is applied to and classifies each group of similar tasks to a more accurate task inside the group. The second layer need not use the support vector machine as the first layer; it may use rule-based classifiers or any other algorithm.

5.3. Accuracy and Training Data Size. We performed an experiment to investigate the relation between training data size and classification accuracy. It is expected that the more the data we use for training, the more the accuracy we can get. We wanted to confirm this expectation and to get the precise relation between the classifier performance and the training data size. For this experiment, we randomly chose 20% of all the collected data points as the test set. Then, we used the remaining 80% of the data set as a training data pool, so that we can sample a certain amount of data points randomly from the pool.

More specifically, suppose that S is the randomly sampled test set whose size is 20% of the collected data set T. Then, the remaining 80% of the collected data is used as the training data pool P, from which we sample different sizes of training data. Therefore, we have

$$P \cup S = T,$$
$$P \cap S = \varnothing. \quad (3)$$

In particular, we tested with ten different sizes of training data: 10%, 20%, 30%, ..., 100% of the training data pool P. For each different training data size of $(i/10)|P|$ for $i = 1, 2, 3, \ldots, 10$, we repeated random sampling from the pool P ten times to train ten different classifier models with the same training data size. In other words, for each i, we train ten different task classifiers f_{ij} for $j = 1, 2, 3, \ldots, 10$ by sampling training data of the same size from the pool P ten times. Then, using each classifier f_{ij}, we measured the classification accuracy acc_{ij} as

$$\text{acc}_{ij} = \frac{\left|\left\{d \in S \mid f_{ij}(d) = t(d)\right\}\right|}{|S|}, \quad (4)$$

where $f_{ij}(d)$ and $t(d)$ are the predicted and original task of the input text d, respectively. Therefore, we have ten different accuracy values for each training data size.

The results of the experiment are shown in Figure 1, where a boxplot is generated with ten different accuracy values for each different training data size. The figure shows that the classifier performance increases as the training data size increases, but the performance increase rate drops as the training data size increases. This result is reasonable and verifies our original hypothesis.

6. Personal Assistant Systems

So far, we have described our proposed task classifier that is based on the support vector machine algorithm. The proposed task classifier may be used to build a personal assistant system. Figure 2 shows the overall architecture of a possible implementation of a personal assistant system. A user's voice command is first transmitted to a server, which is running the speech-to-text converter and the task classifier.

For the speech-to-text converter, any suitable converter may be used; for example, we may use Sphinx speech recognition system [43–45].

The received speech at the server is converted to text, and the converted text is given to the task classifier, a core part of the overall system. The task classifier classifies the given text into the most probable task, and the predicted task is transmitted back to the user's mobile device. Then, the task launcher of the personal assistant system performs the predicted task accordingly on behalf of the user.

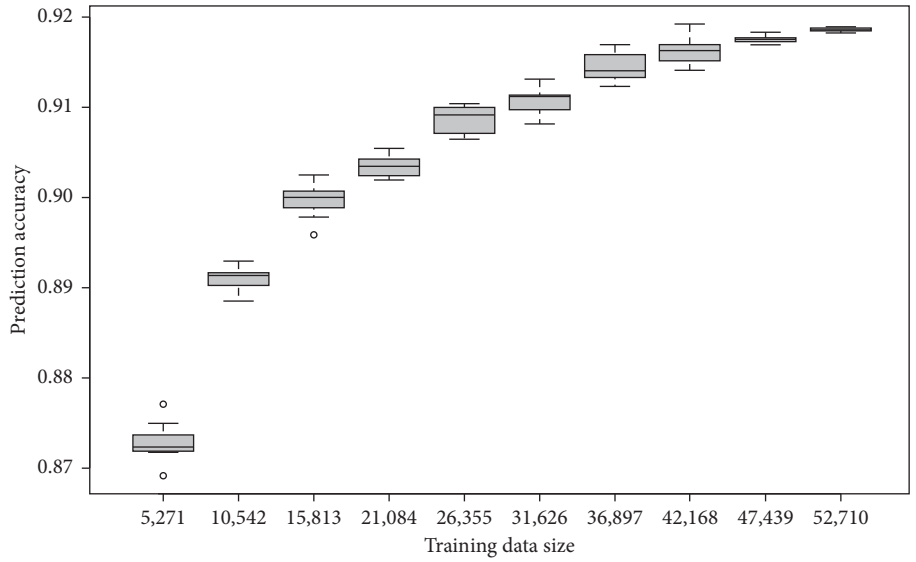


Figure 1: Prediction accuracy increases as we use more training data, but the increase rate drops.

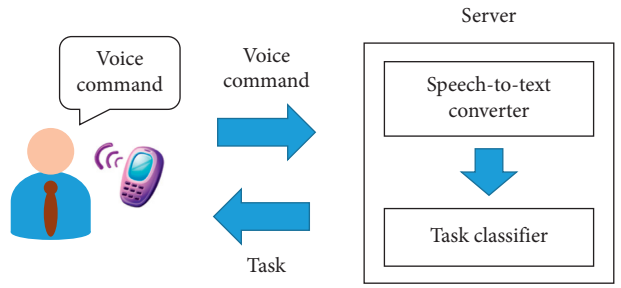


Figure 2: A proposed architecture of a personal assistant system.

Since the proposed task classifier is built upon the linear support vector machine, the computation time typically ranged from a few milliseconds to a few tens of milliseconds. Of course, as we use more powerful servers, we can reduce the computation time more.

7. Conclusion

We presented a method to implement personal assistant services that can understand human's natural language commands. Even though there already exist such services including Siri, Google Now, and Alexa, the internal technologies have not been disclosed and are not well known to the public. Therefore, we investigated whether it is possible to build a task classifier, a core part of personal assistant services, using a well-known machine learning algorithm. Our implementation is based on the support vector machine, a widely used classification algorithm in many domains.

To train our support vector machine with sufficient data, we collected our own training data set by using a popular crowdsourcing platform, the Amazon Mechanical Turk. We predefined thirty-two tasks that are frequently commanded while using mobile devices and collected natural language sentences for each task by using the crowdsourcing platform. Through this process, we were able to collect 65,890 natural language sentences in total.

We tested our task classifier performance with the tenfold cross-validation approach. The evaluation results show that the precision and recall values of our classifier are very high. This result indicates that our simple approach can be employed to implement practical personal assistant services.

The relationship between training data size and classifier performance was also investigated. We confirmed that the performance becomes better as we use more training data, but the performance increase rate drops as we increase the training data size. All of these observations are reasonable, and we hope that our work can motivate and be referred to by many researchers or industries who are trying to build their own personal assistant systems.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported by 2017 Hongik University Research Fund, in part by the National Research Foundation of

Korea (NRF) grant funded by the Korea government (Ministry of Science, ICT & Future Planning (MSIP)) (no. 2017R1C1B5015901) and also in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B03031348).

References

[1] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[2] J. B. Greenblatt and S. Saxena, "Autonomous taxis could greatly reduce greenhouse-gas emissions of US light-duty vehicles," *Nature Climate Change*, vol. 5, no. 9, p. 860, 2015.

[3] J. Aron, "How innovative is Apple's new voice assistant, Siri?," *New Scientist*, vol. 212, no. 2836, p. 24, 2011.

[4] D. Sullivan, "Google Now personalizes everyone's search results," *Search Engine Land*, vol. 12, 2009.

[5] A. Warren, *Amazon Echo: The Ultimate Amazon Echo User Guide 2016 Become an Alexa and Echo Expert Now!*, 2016.

[6] D. Yang, D. Zhang, Z. Yu, and Z. Wang, "A sentiment-enhanced personalized location recommendation system," in *Proceedings of the ACM Conference on Hypertext and Social Media*, pp. 119–128, Paris, France, May 2013.

[7] B. Guo, D. Zhang, D. Yang, Z. Yu, and X. Zhou, "Enhancing memory recall via an intelligent social contact management system," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 1, pp. 78–91, 2014.

[8] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, vol. 999, MIT Press, Cambridge, MA, USA, 1999.

[9] R. Kuhn and R. De Mori, "The application of semantic classification trees to natural language understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 449–460, 1995.

[10] W. Cui, S. Liu, L. Tan et al., "Textflow: towards better understanding of evolving topics in text," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2412–2421, 2011.

[11] Y. Xu, Z. Ye, L. Chen, S. Li, and G. Pan, "TaskShadow-w: NFC-triggered migration of web browsing across personal devices," in *Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp'13)*, pp. 99–102, Zurich, Switzerland, September 2013.

[12] L. Chen, G. Pan, and S. Li, "Touch-driven interaction via an NFC-enabled smartphone," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 504–506, Lugano, Switzerland, March 2012.

[13] T. M. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, "Experience with a learning personal assistant," *Communications of the ACM*, vol. 37, no. 7, pp. 80–91, 1994.

[14] T. Winograd, "Understanding natural language," *Cognitive Psychology*, vol. 3, no. 1, pp. 1–191, 1972.

[15] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 427–434, Melbourne, FL, USA, November 2003.

[16] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[17] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," *Ann Arbor Journal*, vol. 48113, no. 2, pp. 161–175, 1994.

[18] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," *ICML*, vol. 97, pp. 412–420, 1997.

[19] T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, Springer, Berlin, Germany, 1998.

[20] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[21] H. Yu, C. Ho, Y. Juan, and C.-J. Lin, *LibShortText: A Library for Short-Text Classification and Analysis*, Rapport Interne, Department of Computer Science, National Taiwan University, Taipei, Taiwan, 2013.

[22] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: a library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[23] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 42–49, Berkeley, CA, USA, August 1999.

[24] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[25] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, vol. 10, pp. 79–86, Philadelphia, PA, USA, July 2002.

[26] E. Leopold and J. Kindermann, "Text categorization with support vector machines. How to represent texts in input space?," *Machine Learning*, vol. 46, no. 1–3, pp. 423–444, 2002.

[27] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale bayesian logistic regression for text categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, 2007.

[28] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 721–735, 2009.

[29] T. R. Gruber, A. J. Cheyer, D. Kittlaus, et al., "Intelligent automated assistant," US Patent 9 318 108, 2016.

[30] S. Agarwal, D. Nishar, and A. Rubin, "Providing digital content based on expected user behavior," US Patent 8 271 413, 2012.

[31] S. Agarwal, V. Gundotra, and A. Nicolaou, "Providing results to parameterless search queries," US Patent 8 478 519, 2013.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[35] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1-2, pp. 169–186, 2003.

[36] V. N. Vapnik and S. Kotz, *Estimation of Dependences Based on Empirical Data*, Springer-Verlag, Vol. 40, Springer-erlag, New York, NY, USA, 1982.
[37] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, Pittsburgh, PA, USA, July 1992.
[38] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[39] Merriam-Webster.com, "Entry:crowdsourcing," 2012.
[40] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's Mechanical Turk: a new source of inexpensive, yet high-quality, data?," *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.
[41] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on Amazon Mechanical Turk," *Judgment and Decision making*, vol. 5, no. 5, pp. 411–419, 2010.
[42] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with Mechanical Turk," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 453–456, Florence, Italy, April 2008.
[43] K.-F. Lee, *Automatic Speech Recognition: the Development of the SPHINX System*, Springer Science & Business Media, vol. 62, Berlin, Germany, 1988.
[44] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech & Language*, vol. 7, no. 2, pp. 137–148, 1993.
[45] W. Walker, P. Lamere, P. Kwok et al., "Sphinx-4: a flexible open source framework for speech recognition," Technical Report, Mountain View, Sun Microsystems Inc., CA, USA, 2004.

Hindawi
Mobile Information Systems
Volume 2018, Article ID 6462826, 8 pages
https://doi.org/10.1155/2018/6462826

Research Article

EpSoc: Social-Based Epidemic-Based Routing Protocol in Opportunistic Mobile Social Network

Halikul Lenando[1] **and Mohamad Alrfaay**[2]

[1]*Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, Malaysia*
[2]*Faculty of Computer and Information Sciences, Jouf University, Jouf, Saudi Arabia*

Correspondence should be addressed to Halikul Lenando; cool@unimas.my

Received 17 October 2017; Revised 20 February 2018; Accepted 11 March 2018; Published 4 April 2018

Academic Editor: Fabio Gasparetti

In opportunistic networks, the nature of intermittent and disruptive connections degrades the efficiency of routing. Epidemic routing protocol is used as a benchmark for most of routing protocols in opportunistic mobile social networks (OMSNs) due to its high message delivery and latency. However, Epidemic incurs high cost in terms of overhead and hop count. In this paper, we propose a hybrid routing protocol called EpSoc which utilizes the Epidemic routing forwarding strategy and exploits an important social feature, that is, degree centrality. Two techniques are used in EpSoc. Messages' TTL is adjusted based on the degree centrality of nodes, and the message blocking mechanism is used to control replication. Simulation results show that EpSoc increases the delivery ratio and decreases the overhead ratio, the average latency, and the hop counts as compared to Epidemic and Bubble Rap.

1. Introduction

Opportunistic mobile social network (OMSN) [1–4] is a promising networking model for data dissemination. In the OMSN, mobile nodes grab the opportunity of encountering the peer (they are in the communication range of each other) to forward the data. The OMSN incurs intermittent and disruptive connectivity due to node mobility. To tackle with this complex environment, the store-carry-forward scheme is applied in the OMSN. If no connection is available at a particular time, a mobile node stores data in its buffer and carries them until it encounters other mobile nodes to forward the data [5–8]. Various approaches have been proposed to address the information delivery problem in the OMSN such as in [9–12]. The main concerns of OMSN routing approaches are yielding high delivery ratio, low delay, and low overhead or cost on networks and nodes.

Flooding is one of the dominant schemes to disseminate data in the OMSN [13]. Each message will be flooded to every node in the network. Multiple copies of each message are generated and spread in the network. Epidemic [9] routing protocol is the flooding-based routing protocol. When two nodes encounter, they exchange all of their messages. This results in messages spread over the whole network by pairwise contacts between two nodes. If no buffer constraints are applied, Epidemic represents the upper bound in message delivery and latency. Epidemic routing is used as a benchmark and a reference for the most of other routing protocols in the opportunistic network. The main drawback of the Epidemic scheme is its high overhead. Many schemes are proposed to decrease the overhead in Epidemic-based approaches by limiting the number of message replicas [14–16]. An effective scheme to control replication spread is the vaccine [17]. It applies the antipacket mechanism to control replica distribution in Epidemic-based routing. In [18], a new scheme is proposed to control the replication of epidemically distributed information. Based on the vaccine scheme, signal distribution is early controlled by the fully immunized vaccine. In addition, a partially immunized vaccine is initiated when there is a local-forwarding opportunity to vaccine more packets.

In the OMSN, mobile devices are portable by humans so that social features of people can be exploited for networking purposes [19–21]. Social-based protocols utilize social properties of mobile users such as similarity, centrality, and

friendship to improve routing efficiency in the opportunistic mobile social network. This is because social features are more stable and less changeable than other features like mobility patterns. HiBOp [22] and CiPRO [23] exploit the similarity social feature and the user's context information to forward data. LASS [24] takes into account the difference of members' activity within the node's community for data dissemination. In ML-SOR [25], node centrality (different types of centralities), the similarity between communities, and social ties are all exploited to effectively select the forwarding node. MCAR [26] exploits the preferred communities of people during their daily lives for effective information delivery. Direct (inside one community) and indirect (via different communities) contacts are considered in MCAR. In SPRINT-SELF [27], network and node overhead is decreased by exploiting social information of mobile users. The authors consider the social community of nodes and utilize it to predict future behavior based on contact history. In addition, they proposed a new mechanism to avoid selfish nodes for more improvements.

Social features are utilized widely for buffer management. Liu et al. [28] utilized social features and the congestion level to develop the forwarding strategy that drops the message with the minimum social link rather than random dropping. In SRAMSW [29], the buffer management mechanism is combined with social features to enhance the spray-based routing. Expired messages are deleted, successfully delivered messages are acknowledged, and messages are prioritized according to their spray times and residence time. In addition, three social features: centrality, similarity, and friendship, are adopted for better forwarding decision and to avoid the dead-end problem.

We hypothesize that combining social features with the Epidemic-based forwarding scheme improves the efficiency of routing in the OMSN. In this paper, we present an Epidemic-based Social-based routing protocol (EpSoc) that combines the advantages of the forwarding strategy used in the Epidemic routing protocol with the positive impact of exploiting social features. EpSoc exploits the degree centrality social feature to adapt the time to live (TTL) of the routed message. If a message is forwarded to a node which has higher degree centrality (socially active node), its TTL value will be decreased. If these messages are dropped in the active node when TTL is zero, the blocking mechanism is used to reject receiving replications of these messages.

The rest of this paper is organized as follows: the next section reviews the related work. We describe in detail our proposed algorithm EpSoc in Section 3. In Section 4, we introduce the performance evaluation and the discussion results. Finally, Section 5 concludes the paper.

2. Related Works

Delivering data to the destination at the right time with minimum resources is an optimal condition for any given routing protocols. Epidemic has optimal performance in terms of delivery ratio and latency. However, it suffers from high overhead cost. One of the solutions is to exploit social information to improve Epidemic-based routing protocols in the OMSN. Degree centrality is a social feature that is exploited widely in the literature to improve routing in the mobile social networks. For example, Bubble Rap [11] utilizes the mobile node's degree centrality to provide cost-effective routing as compared to Epidemic routing. Bubble Rap exploits two social and structural metrics, namely, centrality and community. It selects high centrality nodes and community members of destination as relays. In the Bubble Rap algorithm, nodes belong to different sizes of communities and have different levels of popularity (i.e., rank). Each node is assumed to have two rankings: global denotes the popularity (i.e., connectivity) of the node in the entire society and local denotes the popularity within its community. Messages are forwarded to nodes that have higher global ranking until a node in the destination's community is found. Then, the messages are forwarded to nodes having a higher local ranking within the destination's community.

Similar to Bubble Rap, CAOR [30] exploits degree centrality and similarity social features to improve routing. However, CAOR constructed autonomous communities based on common interest locations between mobile nodes. Members of a community with high centralities act as the home of this community. CAOR also has a mechanism to turn the routing between lots of nodes to the routing between a few community homes. It also applies the reverse Dijkstra algorithm to determine the optimal relays and compute the minimum expected delivery delay. In [31], cultural algorithm (CA), ant colony optimization (ACO), and social connectivity between users are combined to address the routing problem. Social metrics of nodes including degree and betweenness centralities are analyzed to support forwarding decision in the opportunistic network environment.

Besides exploited in the routing problem, social features were also applied to solve different type of problems. For example, works in [32] exploit social features (degree and betweenness centralities) to solve the throwbox placement problem based on the given graph. A user's degree is equal to the total number of its neighbors, and betweenness is the total number of shortest paths passing through the node. The work also introduced the concept of location degree to measure how many mobile users have the location as one of their top visited locations and the location betweenness to show how important the location is for the entire social graph. Social metrics such as degree centrality, social activeness, and community acquaintance are also applied to enhance data delivery in VSNs in [33]. The social interactions between nodes where nodes of similar interests or nodes belonging to the same community have greater probability to encounter each other. Social centrality is also utilized for congestion control in the postdisaster environment [34].

Unlike the aforementioned works, instead of considering community or homing structuring, our proposed protocol is based on the flooding-based forwarding strategy of Epidemic. This is because we intended to design our forwarding protocol to have high delivery ratio and low delay as the Epidemic-based forwarding strategy. Moreover, we also aimed to avoid extra time required to form and maintain the community structure. Thus, to decrease the overhead, we utilized degree centrality and adapted the message's TTL (Time to live) to control the forwarding nodes' activity. The

message's TTL is considered in literature when developing efficient routing protocols. Miao et al. [33] proposed the adaptive multistep routing protocol for mobile delay-tolerant networks (MDTNs). Their aim is to get the balance between the delay and cost of message delivery. The time to live of the message is used in order to allocate the minimum number of copies necessary to achieve a given delivery probability. In [34], the authors considered the contact information of nodes and the time to live message's property to make route decision and improve performance. They built a replica distribution criterion between two encountered nodes based on residual messages' TTL. In our proposed solution, we do not limit the number of replicas but allow messages to be spread in the network and then we utilize degree centrality to decrease the TTL and consequently decrease the overhead. Also, the blocking mechanism is adopted in the active node to cancel receiving replications of the same message which is seen previously.

3. EpSoc Routing Protocol

EpSoc is the routing algorithm designed to decrease overhead in the Epidemic protocol by embedding social features in routing the message in the opportunistic network. To achieve this objective, we propose two mechanisms. First, the message's TTL is adapted based on degree centrality of the nodes in the OMSN. The second one is the message blocking mechanism that is used to prevent receiving the replications of runout TTL messages in active nodes.

3.1. Degree Centrality. Node centrality indicates the popularity of the node in the network, whereby node centrality in a social network is the reflection of its social relative importance [35]. A higher node degree centrality means the node connects with many numbers of nodes in the network. A degree centrality for a given node i can be calculated as follows:

$$\mathrm{DC}_i = \sum_{k=1}^{N} a(i,k), \tag{1}$$

where N is the number of nodes in the network and $a(i,k) = 1$ if a direct link exists between node i and node k and $i \neq k$.

We adopt the CWindow [11] calculation algorithm to calculate degree centrality. CWindow divides the day into time windows and calculates the average of nodes' degree over these windows to estimate the node's centrality. To decrease the processing overhead results from processing any changes of degree centrality, CWindow calculates the node's degree centrality at regular intervals instead of every time the centrality changes.

We pick the CWindow algorithm to calculate degree centrality because it takes into account the changes in node centrality over time and averages the node's centralities for few window intervals which is suitable for people's behavior in the OMSN. It is also used by other social-based protocols that consider the node's degree centrality such as Bubble Rap [11], Dlife [36], and SCORP [37].

3.2. EpSoc Forwarding Strategy. Figure 1 depicts the forwarding process in EpSoc, where two mechanisms are applied.

In Figure 1(a), node $N1$ encountered three nodes: $N2$, $N3$, and $N4$. $N2$ has higher degree centrality (DC) than $N1$. The TTL value of the forwarded messages to $N2$ is decreased by dividing it by the DC value of $N2$. We call these messages socially infected messages. Node $N2$ registers the ID of these messages as seen messages in its blocking register. Node $N2$ will drop the socially infected message when it expires. In Figure 1(b), the Epidemic forwarding strategy which we adopted in our algorithm causes replications of socially infected messages to be sent to node $N2$ from other nodes such as $N4$. In this case, node $N2$ will reject receiving the message with the reason that it is a seen message (its ID is stored in the block register).

Based on (1), the TTL value is adapted using the following equation:

$$\mathrm{TTL}_{\mathrm{new}} = \frac{\mathrm{TTL}_{\mathrm{old}}}{\mathrm{DC}_k} \quad \{\text{if } \mathrm{DC}_k > \mathrm{DC}_i\}. \tag{2}$$

The two combined mechanisms adopted in EpSoc enhance routing performance. If a node is socially active, it meets more nodes in the network and has a higher potential to deliver more messages. Decreasing messages' TTL value in active nodes results in releasing space in their buffers and increases the ability to deliver more messages. This will increase the delivery ratio. The blocking mechanism controls the replications by preventing decreased TTL messages from hitting again the previously traversed actives nodes. The result is decreasing network overhead. Regarding average latency, the messages that are delivered by active nodes have a shorter end-to-end delay compared to other messages delivered by lower activity nodes, so average latency will be decreased too. We conclude that decreasing message life socially by combining the message blocking scheme affects positively the performance of the Epidemic routing scheme in the OMSN.

3.3. Pseudocode of the EpSoc Message Forwarding. We used CWindow to calculate the centrality of the node. Each node N_i records the encountered nodes in the network. When node N_i encounters N_j, the degree centrality values are calculated using CWindow. Then, centrality values are exchanged between both nodes. N_i compares its centrality value DC_i with the N_j centrality value DC_j. If DC_j is greater than DC_i, which means that N_j is socially more active than N_i, then for each message m_i in the N_i buffer Buf_{N_i}, the TTL value $m_{i_{\mathrm{TTL}}}$ is decreased by dividing it by the node N_j centrality value DC_j. The ID of each socially infected message is registered in the N_j blocking register Block_{N_j}. The time complexity of the EpSoc forwarding algorithm is $\mathrm{O}(M)$, where M is the number of messages carried in the node's buffer and needed to be sent or forwarded.

Algorithm 1 shows the complete pseudocode of EpSoc.

4. Performance Evaluations

4.1. Data Set. We adopt the Cambridge experimental data set of haggle [38]. This data set includes traces of Bluetooth

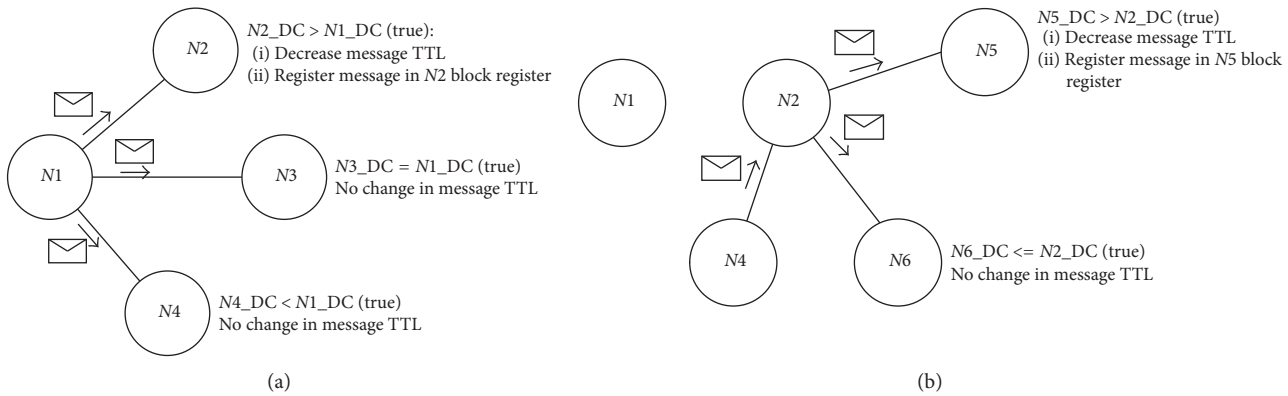


Figure 1: EpSoc forwarding scheme.

(1) **For** all $N_i \in N$
(2) **if** N_i encounter N_j
(3) Calculate DC_i, DC_j
(4) $DC_i \leftrightarrows DC_j$
(5) **For** all $m_i \in Buf_{N_i}$
(6) **if** $m_{i_{ID}} \notin Block_{N_j}$
(7) **if** $DC_j > DC_i$
(8) $m_{i_{TTL}} \leftarrow m_{i_{TTL}}/DC_j$
(9) $Buf_{N_j} \leftarrow m_i$ forward message to N_j
(10) $Block_{N_j} \leftarrow m_{i_{ID}}$
(11) **End if**
(12) **End if**
(13) **End for**
(14) **End if**
(15) **End for**

Algorithm 1: Pseudocode of the forwarding strategy in EpSoc.

sightings by groups of users carrying small devices (iMotes) for a number of days in campus environments. The experiments are conducted in the computer laboratory that includes the undergraduate first-year and second-year students and also some Ph.D. and postgraduate students which lasted for 11 days.

4.2. Simulation Setup. We use the opportunistic network environment (ONE) [39] simulator to evaluate our algorithm. Also, comparison with Epidemic and Bubble Rap routing protocols is included to measure the performance of our proposed algorithm EpSoc. We want to justify that the Epidemic algorithm performs better when considering social features in forwarding messages. The simulator settings are tabulated in Table 1.

In each experiment, we compare the performance of the protocols EpSoc, Epidemic, and Bubble Rap based on the following metrics.

4.2.1. Successful Delivery Ratio. It is the ratio between the number of delivered messages and the total number of created messages. The ideal value of the successful delivery ratio is 1.0 when all created messages are delivered to their destinations.

Table 1: Simulation settings.

Simulation time	987529 seconds
Interface	Bluetooth interface
No. of nodes	36
Transmit speed	250k (2 Mbps)
Mobility	Real trace data (Cambridge)
Buffer size	1, 5, 15, 25, 35, 45, and 55 MB
Routing protocols	Epidemic, EpSoc, and Bubble Rap
Message size	128k
Event interval	30 to 40 seconds
Initial message TTL	10 m, 30 m, 1 h, 3 h, 5 h, 12 h, 1 d, 1.5 d, 2.5 d, 3 d, 4 d, and 1 w

4.2.2. Overhead Ratio. It is the additional bytes that are sent for successfully delivering a message to a destination.

4.2.3. Average Latency. It is the average of the time elapsed between message creation and delivery.

4.2.4. Average Hop Count. It is the average of the number of hops that messages must take in order to reach the destination.

4.3. Experiments and Discussion. To evaluate our work, we will consider two features: buffer size and message TTL value. These two features have a high impact on routing performance in the OMSN. In our work, we affect these two features. We adjust the TTL value socially and use the blocking mechanism to manage buffer storing. Consequently, our experiments are to measure the performance of EpSoc when varying the TTL and buffer size. The comparison will be with Epidemic and Bubble Rap protocols.

4.3.1. Varying Buffer Size. For varying the buffer size, we fixed the value of TTL to 2.5 d. Figures 2–5 show the performance comparisons between EpSoc, Epidemic, and Bubble Rap in terms of delivery ratio, overhead ratio, average latency, and average hop count, respectively.

Figure 2 shows the delivery ratio with buffer size. Generally, for Epidemic, Bubble Rap, and EpSoc, increasing the

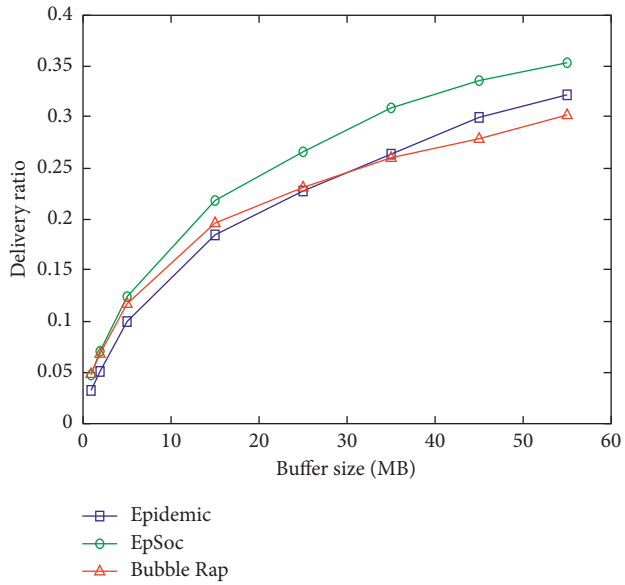


Figure 2: Delivery ratio versus buffer size.

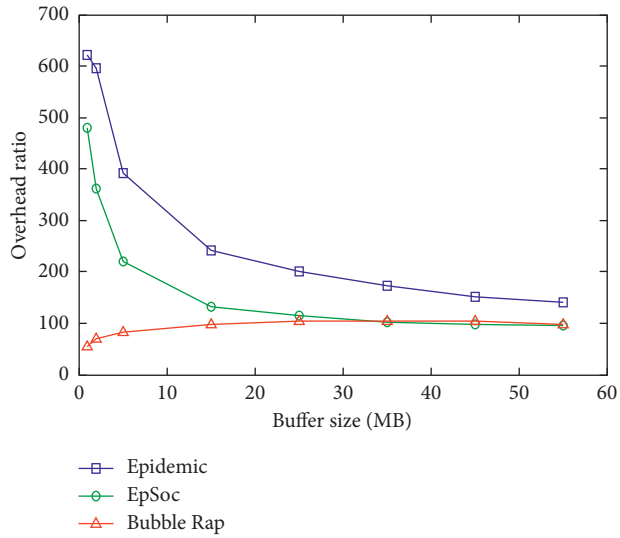


Figure 3: Overhead ratio versus buffer size.

buffer size will increase the delivery ratio. This is because more messages can be carried by intermediate nodes which consequently deliver more messages to destinations. Changing the number of delivered messages affects the delivery ratio, overhead ratio, average latency, and average hop count. Regarding the delivery ratio, it is clear that delivering more messages results in a higher value, while the decrease in delivered messages results in a lower value. In the lower buffer size scenario (1–25 MB), the delivery ratio of Epidemic is the lowest because of the redundancy. Bubble Rap and EpSoc outperform Epidemic due to utilizing social features. In higher buffer size scenarios (35–55 MB), Epidemic achieves higher delivery ratio than Bubble Rap. Larger buffer size alleviates the negative impact of dropping messages because of replication and enables Epidemic to deliver more messages. Our protocol EpSoc outperforms both Epidemic and Bubble Rap. Blocking runout TTL messages from being received by active nodes results in more space in their buffer and therefore can carry more different messages when encountering other nodes and later deliver them to destinations. In addition, the decrease of TTL of the messages that are forwarded to the more active nodes results in better utilization of the buffer's space. Decreased TTL message copies are dropped earlier enabling carrying more other messages. Active nodes deliver the message quickly. Therefore, the number of delivered messages is increased, and consequently, the delivery ratio is increased.

The relation of overhead ratio with buffer size is shown in Figure 3. When increasing the buffer size, the overhead is decreased for all algorithms. Regarding Bubble Rap, overhead is decreased with buffer increase because no replication exists. We observe from Figure 3 that both protocols Bubble Rap and EpSoc outperform Epidemic significantly and EpSoc outperforms Epidemic. We mentioned formerly that applying the strategy of blocking messages and decreasing TTL increase the number of delivered messages. In addition, the blocking message to be resent to active nodes decreases the number of replications in the network and hence decreases the forwardings. As a result, decreasing the forwardings and increasing the delivered messages decrease the overhead ratio in the network. Bubble Rap achieves lower overhead than EpSoc for lower buffer size scenarios (1, 2, 5, and 15 MB). This is because of two reasons: first, the replication and Epidemic strategy adopted in EpSoc, and second, low buffer size causes dropping relayed messages early due to buffer overflow which in turn decreases the efficiency of our proposed algorithm compared to Bubble Rap in terms of overhead. However, for larger buffer size scenarios (25, 35, 45, and 55 MB), EpSoc is more efficient and its overhead ratio is very close to that of Bubble Rap (for 35, 45, and 55 MB, it is slightly better than Bubble Rap).

Figure 4 shows that average latency for all the three protocols, which increases when the buffer size is increased. A low-sized buffer only relays low-latency messages which will reach their destinations quickly. On the other hand, a higher-sized buffer allows messages to be carried for a longer time which contribute to a higher average latency. EpSoc decreases the average latency significantly. EpSoc optimizes the buffer usage where messages forwarded to active nodes have low TTL. With the low TTL, a relay node has more free space for new messages in its buffer as the buffer quickly dropped the old messages.

In Figure 5, the average hop count is recorded. The average hop count increases when the buffer size is increased. Epidemic has more hop count which indicates more nodes experiencing the duplicated messages, whereas Bubble Rap has lower hop count because it prevents replication of messages. For EpSoc, the number of hop count is contributed by allowing replication as in Epidemic.

The worst achievement is because of redundancy. Bubble Rap does not apply replication so that it outperforms EpSoc. EpSoc achieves better average hop count than Epidemic. This is because of the message blocking strategy which decreases the number of replications in the network and hence decreases the forwardings.

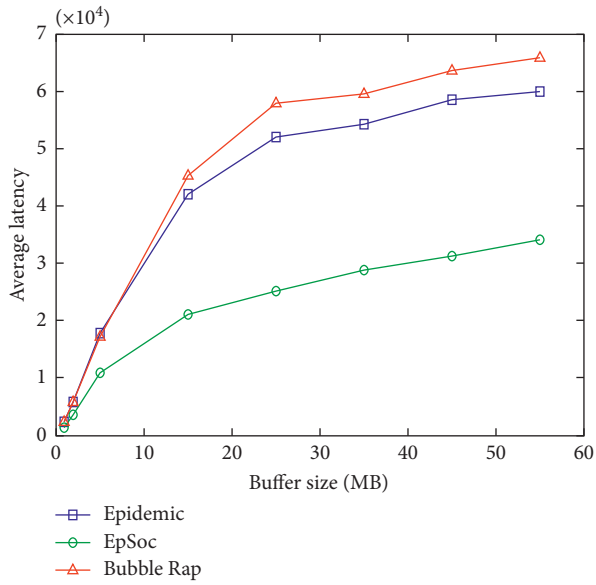


Figure 4: Average latency versus buffer size.

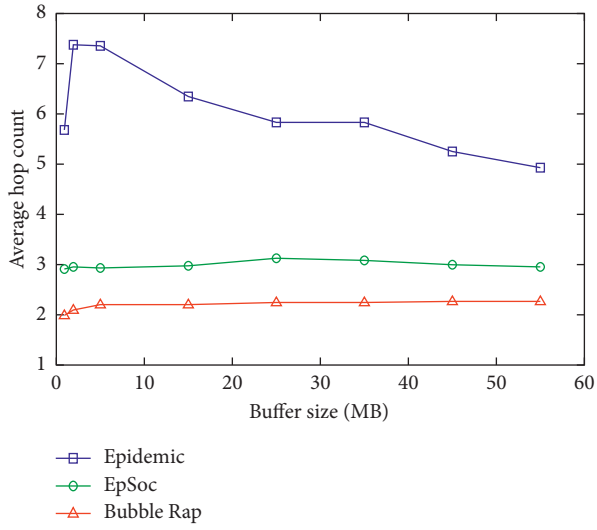


Figure 5: Average hop count versus buffer size.

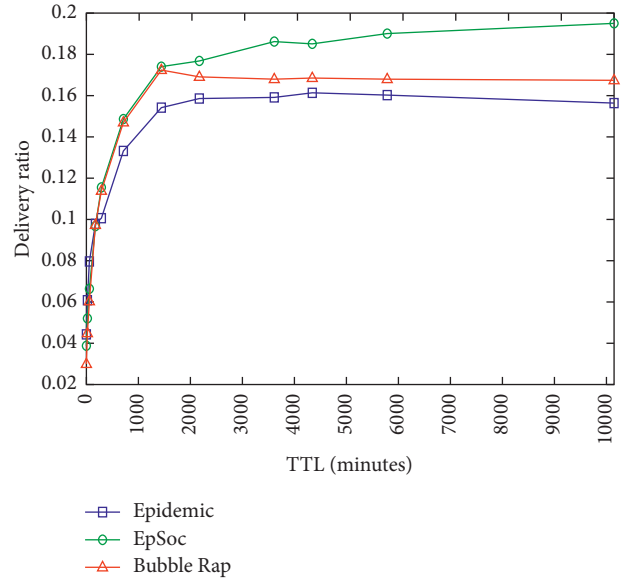


Figure 6: Delivery ratio versus TTL.

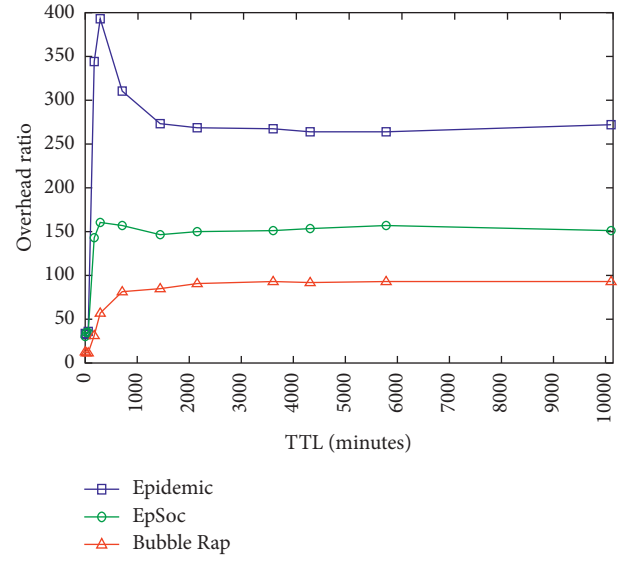


Figure 7: Overhead ratio versus TTL.

4.3.2. Varying Initial TTL. TTL determines the life of the message in the network. A high value of TTL gives high chances of the message delivered to the target destination and vice versa.

In Figure 6, the relation of delivery ratio with TTL is depicted. For lower TTL (10 m–3 h), the delivery ratio of Epidemic is slightly higher than that of Bubble Rap and EpSoc. The reason is that, for short TTL messages, exploiting social features will not be very effective due to quickly messages dropping. In addition, the higher replication occurred in Epidemic increases the number of delivered messages. When TTL is increased, Bubble Rap and EpSoc outperform Epidemic due to utilizing the social features. In high TTL scenarios (1.5 d–1 w), EpSoc outperforms Bubble Rap. This indicates an efficiency of EpSoc in its social feature selection.

Shortening the life of messages in the active nodes and blocking runout TTL messages from hitting again the active nodes cause an increment in the delivered messages and decrement in forwardings. Therefore, delivery ratio grows up.

Figure 7 compares the performance of the protocols with different values of TTL in terms of overhead ratio. When TTL is very low (10 m–1 h), Epidemic, EpSoc, and Bubble Rap achieve low overhead. The reason is that messages are dropped quickly. For high TTL values, Bubble Rap achieves the best and Epidemic the worst.

Our protocol EpSoc manages to decrease the overhead better than Epidemic. This is because EpSoc is a combination of the flooding-based forward strategy with social features.

From Figure 8, in terms of average latency, EpSoc appears to be outperforming others especially with high TTL

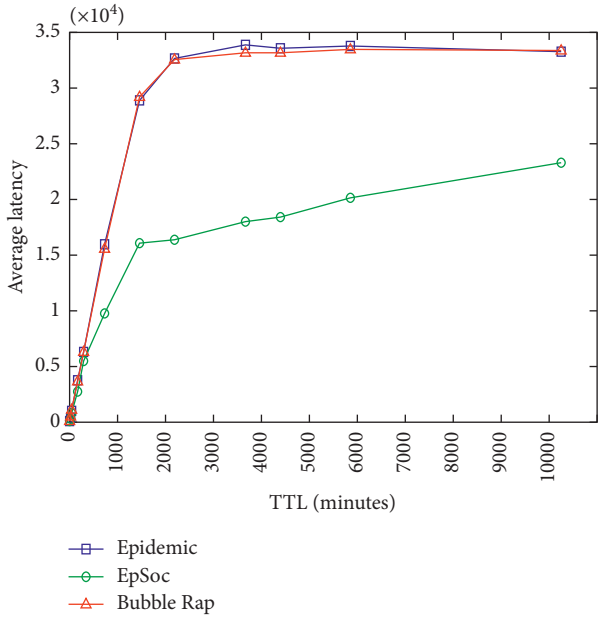


Figure 8: Average latency versus TTL.

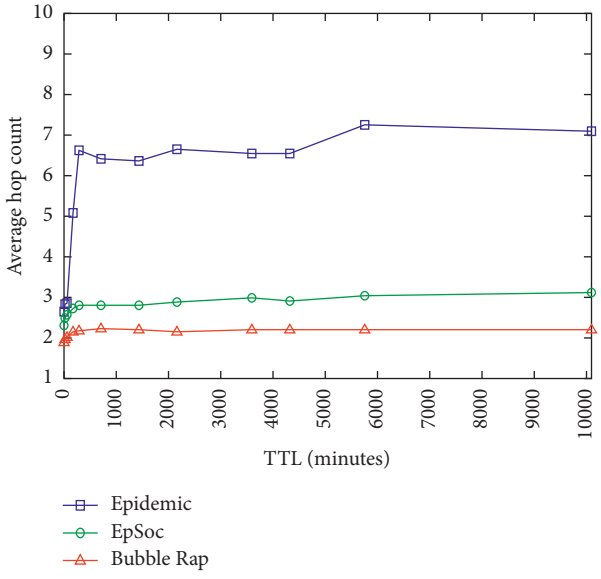


Figure 9: Average hop count versus TTL.

values (12 h–1 w). This is because our algorithm always enables the active node to carry messages with lower TTL.

From Figure 9, we observed that if TTL is very low (10 m–1 h), all routing protocols in the experiment have low average hop count. This is due to the low number of forwardings between nodes as message life exhausted quickly. Bubble Rap and EpSoc have almost stable performance when TTL is increased. Exploited social features in Bubble Rap and EpSoc and applying the blocking mechanism in EpSoc decrease the effect of changing the value of TTL on the number of traversed nodes to the destination. On the contrary, when the TTL is increased, the average hop count of Epidemic increases significantly because of the flooding-based forwarding strategy. For EpSoc, the average hop count is decreased significantly compared to Epidemic because of the social effect of active nodes.

5. Conclusion

In this paper, we investigate the flooding-based forwarding strategy, that is, Epidemic with social features to improve the routing performance in the opportunistic mobile social network (OMSN). Inspired by the advantages of Epidemic routing protocols in terms of delivery ratio and delivery delay and by exploiting the social activity of nodes, we formulated a flooding-based social-based routing protocol named as EpSoc. Simulation experiments using real data sets are conducted to evaluate our protocol performance. From the presented results, our approach increases the delivery ratio and decreases the delivery overhead, average latency, and average hop count as compared to the Epidemic protocol. As for benchmark social-based routing protocol performance (Bubble Rap), our protocol decreases the average latency significantly with a better delivery ratio in high buffer size and low TTL scenarios. Generally, we manage to exploit the advantage of Epidemic and Bubble Rap to improve the data dissemination in the OMSN.

Additional Points

Significance. Social features of a node can be utilized to have an effective role in the routing protocol in the OMSN network. This paper presents the routing protocol that exploits degree centrality to increase the delivery ratio and decrease the overhead and latency. Moreover, exploiting other social features such as similarity and community may lead to being more efficient routing protocol. Therefore, combining social features with other forwarding techniques such as the Epidemic-based strategy is significant to have an efficient forwarding strategy in the OMSN to support green technologies.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot, "The diameter of opportunistic mobile networks," in *Proceedings of the 2007 ACM CoNEXT Conference*, p. 12, ACM, New York, NY, USA, December 2007.

[2] M. Conti, S. Giordano, M. May, and A. Passarella, "From opportunistic networks to opportunistic computing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 126–139, 2010.

[3] K. Zhu, W. Li, X. Fu, and L. Zhang, "Data routing strategies in opportunistic mobile social networks: taxonomy and open challenges," *Computer Networks*, vol. 93, pp. 183–198, 2015.

[4] Y. Liu, Z. Yang, T. Ning, and H. Wu, "Efficient quality-of-service (QoS) support in mobile opportunistic networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4574–4584, 2014.

[5] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc

networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.

[6] T. Le and M. Gerla, "Social-distance based anycast routing in delay tolerant networks," in *Proceedings of the 2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Vilanova i la Geltru, Spain, June 2016.

[7] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, NY, USA, August 2003.

[8] M. Alajeely, R. Doss, and A. Ahmad, "Routing protocols in opportunistic networks–a survey," *IETE Technical Review*, pp. 1–19, 2017.

[9] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep. CS-200006, Duke University, Durham, NC, USA, 2000.

[10] A. Lindgren, A. Doria, and O. Schelen, *Probabilistic Routing in Intermittently Connected Networks*, Springer, Berlin, Germany, 2004.

[11] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[12] P. Pholpabu and L. L. Yang, "Routing protocols for mobile social networks achieving trade-off among energy consumption, delivery ratio and delay," in *Proceedings of the 2015 IEEE/CIC International Conference on Communications in China (ICCC)*, Shenzhen, China, November 2015.

[13] M. Nikoo, F. Ramezani, M. Hadzima-Nyarko, E. K. Nyarko, and M. Nikoo, "Flood-routing modeling with neural network optimized by social-based algorithm," *Natural Hazards*, vol. 82, no. 1, pp. 1–24, 2016.

[14] E. Bulut, Z. Wang, and B. K. Szymanski, "Cost-effective multiperiod spraying for routing in delay-tolerant networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 5, pp. 1530–1543, 2010.

[15] T. Matsuda and T. Takine, "(p,q)-epidemic routing for sparsely populated mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 783–793, 2008.

[16] P. Mundur, M. Seligman, and J. N. Lee, "Immunity-based epidemic routing in intermittent networks," in *Proceedings of the 2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, San Francisco, CA, USA, June 2008.

[17] P. Y. Chen, S. M. Cheng, and K. C. Chen, "Optimal control of epidemic information dissemination over networks," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2316–2328, 2014.

[18] C. Y. Aung, I. W. H. Ho, and P. H. J. Chong, "Store-carry-cooperative forward routing with information epidemics control for data delivery in opportunistic networks," *IEEE Access*, vol. 5, pp. 6608–6625, 2017.

[19] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: positive and negative social effects," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 387–401, 2013.

[20] C. C. Sobin, V. Raychoudhury, G. Marfia, and A. Singla, "A survey of routing and data dissemination in delay tolerant networks," *Journal of Network and Computer Applications*, vol. 67, pp. 128–146, 2016.

[21] P. Pholpabu and L.-L. Yang, "Social contact probability assisted routing protocol for mobile social networks," in *Proceedings of the IEEE 79th Vehicular Technology Conference (VTC Spring)*, Seoul, South Korea, May 2014.

[22] C. Boldrini, M. Conti, and A. Passarella, "Exploiting users' social relations to forward data in opportunistic networks: the HiBOp solution," *Pervasive and Mobile Computing*, vol. 4, no. 5, pp. 633–657, 2008.

[23] H. Nguyen and S. Giordano, "Context information prediction for social-based routing in opportunistic networks," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1557–1569, 2012.

[24] Z. Li, C. Wang, S. Yang, C. Jiang, and X. Li, "LASS: local-activity and social-similarity based data forwarding in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, pp. 174–184, 2015.

[25] A. Socievole, E. Yoneki, F. De Rango, and J. Crowcroft, "ML-SOR: message routing using multi-layer social networks in opportunistic communications," *Computer Networks*, vol. 81, pp. 201–219, 2015.

[26] P. Pholpabu and L.-L. Yang, "A mutual-community-aware routing protocol for mobile social networks," in *Proceedings of the Global Communications Conference (GLOBECOM)*, Austin, TX, USA, December 2014.

[27] R. Ciobanu, C. Dobre, V. Cristea, F. Pop, and F. Xhafa, "SPRINT-SELF: social-based routing and selfish node detection in opportunistic networks," *Mobile Information Systems*, vol. 2015, Article ID 596204, 12 pages, 2015.

[28] Y. Liu, K. Wang, H. Guo, Q. Lu, and Y. Sun, "Social-aware computing based congestion control in delay tolerant networks," *Mobile Networks and Applications*, vol. 22, no. 2, pp. 1–12, 2016.

[29] J. Guan, Q. Chu, and I. You, "The social relationship based adaptive multi-spray-and-wait routing algorithm for disruption tolerant network," *Mobile Information Systems*, vol. 2017, Article ID 1819495, 13 pages, 2017.

[30] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *IEEE Transactions on Computers*, vol. 63, pp. 1682–1695, 2014.

[31] A. C. K. Vendramin, A. Munaretto, M. R. Delgado, M. Fonseca, and A. C. Viana, "A social-aware routing protocol for opportunistic networks," *Expert Systems with Applications*, vol. 54, pp. 351–363, 2016.

[32] Y. Zhu, C. Zhang, X. Mao, and Y. Wang, "Social based throwbox placement schemes for large-scale mobile social delay tolerant networks," *Computer Communications*, vol. 65, pp. 10–26, 2015.

[33] J. Miao, O. Hasan, S. B. Mokhtar, L. Brunie, and G. Gianini, "A delay and cost balancing protocol for message routing in mobile delay tolerant networks," *Ad Hoc Networks*, vol. 25, pp. 430–443, 2015.

[34] H. Chen and W. Lou, "Contact expectation based routing for delay tolerant networks," *Ad Hoc Networks*, vol. 36, pp. 244–257, 2016.

[35] F. Xia, L. Liu, J. Li, J. Ma, and A. V. Vasilakos, "Socially aware networking: a survey," *IEEE Systems Journal*, vol. 9, pp. 904–921, 2015.

[36] W. Moreira, P. Mendes, and S. Sargento, "Opportunistic routing based on daily routines," in *Proceedings of the World of wireless, mobile and multimedia networks (WoWMoM)*, San Francisco, CA, USA, June 2012.

[37] W. Moreira, P. Mendes, and S. Sargento, "Social-aware opportunistic routing protocol based on user's interactions and interests," in *Proceedings of the International Conference on Ad Hoc Networks*, vol. 129, pp. 100–115, Springer International Publishing, Barcelona, Spain, August 2014.

[38] Website of CRAWDAD project, http://crawdad.org/cambridge/haggle/20090529/.

[39] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Brussels, Belgium, March 2009.

Hindawi
Mobile Information Systems
Volume 2018, Article ID 7689549, 9 pages
https://doi.org/10.1155/2018/7689549

Research Article

Reducing Smartwatch Users' Distraction with Convolutional Neural Network

Jemin Lee,[1] **Jinse Kwon**,[2] **and Hyungshin Kim**[2]

[1]*Industrial Engineering and Management Research Institute, KAIST, Daejeon, Republic of Korea*
[2]*Department of Computer Science and Engineering, Chungnam National University, Daejeon, Republic of Korea*

Correspondence should be addressed to Hyungshin Kim; hyungshin@cnu.ac.kr

Received 1 June 2017; Revised 24 December 2017; Accepted 29 January 2018; Published 15 March 2018

Academic Editor: Federica Cena

Smartwatches provide a useful feature whereby users can be directly aware of incoming notifications by vibration. However, such prompt awareness causes high distractions to users. To remedy the distraction problem, we propose an intelligent notification management for smartwatch users. The goal of our management system is not only to reduce the annoying notifications but also to provide the important notifications that users will swiftly react to. To analyze how to respond to the notifications daily, we have collected 20,353 in-the-wild notifications. Subsequently, we trained the convolutional neural network models to classify important notifications according to the users' contexts. Finally, the proposed management allows important notifications to be forwarded to a smartwatch. As experiment results show, the proposed method can reduce the number of unwanted notifications on smartwatches by up to 81%.

1. Introduction

Smartwatches have become one of the first popular wearables, with the launch of high-quality flagship products by major global companies. According to recent surveys [1–3], one of the key features of smartwatches is user notification about various events, such as new messages or software updates. Several researchers have already investigated interruptions caused by notifications [4–6]. Based on the research results, mobile notifications at an inopportune moment at which users concentrate on their tasks lead to disruptive effects.

In the current notification delivery system, all notifications can be shown on all connected mobile devices simultaneously. When a user is wearing a smartwatch, they are severely distracted by notifications delivered from a smartphone because the smartwatch is a wrist-worn device [2]. To block unwanted notifications, users would manually change the settings from time to time. The burden of management causes the abandonment of smartwatches [2]. Therefore, the notification delivery system should notify only important notifications that need to be swiftly reacted considering the sender, topic, or location.

To lessen the distractive effects, many researchers have proposed several approaches that precisely predict the opportune moment to send notifications to users with machine learning techniques [7–9]. However, attention management is still lacking in emerging situations where users carry multiple mobile devices. Most previous research on notifications and interruptions only focused on a single device (e.g., a smartphone).

In this paper, we present an intelligent notification delivery system for a smartphone and a smartwatch. We extend our pilot work that filtered unwanted notifications using deep neural networks (DNNs). Unlike the pilot work, we not only collect more data from more users but also widely compare five machine learning algorithms with individual data and generic data. To train a model, we collect real users' responses and context data. Based on the users' responses and the previous work's criteria [8], we unobtrusively label notifications for a ground-truth value. The previous work [8] has found that users handle the notification within a certain time and launch related applications only if the arrival time is an opportune moment. We bring this assumption into our work to programmatically label important notifications without

any questionnaire. To infer the notification labels, we combine the user's response time (time difference between arriving and removing) and app launch (indicating whether an application is launched).

To train the convolutional neural network (CNN) models, we collected 20,352 *in-the-wild* notifications from 13 users for approximately 5 weeks. We performed an analysis on the collected data to extract features. Subsequently, we transformed the collected data to an $\mathbb{R}^{2\times5}$ matrix according to the correlation of features. To obtain better models, we compared the CNN (sparse connection) to the DNN (dense connection). As a result, the CNN outperforms other predictions models with a slightly higher *F*-score value (mean 88%) due to the correlation of features and the transfer learning effect. In addition, we compare the *F*-score of the prediction models trained on individual data with a generic prediction model trained on all users' data. Overall, the personal models are slightly better than the generic models. For predicting important notifications, the transfer learning-based CNN model achieved 91% of the precision on average. Accordingly, our model filtered unimportant notifications up to 81%. The effect of our delivery system depends on the ratio of important notifications and the model accuracy.

Our contributions are summarized as follows:

(i) With our mobile application, we collected 20,352 notifications and context data from 13 users for approximately 5 weeks. While gathering data, our tool programmatically infers users' interactions to decide the notification label.

(ii) We extracted 10 features from the collected data and analyzed their correlations to transform them into images. With the transformed data, we trained the convolutional neural network models corresponding to each user on a server equipped with high-performance GPU.

(iii) Based on the quantitative analysis, the impact of the proposed method was validated. Our results show that the trained models filtered unimportant notifications up to 81%. In addition, we reveal the impact of multidevices, which caused the inaccurate predictions by users.

2. Related Work

The human computer interaction research groups have studied various techniques to precisely infer users' interruptibility. In the desktop computer environment, they proposed the interruptibility management (IM) system for multiple applications [5, 10, 11]. For more accurate systems, context-aware interruptibility systems were proposed [12, 13], which require a user to wear sensors for extracting context. These approaches are based on sensors attached on the human body that can trigger notifications in an opportune moment by precisely recognizing the context.

In recent studies, researchers have exploited smartphones that are equipped with a variety of sensors to build the IM system [7, 8, 14–19]. For a more advanced system, nonobstructive approaches have been presented [7, 16]. To build the interruptibility models, these approaches unobtrusively monitored variation of context and system configurations without the user's involvement and questionnaires. Moreover, the notification's contents were considered as a context to build a better model [8, 15]. Turner et al. [19] proposed a decision-on-information-gain model to understand the users' microdecisions against notifications. Attelia [18] automatically mined important usage information to predict breakpoints for interruptions.

In addition, the OS-level IM system was designed in terms of privacy protection and deep context extraction [17]. For wearable devices, Kern and Schiele [20] proposed a delivery mechanism that relays notifications corresponding to six contexts that they defined.

However, all prior works have focused on predicting an opportune moment in a single mobile device. Those works have not yet considered emerging situation in which many people carry multiple mobile devices daily. Unlike previous works, we have focused on reducing notification delivery to a smartwatch from a smartphone to reduce user distraction.

3. Dataset

We focus on users who use a smartphone and a smartwatch simultaneously. Typically, not all users wear a smartwatch on the wrist. To collect data, we hand out the LG-Urbane W150 to 13 participants who are willing to join our experiments even without any monetary incentive. Table 1 lists the participant demographics. These participants consist of 10 males and 3 females with the age span between 24 and 35 years. As shown in Figure 1, the data were gathered during approximately 5 weeks on average. Across approximately 5 weeks, we finally collected 20,352 notifications.

4. Data Collection

In this section, we briefly describe how the notification type is unobtrusively labeled and what the type of sensor data that are collected.

4.1. Implementation. To collect notifications, we implemented an Android application that runs on a smartphone as a background service to programmatically decide notification labels as well as monitor the contexts when a notification is received. For deciding important notifications, our application was developed a few APIs, for example, Notification Listener Service (https://developer.android.com/reference/android/service/notification/NotificationListenerService.html) and UsageStatsManager (https://developer.android.com/reference/android/app/usage/UsageStatsManager.html) that are supported in API level 21 (Android 5.0). With Notification Listener Service, we can identify the arrival and removal times of the notifications. With UsageStatsManager, we can observe the states of the mobile application usage. By combining the two APIs, our mobile application can automatically decide important notifications. In addition, our mobile application exploits a third party library for computational social science [21], as well as SensorManager and SensorDataManager to obtain the contexts and store a large amount of data, respectively.

Table 1: Participant demographics.

Number of users	13 (10 males and 3 females)
Age	24, 25 (4), 26, 27 (2), 29, 30 (2), 34, and 35
Occupation	Students (9), office workers (3), and others (1)
Smartphone	Galaxy-s6 (3), Galaxy-s4 (2), Galaxy-note 3, Galaxy-note (2), Galaxy-a8, Galaxy-a7, Galaxy-Grand Max, Nexus 5, and Nexus 5x
Smartwatch	LG-Urbane W150 (13)

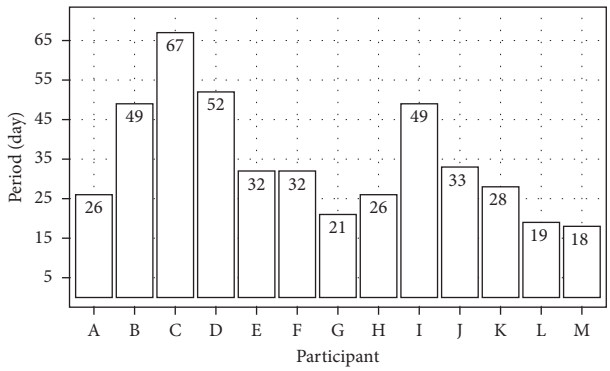


Figure 1: Data collection period across 13 participants.

In addition, it uses the Android OS API, Activity Recognition (https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi), for monitoring a user's activity. Table 2 lists the 10 contexts collected by our monitoring application.

4.2. Automated Labeling in Incoming Notifications. To infer the interactions and responses without any questionnaires, we combine a user's response time (indicating how quick a user responds) and the application launch (application relevant to the notification). We begin by cross-referencing the arrived notifications with a user's response time and the application launch that triggers the notifications. We consider a user's response time as a key factor to infer a user's interruption level because the delay to respond to notifications is highly relevant to important level of notifications [2, 8, 22, 23].

Figure 2 shows the high-level example of label notification when a user is interacting with a smartphone, while wearing a smartwatch. For a delay time threshold, we apply 10 minutes to each notification. The threshold was determined based on a previous work [8], which showed that approximately 60% of the interactions with notifications occur within 10 minutes. To prevent the missing of important notifications, our assumption for labeling is conservative. According to a previous work [24], while not all kinds of notifications are important, many were clicked within 30 seconds. Consequently, the threshold of 10 minutes implies enough margin to avoid missing them.

Figure 3 illustrates the results for labeling. Figure 3(a) shows the distribution of each label condition. Figure 3(b) shows the final labeling by cross-referencing. Finally, the result shows that the users represent various proportions of important notifications. According to the automated labeling based on combining a user's response time and application launch, the important notification rate of each user ranges from 35% to 90%, with the average being 68%. From the labeling results, we observed that user K is distracted by most of the notifications because user K has the lowest important notification rate.

Table 2: Feature group from the collected sensor data.

Group	Features
Notification's contents	Sender application name, Priority, and Title
Physical activities	Classifying activities into six classes: InVehicle, OnBicycle, OnFoot, Running, Still, Tilting, Walking, and Unknown
Time	Time of day, Day of the week, and Glance time
Sensor data	Recent phone usage, Phone's screen status, and Proximity

5. Building Prediction Models

In this section, we briefly describe how we handled the data for training the machine learning models, model structures, and how we trained them.

5.1. Preprocessing. To train the models, the categorical data we collected should be transformed into numerical data. Simply, we changed the nominal data (category name) to a unique digit number. As a result, each feature represents quite a different numerical range. For example, the title feature ranges from 1 to 152 in a user. However, the proximity feature has a binary number of either 0 or 1. Different scales of the features make the training difficult and slow to be converged. To remedy this issue, we normalized the data ranging from 0 to 1 with the following equation:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}, \tag{1}$$

where $x = (x_1, \ldots, x_n)$ and z_i is ith normalized data. To train the DNN, an input data shape of an $\mathbb{R}^{10\times1}$ matrix is used. A correlation of the input data is not important because the DNN is densely connected. However, in a case where we trained the CNN, a correlation of input data should be considered due to sparse connection and weight sharing. To consider the correlations among features, we computed the following Pearson correlation:

$$\operatorname{corr}(x, y) = \frac{\operatorname{cov}(x, y)}{\operatorname{sd}(x) \cdot \operatorname{sd}(y)}. \tag{2}$$

Figure 4 represents the feature correlations of user J. It results in an $\mathbb{R}^{2\times5}$ matrix. Based on the correlation result, the input data of user J is the following matrix:

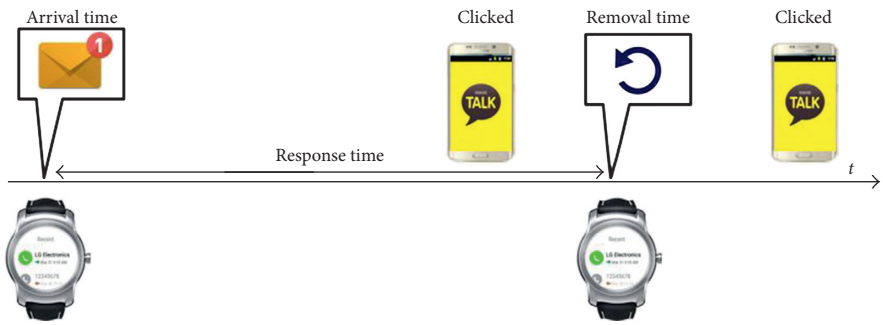


Figure 2: Labeling a notification on how quickly it responded and whether an application that triggers it is launched.

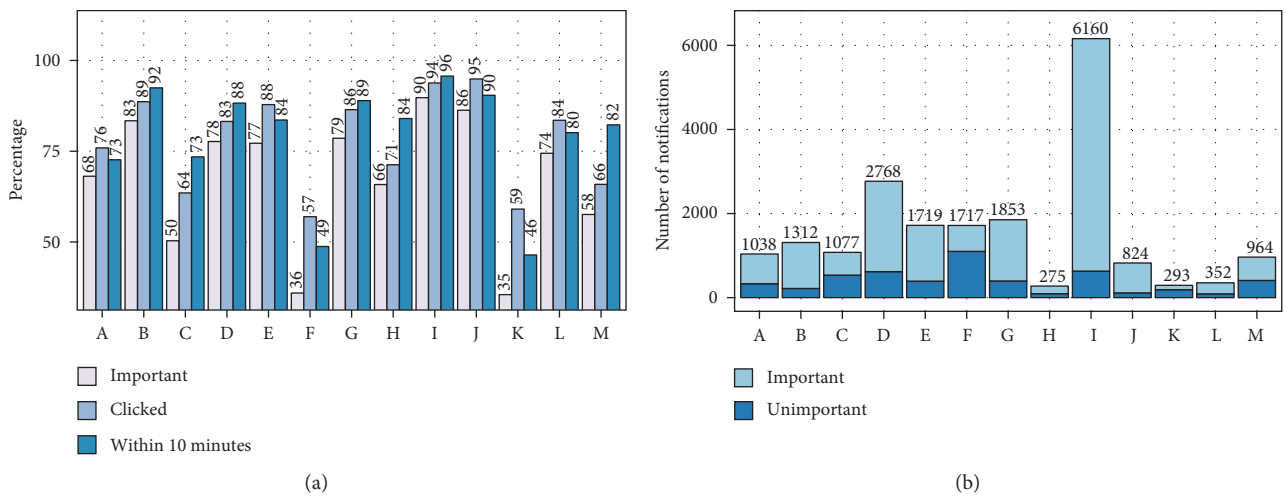


Figure 3: 20,352 notifications among 13 users for approximately 5 weeks; the bar plots of (a) the distribution of label types and (b) the distribution of important and unimportant notifications.

$$\begin{bmatrix} F_{\text{App name}} & F_{\text{Title}} & \cdots & F_{\text{Recent phone usage}} \\ F_{\text{Proximity}} & F_{\text{Priority}} & \cdots & F_{\text{Seen time}} \end{bmatrix}. \quad (3)$$

Likewise, we applied data transformation to the other users. As space is limited here, we have omitted all users' correlation data. For data preprocessing, we used the caret package (http://caret.r-forge.r-project.org/) in *R*.

5.2. Machine Learning Models. We used the following machine learning algorithms to predict important notifications: (1) naive Bayes (NB), (2) support vector machine (SVM), (3) random forest (RF), (4) deep neural networks (DNNs), and (5) convolutional neural networks (CNNs). To apply data into the naive Bayes algorithm, we used the categorical data type. In the support vector machine, we selected the nonlinear kernel function which is the radial basis function (RBF). In this case, the RBF kernel-based SVM was much better than the linear kernel-based SVM. To find the opportune radial kernel, we adjusted the C and sigma parameters with autotuning methods in the caret package. The ranges of the optimal C and sigma options for each user are 0.25–5 and 0.08–0.11, respectively. Likewise, in the random forest, we also explored a variety of hyperparameters. The best options were mtry = 45 and ntrees = 500 across all users.

A variety of deep learning models have been proposed for diverse applications and sensor types. Examples include the CNN, widely used for image classification [25, 26] and recently for text classification [27], and the DNN, used for speech recognition [28]. In our pilot work [29], we assumed that an important notification depends on the notification's contents and the user's context. Subsequently, we trained the fully connected 11-layer feedforward neural network consisting of 9 hidden layers. We bring this assumption and the mode into the current work by extending the input size to 10. Figure 5 illustrates the slightly extended DNN structure. In addition to the DNN model, we trained CNN model with same data to directly compare with the DNN model. Unlike the DNN, the CNN handles data correlation according to the kernel size. Specifically, the CNN generates useful features via its learnable filters. As shown in Figure 6, we implemented the CNN with two convolutional layers, followed by a pooling layer, a fully connected layer, and a logistic regression layer (sigmoid).

The convolution operation sums the contributions from different dimensional data in the input layer as follows:

$$y_l = \sum_{k=1}^{K} h_k * w_{kl}, \quad (4)$$

where y_m is the mth plane of the output data from each convolutional layer, h_k is the kth plane of the input data that

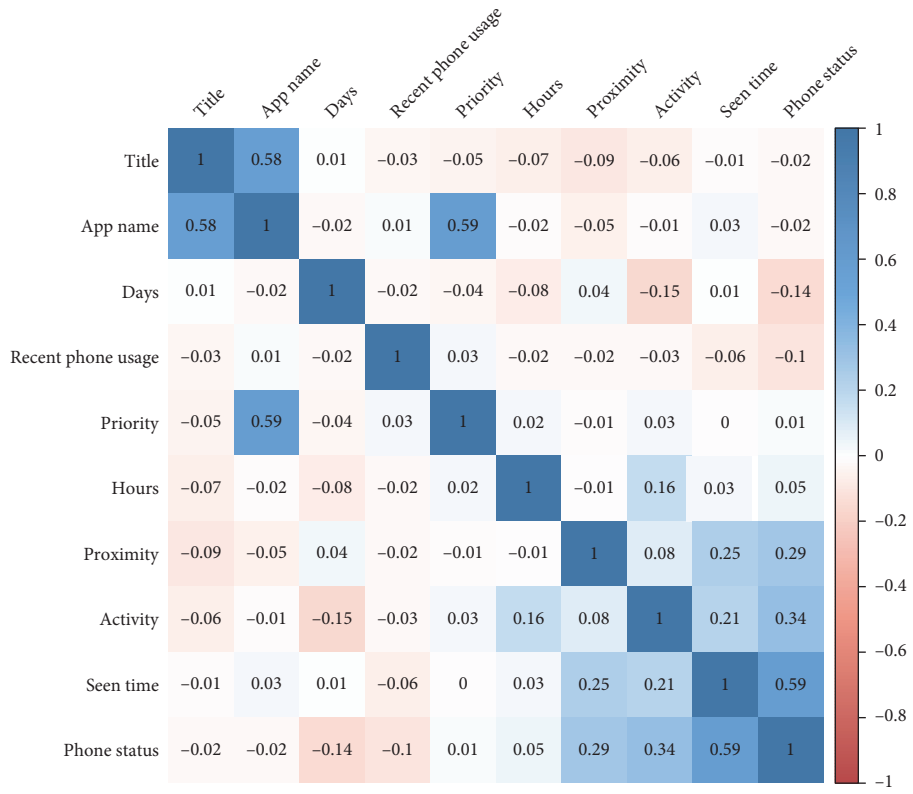


Figure 4: Pearson correlations of features in user J.

has K planes in total, and w_{km} is the kth plane of kernel m. We used a two-dimensional input layer. Therefore, K represents a single channel. After the input layer, K is the size of the activation map. The detailed convolutional operations are as follows:

$$z_{i,j,k} = b_k + \sum_{u=1}^{f_h} \sum_{v=1}^{f_w} \sum_{k'=1}^{f_{n'}} x_{i',j',k'} \cdot w_{u,v,k',k}, \tag{5}$$

with

$$\begin{aligned} i' &= u \cdot s_h + f_h - 1, \\ j' &= v \cdot s_w + f_w - 1, \end{aligned} \tag{6}$$

where $z_{j',j',k'}$ is the output of the neuron located in row i' and column j' in feature map k' of the convolutional layer (layer l). s_h and s_w are the vertical and horizontal strides, respectively; f_h and f_w are the height and width of the receptive field (i.e., filter size), respectively; $f_{n'}$ is the number of feature maps in the previous layer (layer $l-1$). For the vertical and horizontal strides, we used one value. To maintain the same height and width for all layers, we used *zero padding*. $x_{i',j',k'}$ is the output of the neuron located in layer $l-1$, row i', column j', and feature map k' (or channel k' if the previous layer is the input layer). b_k is the bias term for feature map k (in layer l). $w_{u,v,k',k}$ is the connection weight between any neuron in feature map k of layer l and its input located at row u, column v (relative to the neuron's receptive field), and feature map k'.

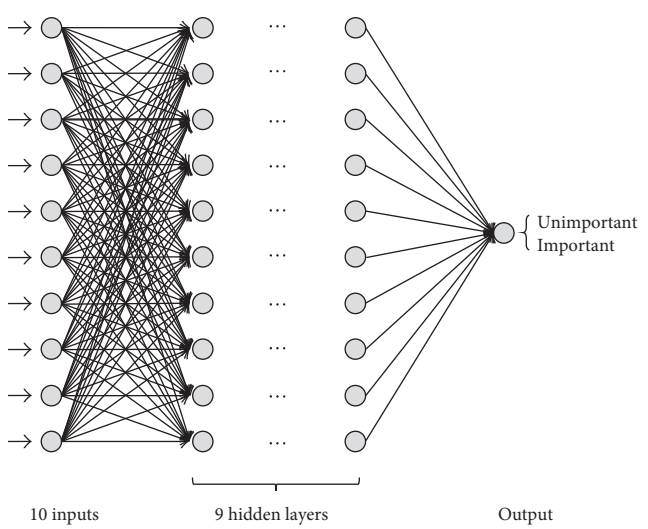


Figure 5: Fully connected feedforward neural network structure.

As an activation function, we used a rectified linear unit (ReLU) because this unit not only helps models to converge but also to avoid the vanishing gradient problem in models. $z_{j',j',k'}$ is applied to the ReLU as

$$h_{i',j',k'} = \max\left(0, z_{i',j',k'}\right). \tag{7}$$

In the last layer, the CNN represents a logistic regression to convert continuous data into notification labels. The last of the layer is computed as

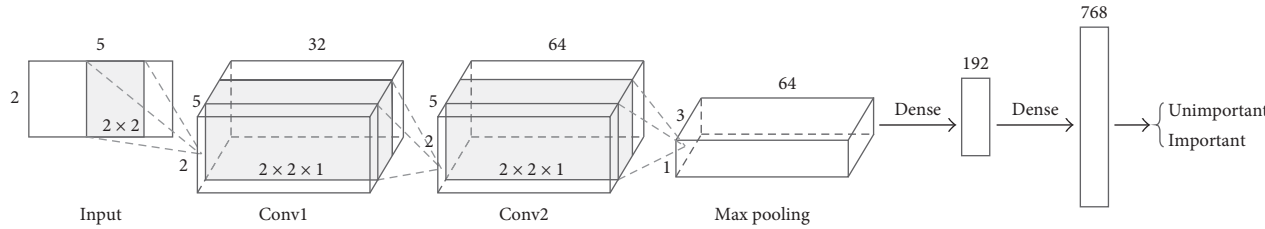


Figure 6: The convolutional neural network structure with 2 convolutional layers and 1 fully (dense) connected layer.

$$\hat{y} = \frac{1}{1 + \exp(-w^T x)}. \quad (8)$$

To compute the training error, we used the following cross entropy:

$$\text{Cost} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log \hat{y} + \left(1 - y^{(i)}\right)\log(1 - \hat{y})\right]. \quad (9)$$

Subsequently, we trained the neurons with the *back-propagation* algorithm that propagates the cost to whole layers.

5.3. Model Training. To train diverse models, we used two frameworks: caret (*R* environment) and TensorFlow (Python environment). The caret package (http://topepo.github.io/caret/index.html) supports many traditional machine learning algorithms. Therefore, we exploited this package for the model training of NB, SVM, and RF.

We trained the DNN and CNN on Google's TensorFlow (https://www.tensorflow.org/) with the training data, by dividing the collected data into two parts, 70% for training and 30% for testing.

To successfully train the DNN with small individual data, we used the whole dataset from all the users. Subsequently, we reused the lower layers of the trained networks: this is called transfer learning. TensorFlow provides the tf.stop_gradient() function to freeze particular layers for fine tuning. Therefore, with personal data, it is easily possible to fine tune some higher-level portions of the network. We explored how many layer needs to be retrained. As a result, we found no significant difference among the layers. To reduce overfitting and computational problems, we decided to retrain only the final layer and completed building the neural network models.

To avoid overfitting concerns, we took two steps: first, we applied the *dropout* in *conv1* and *conv2* layers and the fully connected layer. Dropout is widely used in CNNs to avoid overfitting during model training [30].

In our case, we set up 10% rate as the dropout option. Second, we implemented an early stop mechanism, which forces training to be terminated when the validation error starts to increase. We initialized the weight vectors with the *Xavier* method [31] and the learning rate at 10^{-5}. In addition, we chose the Adam optimizer to train the weight vectors [32]. We performed more than 20,000 epochs. However, we stopped the training process early when the difference between the validation and training costs was larger than 10^{-2}. In addition, we performed full-batch learning, which means that the entire training data were used for a single gradient descent. We implemented our two models on a local server equipped with NVIDIA GTX 1080 (8.8 TFLOPS). After the whole training process, we finally built the deep learning models to infer whether a notification is relayed to a smartwatch with the sensed contexts.

6. Evaluation

In this section, we elaborate how accurate our classifier is in predicting important notifications. In addition, we compare the accuracy of the prediction models trained on user's personal data with a generic prediction model trained on all users' data.

6.1. Evaluating Prediction Models. We evaluated five prediction models by comparing the prediction results with the ground-truth labels that are unobtrusively determined on the testing data, by using the testing data, which accounts for 30% of the collected data. Figure 7 shows the precision, recall, and *F*-score of all models.

Our results demonstrate no significant difference in performance among the five machine learning algorithms. However, the *F*-score of fine tuning the CNN (mean 88%) is slightly better than others. We expect that performance gap among other models is increased by more users' data. As a recall, the CNN (mean 91%) outperformed the others.

The low precision results in the unimportant notification deliveries to a smartwatch. More severely, the low recall leads to negative effects that reduce information awareness. Therefore, to reduce the negative effects, we focus on maximizing the recall by sacrificing the precision.

In spite of the 91% recall on average, a few predictors show relatively poor results. However, we claim that missing an important notification on a smartwatch is not a critical issue because the smartwatch is a secondary device. Therefore, based on previous works [33], a smartphone is the key device that checks for notifications when users carry four devices including a tablet, a PC, a smartwatch, and a smartphone. Basically, users can check all the notifications on their smartphones even if they are not relayed to the smartwatches. To show the effectiveness, we calculated the number of notifications that were filtered out. To calculate them, we simply subtracted the notifications that were classified as important from all notifications. Figure 8 shows the filtered notification rate for each user. The filtered notification rate is 36% on average. Obviously, the filtering effects relied on the number of unimportant notifications and the recall of models.

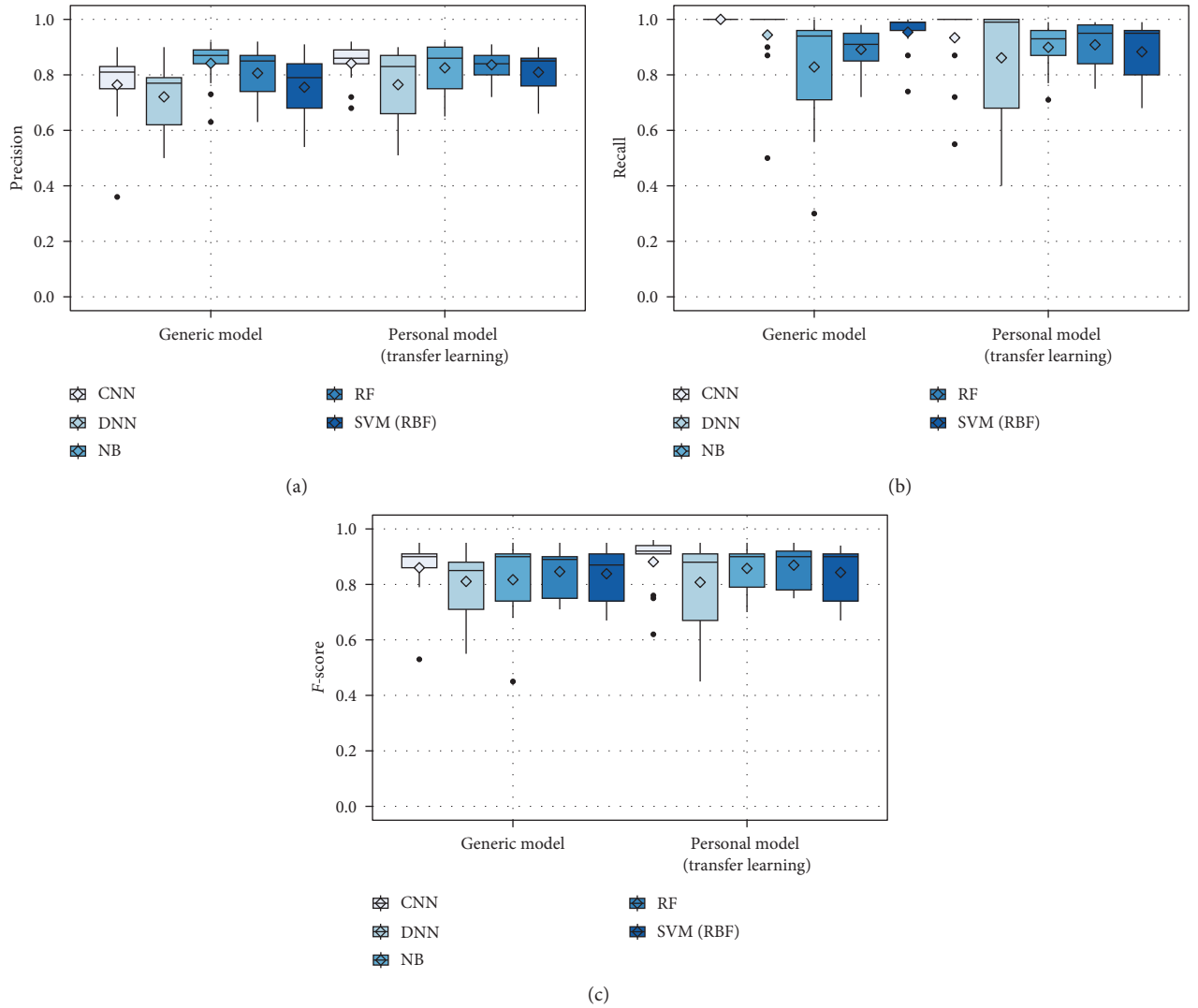


Figure 7: Prediction results of the CNN, DNN, NB, RF, and SVM models: (a) precision, (b) recall, and (c) *F*-score.

6.2. Generic versus Personal Models. We compare the performance of the predictions models trained on individual data with a generic model trained on all users' data. In most cases, the personal model is better than the generic model. However, in some cases, when the amount of data is small, the generic model shows a better *F*-score than the personal model. This is primarily due to the lack of training data. Unlike other machine learning algorithms, the personal model of the CNN stably outperformed its generic model across all users. As mentioned before, we performed fine tuning with the generic model to build the personal models: this is called transfer learning. Its benefits are even greater when we consider users with little or no training data.

7. Discussion and Limitations

We are aware that notification filtering may affect the user experience of smartwatches. However, we claim that the user experience degradation of the smartwatch is not critical for the following reasons: first, users tend to addictively check their smartphones [34]. In addition, based on ANOVA, recent works [9] have revealed that the alert type of the smartphone (silence and vibration) does not show statistically significant effects on the notification awareness. Specifically, users reported that they just missed the notifications for 14.63% of the time, even when their smartphones were in silent mode. This means that the notification awareness of users is not dependent on any other factors such as the alert type of the smartphone or the presence of the smartwatch due to frequent smartphone usage. Next, users are not dependent on the smartwatches because smartwatches are secondary devices. A recent work [3] investigated the level of discomfort if devices were running out of battery. According to the questionnaire survey, 33% of the smartwatch users reported *neutral*. However, 46% of the users responded with *very uncomfortable*. In addition, Weber et al. found that users prefer a smartphone for receiving notifications among the four devices (a tablet, a smartwatch, a smartphone, and a PC) regardless of the notification type [33]. This is because the smartphone is a key device for online connectivity and communication with other people.

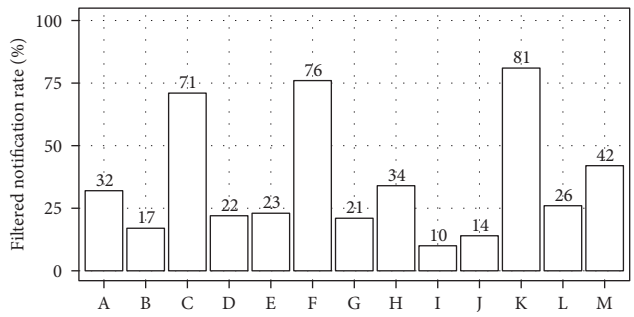


Figure 8: Filtered notification rate.

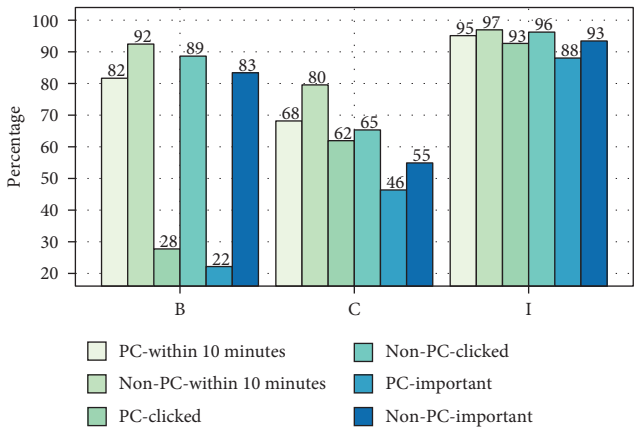


Figure 9: The labeling effects using PCs.

In a few users, our classifier shows poor prediction results. The reasons are as follows: first, some users just glanced at the wrist and their smartphones to decide whether to interrupt their current activities to deal with the notification. In this case, the proposed labeling method misclassifies this notification as dismissed because there is no interaction. Next, some users read the notifications and dismissed them on their PCs. In this case, our labeling method does not capture the interactions. To investigate the effects by using PCs, we conducted one additional experiment. The experiment was divided into two conditions: allowing three users to use PCs and not to use PCs. The experiments were conducted for a week, respectively.

Figure 9 shows the three class types under two conditions. If a user uses a PC, many notifications are misclassified as unimportant because the application launch occurs in the PC. Therefore, we forced them not to use any device except a smartphone and a smartwatch for checking notification. Despite our instructions, a few users still used the PCs for their official tasks such as sending documents. Therefore, a poorly trained model stems from misclassified notification labels.

8. Conclusion

Even though smartwatches improve the awareness of incoming notifications, they aggravate the disruptive nature of notification delivery because they are worn on the human body. To reduce a smartwatch user's distraction, we proposed a notification delivery system that relays only important notifications predicted by CNN models. To build our models, we collected 20,352 notifications and sensor data from three users using a mobile application, which unobtrusively monitors all the data. Subsequently, we implemented a binary classifier that identifies important notifications. For important notification prediction, our classifier attained 76% and 91%, respectively, for the precision and recall on average, spanned across all users. This classifier can reduce the distraction in smartwatch users without noticeable degradation in the users' awareness.

In the future, we plan to deploy our notification delivery system to real users. In addition, we implemented the OS-level software without cloud assistance not only to capture the detailed context (i.e., full text) but also to avoid privacy intrusion.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Research Foundation (NRF) of Korea funded by the Ministry of Education (NRF-2017R1D1A1B03034705)

References

[1] A. Visuri, Z. Sarsenbayeva, N. van Berkel et al., "Quantifying sources and types of smartwatch usage sessions," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*, pp. 3569–3581, ACM, New York, NY, USA, 2017.

[2] M. E. Cecchinato, A. L. Cox, and J. Bird, "Always on(line)?: user experience of smartwatches and their role within multi-device ecologies," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*, pp. 3557–3568, ACM, Denver, CO, USA, May 2017.

[3] C. Min, S. Kang, C. Yoo et al., "Exploring current practices for battery use and management of smartwatches," in *Proceedings of the IEEE International Symposium on Wearable Computers (ISWC)*, pp. 11–18, Osaka, Japan, 2015.

[4] S. T. Iqbal and E. Horvitz, "Notifications and awareness: a field study of alert usage and preferences," in *Proceedings of International Conference on Computer Supported Cooperative Work*, pp. 27–30, ACM, Shanghai, China, April 2010.

[5] E. Horvitz, P. Koch, and J. Apacible, "Busybody: creating and fielding personalized models of the cost of interruption," in *Proceedings of the ACM International Conference on Computer Supported Cooperative Work (CSCW)*, pp. 507–510, Chicago, Illinois, USA, November 2004.

[6] L. Leiva, M. Böhmer, S. Gehring, and A. Krüger, "Back to the app: the costs of mobile application interruptions," in *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services, MobileHCI*, pp. 291–294, San Francsico, CA, USA, September 2012.

[7] V. Pejovic and M. Musolesi, "InterruptMe: designing intelligent prompting mechanisms for pervasive applications," in *Proceedings of the ACM International Joint Conference on Pervasive*

and Ubiquitous Computing (UbiComp), pp. 897–908, Downtown Seattle, WA, USA, September 2014.

[8] A. Mehrotra, M. Musolesi, R. Hendley, and V. Pejovic, "Designing content-driven intelligent notification mechanisms for mobile applications," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pp. 813–824, Osaka, Japan, September 2015.

[9] A. Mehrotra, V. Pejovic, J. Vermeulen, R. Hendley, and M. Musolesi, "My phone and me: understanding people's receptivity to mobile notifications," in *Proceedings of the ACM International Conference on Human Factors in Computing Systems (CHI)*, pp. 1021–1032, San Jose, CA, USA, May 2016.

[10] S. S. Intille, J. Rondoni, C. Kukla, I. Ancona, and L. Bao, "A context-aware experience sampling tool," in *Proceedings of the ACM International Conference on Human Factors in Computing Systems (CHI)*, pp. 972-973, Ft. Lauderdale, Florida, USA, April 2003.

[11] S. T. Iqbal and B. P. Bailey, "Oasis: a framework for linking notification delivery to the perceptual structure of goal-directed tasks," *ACM Transactions on Computer-Human Interaction*, vol. 17, no. 4, pp. 1–28, 2010.

[12] J. Ho and S. S. Intille, "Using context-aware computing to reduce the perceived burden of interruptions from mobile devices," in *Proceedings of the ACM International Conference on Human Factors in Computing Systems (CHI)*, pp. 909–918, Portland, OR, USA, April 2005.

[13] D. Siewiorek, A. Smailagic, J. Furukawa et al., "SenSay: a context-aware mobile phone," in *Proceedings of the IEEE International Symposium on Wearable Computers (ISWC)*, pp. 248-249, Washington, DC, USA, October 2003.

[14] S. Rosenthal, A. Dey, and M. Veloso, "Using decision-theoretic experience sampling to build personalized mobile phone interruption models," *IEEE Pervasive Computing*, vol. 6696, pp. 170–187, 2011.

[15] J. E. Fischer, N. Yee, V. Bellotti, N. Good, S. Benford, and C. Greenhalgh, "Effects of content and time of delivery on receptivity to mobile interruptions," in *Proceedings of the ACM International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pp. 103–112, Lisbon, Portugal, 2010.

[16] M. Pielot, R. D. Oliveira, H. Kwak, and N. Oliver, "Didn't you see my message?: predicting attentiveness to mobile instant messages," in *Proceedings of the ACM International Conference on Human Factors in Computing Systems (CHI)*, pp. 3319–3328, Toronto, ON, Canada, 2014.

[17] K. Lee, J. Flinn, and B. Noble, "The case for operating system management of user attention," in *Proceedings of the ACM International Workshop on Mobile Computing Systems and Applications (HotMobile)*, pp. 111–116, Santa Fe, NM, USA, February 2015.

[18] T. Okoshi, H. Nozaki, J. Nakazawa, H. Tokuda, J. Ramos, and A. K. Dey, "Towards attention-aware adaptive notification on smart phones," *Pervasive and Mobile Computing*, vol. 26, pp. 17–34, 2016.

[19] L. D. Turner, S. M. Allen, and R. M. Whitaker, "Reachable but not receptive: enhancing smartphone interruptibility prediction by modelling the extent of user engagement with notifications," *Pervasive and Mobile Computing*, vol. 40, pp. 480–494, 2017.

[20] N. Kern and B. Schiele, "Context-aware notification for wearable computing," in *Proceedings of the IEEE International Symposium on Wearable Computers (ISWC)*, pp. 223–231, Osaka, Japan, September 2015.

[21] N. Lathia, K. Rachuri, C. Mascolo, and G. Roussos, "Open source smartphone libraries for computational social science," in *Proceedings of the ACM International Conference on Pervasive and Ubiquitous Computing adjunct publication (UbiComp)*, pp. 911–920, Zurich, Switzerland, September 2013.

[22] M. E. Cecchinato, A. L. Cox, and J. Bird, "Working 9-5?: professional differences in email and boundary management practices," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*, pp. 3989–3998, Seoul, Republic of Korea, April 2015.

[23] A. S. Shirazi and N. Henze, "Assessment of notifications on smartwatches," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15)*, pp. 1111–1116, ACM, Copenhagen, Denmark, August 2015.

[24] A. S. Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt, "Large-scale assessment of mobile notifications," in *Proceedings of International Conference on Human Factors in Computing Systems (CHI)*, pp. 3055–3064, ACM, Toronto, ON, Canada, 2014.

[25] X. Zhang, J. Zou, K. He, and J. Sun, "Accelerating very deep convolutional networks for classification and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 1943–1955, 2016.

[26] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[27] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2017.

[28] K. Chen and A. Salman, "Learning speaker-specific characteristics with a deep neural architecture," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1744–1756, 2011.

[29] J. Lee, J. Kwon, and H. Kim, "Reducing distraction of smartwatch users with deep learning," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, MobileHCI '16*, pp. 948–953, ACM, New York, NY, USA, 2016.

[30] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, Society for Artificial Intelligence and Statistics, pp. 249–256, Sardinia, Italy, May 2010.

[32] D. Kingma and J. Ba, "Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980," in *Proceedings of 3rd International Conference for Learning Representations*, San Diego, CA, USA, 2015.

[33] D. Weber, A. Voit, P. Kratzer, and N. Henze, "In-situ investigation of notifications in multi-device environments," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'16)*, ACM, pp. 1259–1264, Heidelberg, Germany, 2016.

[34] O. Turel and A. Serenko, "Is mobile email addiction overlooked?," *Communications of the ACM*, vol. 53, no. 5, pp. 41–43, 2010.

Hindawi
Mobile Information Systems
Volume 2018, Article ID 2649598, 16 pages
https://doi.org/10.1155/2018/2649598

Review Article

Demystifying Authentication Concepts in Smartphones: Ways and Types to Secure Access

Sandeep Gupta, Attaullah Buriro, and Bruno Crispo

Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Correspondence should be addressed to Sandeep Gupta; sandeep.gupta@unitn.it

Received 22 August 2017; Revised 11 December 2017; Accepted 9 January 2018; Published 11 March 2018

Academic Editor: Fabio Gasparetti

Smartphones are the most popular and widespread personal devices. Apart from their conventional use, that is, calling and texting, they have also been used to perform multiple security sensitive activities, such as online banking and shopping, social networking, taking pictures, and e-mailing. On a positive side, smartphones have improved the quality of life by providing multiple services that users desire, for example, anytime-anywhere computing. However, on the other side, they also pose security and privacy threats to the users' stored data. User authentication is the first line of defense to prevent unauthorized access to the smartphone. Several authentication schemes have been proposed over the years; however, their presentation might be perplexing to the new researchers to this domain, under the shade of several buzzwords, for example, active, continuous, implicit, static, and transparent, being introduced in academic papers without comprehensive description. Moreover, most of the reported authentication solutions were evaluated mainly in terms of accuracy, overlooking a very important aspect—the usability. This paper surveys various types and ways of authentication, designed and developed primarily to secure the access to smartphones and attempts to clarify correlated buzzwords, with the motivation to assist new researchers in understanding the gist behind those concepts. We also present the assessment of existing user authentication schemes exhibiting their security and usability issues.

1. Introduction

The birth of smartphones can be traced back to 1973, when Motorola launched their first phone—the Dynatac 8000X [1]. In the last 40 years, mobile device manufacturers have invested heavily in the innovation of mobile phones, transforming a device invented merely for calling and short text messaging into the personal, portable and powerful device of nowadays, equipped with many advanced software and hardware features.

Smartphones, undoubtedly, bring rich digital experiences to the users by offering personalized services, for example, chatting, e-mailing, GPS-navigation, net banking, online shopping, social networking, and video conferencing. Most of these services collect and store a large amount of the user's personal data on the device; thus, any unauthorized access to the user's data could have unfavorable consequences. Hence, it becomes extremely important to prevent any unauthorized access to the smartphone. Typically, access to modern smartphones is secured by enabling different authentication solutions, such as PINs/passwords, face recognition, and fingerprint.

By and large multiple terminologies in the field of authentication are being used by researchers not always with clear definitions, which is obviously disconcerting for students and new researchers. Triandopoulos et al. [2] described one-time authentication as "one-time passcodes" or "one-time password" (OTP) as the second authentication factor, although OTP is a more widely accepted term. Crouse et al. [3] described continuous authentication as a periodical composition of one-shot authentication. However, Feng et al. [4] mentioned periodic authentication as equivalent to automatic logouts due to user's inactivity. Patel et al. [5] considered continuous authentication and active authentication systems as the same. Similarly, Dutt et al. [6] suggested the use of transparent modalities in conjunction with explicit authentication methods, such as passwords, PINs, or secret patterns for authenticating users, whereas the study by De Luca et al. [7] considered the use of a transparent modality with or without other schemes and termed it *implicit authentication*. That modality could be used as standalone or to complement the explicit authentication schemes to enhance

their usability [8, 9]. More specifically the concept of *transparent authentication* is explained as implicitly fingerprinting the user's device interaction logs to authenticate the user [10].

Causey [11] considered *risk-based authentication* similar to an *adaptive authentication* scheme. Traore et al. [12] described *risk-based authentication* on the basis of contextual and historical information, extracted from their activities, to build users' risk profiles, for making later the authentication and authorization decisions. Ayed [13] patented the idea for *adaptive authentication* in mobile phones by specifying that *adaptive authentication* uses different authentication methods and different data protection methods depending on the user's location, availability of the network, and the importance of the data. It is pretty much evident from the above discussion that these definitions are correlated, but there is need to relate them to each other by trying to provide consistent definitions for all these terms.

We start this paper by explaining the prevalent ways to authenticate humans along with different types of authentication mechanisms, in the context of smartphones. Then, we try to homogenize different terminologies used in the context of user authentication with the vision that it will benefit the new researchers in understanding existing approaches. Our contribution can help new researchers to get acquainted with different user authentication concepts along with the assessment of their solutions on the basis of modalities, usability, and security.

The rest of work is organized as follows: Section 2 presents the different ways and types of authentication mechanisms. Ways refer to the common factors used to authenticate humans, while types refer to different authentication mechanisms, for example, one-shot, multifactor, continuous, and multimodal, utilizing these factors. Also, we discuss design goals for usable authentication systems and usability evaluation methods. Section 3 surveys the different state-of-the-art solutions proposed over the years for user authentication on smartphones. The related work on the ways and types of user authentication concepts available for smartphones is evaluated on the basis of their usability and security. Finally, Section 4 concludes the paper.

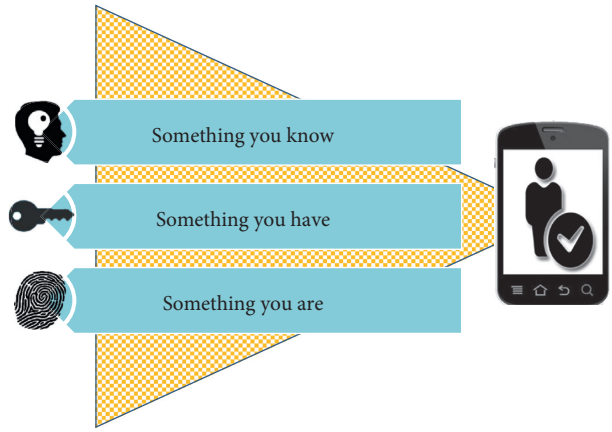


Figure 1: Ways to authenticate humans.

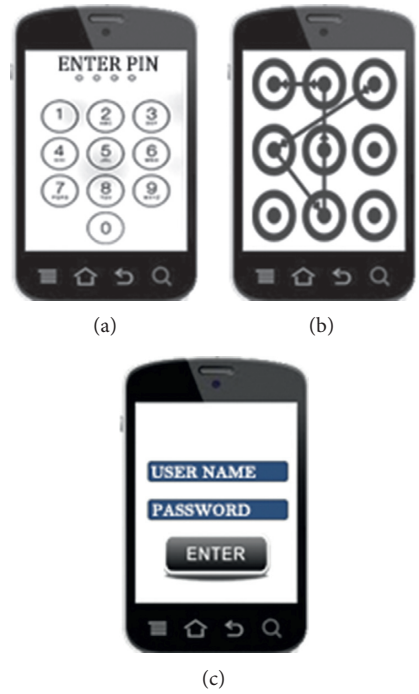


Figure 2: (a) PIN, (b) graphical pattern, and (c) password.

2. Comprehensive Study

In this section, we explain the ways to authenticate the users and the types of authentication mechanisms developed using them, in the context of smartphones.

2.1. Ways to Authenticate Users. The ways in which humans can be authenticated are broadly categorized in three categories [14], that is, "Something you know," "Something you have," and "Something you are," as depicted in Figure 1.

2.1.1. Something You Know. Knowledge-based authentication (KBA) schemes, that is, PINs (Figure 2(a)), graphical passwords (Figure 2(b)), and password (Figure 2(c)), are the most widely used schemes on the smartphones. KBA is based on some sort of a secret knowledge that user sets up earlier during the enrollment and needs to remember as long as he or she continues using the scheme.

2.1.2. Something You Have. This mechanism is also referred as token-based authentication. Many service providers and financial institutions are offering sensitive services, such as net banking, e-wallet, and e-commerce, adopting 2-factor authentication, that is, one-time passcodes (OTPs) along with usual username/password for authentication purpose. Service providers usually supply a small security device to each of their users for generating the one-time passcodes.

OTP schemes can be easily implemented on smartphones (Figure 3(a)) which could be sent either via SMS on the registered number or user could generate this OTP offline (Figure 3(b)) on the mobile apps provided by service

(a) (b) (c)

Figure 3: (a) One-time passcode (OTP) via SMS, (b) offline OTP using app, and (c) paired devices.

providers. Additionally, wearable devices (Figure 3(c)) could be used for receiving the OTPs via SMS.

2.1.3. Something You Are. This authentication mechanism relies on the measurement of biometric characteristics of users and is further classified as physiological and behavioral biometrics. Figure 4 illustrates the commonly available authentication ways for smartphone users under this category.

On smartphones, physical traits, that is, ear and face, can be collected using the built-in hardware, that is, camera; however, fingerprint and iris recognition require additional dedicated hardware. Similarly, behavioral biometric modalities, such as gait, grip, swipe, pickup, touch, and voice, can be profiled unobtrusively, using various built-in sensors [15], namely, accelerometer, gyroscope, magnetometer, proximity sensor, touch screens, and microphone. Touch-based solutions authenticate users based on their unique interactions with the device, while they perform a specific task. Additionally, behavioral biometric-based authentication is cost-effective; they generally do not require any special hardware and are considered lightweight in implementation [8].

2.2. Types of Authentication Mechanisms. Researchers have been investigating the utilization of different ways, that is, PIN, passwords, OTP, face, touch, and so on, to design and develop the different types of authentication solutions. These types are briefly explained below:

2.2.1. One-Shot Authentication. One-shot authentication is a type of authentication mechanism in which users' credentials are verified at the beginning of the session [16–18]. This is simply a process where a user claims his or her identity by providing the correct credentials or fulfilling the challenges in order to gain the access to a device. For example, PINs, passwords, graphical patterns, fingerprints, face, and iris are some of the commonly used modalities on the smartphones, for authenticating users. If the verification is successful (e.g., right password is entered), the access is granted; otherwise, the access is denied. Session remains valid until the user signs off or closes the session.

2.2.2. Periodic Authentication. Periodic authentication is simply the variant of "one-shot authentication" in which idle timeout duration is set, for closing the session, automatically [4, 19]. If a user remains inactive for more than the idle timeout duration, the device locks itself.

2.2.3. Single Sign-On (SSO) Authentication. Single sign-on (SSO) is a long-term or persistent authentication type in which a user remains signed on till the time he or she revokes or terminates the session. In case, if the system observes any discrepancy with respect to fix set of attributes, for example, change in location, network connection, and anomaly in usage pattern, the session is terminated or the user is asked for reauthentication [20–22]. VMware identity manager provides APIs to implement mobile sign-on authentication for airwatch-managed Android devices [23]. Similarly, Google offers G Suite apps for single sign-on for Android devices which can be done by pairing smartphones with smartwatches [24].

2.2.4. Multifactor Authentication. Multifactor authentication utilizes the concept of combining 2 or more authentication ways, that is, e-mail verification, OTP via SMS, phone call to the predefined numbers, push notification to the paired device, smart tokens, and so on, along with the usual method of authentication [25–27]. A very common practice is registering ones mobile number with service providers, and whenever the corresponding user accesses that service for sensitive operation, for example, online banking, service provider sends the one-time passcodes (OTPs) via SMS, getting assured that a legitimate user has requested access to that service.

2.2.5. Static and Dynamic Authentication. The static authentication mechanism presents the fixed set of challenges to the users, whereas dynamic authentication mechanism capitalizes the concept in which diverse set of prestored challenges are presented every time users unlock their smartphones [28, 29].

2.2.6. Continuous Authentication. As the name implies, continuous authentication mechanisms are developed to authenticate a legitimate owner throughout their entire session. If any anomaly is detected by the device, the access to the device is stopped, immediately, and the device asks for explicit reauthentication [4, 29, 30]. In other words, the users are passively and periodically monitored throughout their interactive session with any device or system [5]. This concept

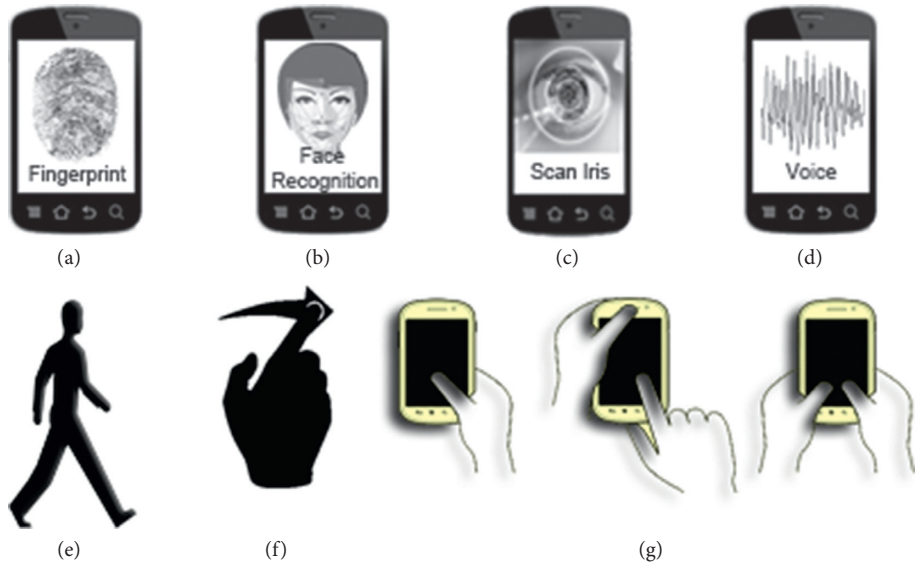


Figure 4: (a) Fingerprint, (b) face, (c) iris, (d) voice, (e) gait, (f) swipe, and (g) touch.

seems to promise higher security as compared to the other authentication mechanisms, such as *one-shot* authentication, *one-time* authentication, and *periodic* authentication, but at the same time much more complex to implement. Additionally, it is desirable that a *continuous* authentication system should not interrupt the user's normal activity and be lightweight, that is, on battery consumption.

2.2.7. Transparent Authentication. This concept stresses more on the procedure of collecting and analyzing user authentication identifiers [4, 10]. More specifically, if the system performs authentication steps in background (without requiring explicitly user cooperation) [10, 31], they are termed as *implicit, transparent, or unobtrusive* authentication systems. However, various authentication types (one-shot, risk-based, or continuous) could collect input transparently.

2.2.8. Risk-Based Authentication. Risk-based authentication schemes are mostly based on nonstatic authentication decision engine, where the decision to accept or reject authentication is based on a risk score computed in real-time, which is compared with the stored risk profiles of the users, and then the system challenges the users for authentication [32], accordingly. For instance, if a user is checking a bank account balance from a verified secure location (home or workplace), verification of identity should not be required. While in case of nonverified location, for example, the service requires additional evidence about the identity of the user thus asking for the authentication credentials. Nowadays, risk-based authentication schemes tend to offer frictionless authentication providing user experience, that could be tailored as per threats observed by the service providers [11, 12, 33, 34].

2.2.9. Adaptive Authentication. Adaptive user authentication boasts the concept having ability to change and to prepare for different conditions and situations, while securing any unauthorized access [13, 35, 36]. It entails for multifactor user authentication mechanisms which should be readily configurable and deployable.

2.2.10. Unimodal and Multimodal Authentication. This term is typically used for biometric authentication schemes. The literal meaning of modality (https://dictionary.cambridge.org/dictionary/english/modality) is a particular way of doing or experiencing something. This concept is based on the number of modalities or traits being used in the authentication systems [37–39]. Unimodal authentication systems leverage only a single biometric modality or trait, whereas multimodal systems are developed by combining two or more modalities. Multimodal authentication systems demonstrate several advantages, such as higher recognition rate, accuracy, and universality [39].

2.3. Usable Authentication System Design Goals. Usability along with security plays a pivotal role in evaluating user authentication schemes. This leads to an important question—how to trade-off between security and usability [40]? We present the guidelines described by Yee for usable security designs [41]. Yee's work focused on addressing valid and nontrivial concerns specific to usable security. We explain below the design goals from usability perspective as suggested in [41]:

(i) *Appropriate boundaries*: this goal is based on *the principle of boundaries* [42]. In order to distinguish among objects and actions along the boundaries, which are relevant to users, system should expose the boundaries and must acknowledge the users. For example, in the context of mobile devices, popular Operating Systems (OS), such as Android (Ver. 6 onwards) and iOS, allow users to grant permissions to the applications and services accessing resources while installing them. Here,

the object could be assumed as the apps or services for the devices and actions could be defined as the indicators that the apps or services demand from users to serve them and to use the system's resources. However, boundaries are the thin line that defines the users' decisions affecting the security of system due to human factors.

(ii) *Path of least resistance*: choosing the most natural method in granting the authority is the most secure way.

(iii) *Explicit authorization*: any authorization to other actors must only be granted in accordance with user actions which should be well understood by a user while acknowledging the consent.

(iv) *Visibility*: a user should be aware of others' active authority affecting any security-relevant decisions.

(v) *Revocability*: a user should be able to revoke others' authority to access the system.

(vi) *Self-awareness*: maintain accurate awareness of the user's own authority to control the system.

(vii) *Trusted path*: protect the user's channels to any entity that manipulate authority on the user's behalf.

(viii) *Identifiability*: any specific objects and specific actions must be clearly identifiable and apparent to the user.

(ix) *Expressiveness*: enable the user to express safe security policies in terms that fit the user's goals.

(x) *Clarity*: notify the consequences of any security-relevant decisions precisely that the user is most likely to perform.

2.4. Usability Evaluation. System usability scale (SUS) questionnaire [43] is utilized to gather subjective assessments about the usability of the proposed systems [8]. The questionnaire consists of 10 questions or statements. The response to each question/statement is measured on a 5-point scale ranging from "strongly disagree" to "strongly agree." The final SUS score ranges between 0 and 100, where a higher value indicates a more usable system. The system usability scale (SUS) template for questionnaire and scoring is available online [44].

3. Literature Review and Analysis

In this section, we review the recent literature emphasizing on the types of authentication mechanisms and the ways on which they are developed and analyze them from security and usability point of view. More specifically, we present the assessment of commonly used user authentication mechanisms on smartphones, focusing on the security and usability issues.

3.1. Ways of Authentication. The usability of authentication mechanisms is one of the dominant attributes that influence users' acceptance of a particular authentication scheme [45]. The ISO standard:13407 defines usability as "*the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction, in a specified context of use*" [46]. Further, the study [47] suggests that the usability can be done on the basis of three criteria: task performance, user satisfaction, and user cost.

Conventional authentication schemes, that is, PIN, passwords, and graphical patterns, are no more considered secure and convenient [48] because they are not able to distinguish between the users, rather they authorize everyone (regardless of whether that person is the legitimate owner of the device or not) who enter the correct credentials. Physiological biometric-based solutions are considered more secure because it is assumed that human body traits cannot be shared, copied, lost or stolen. Moreover, they genuinely authenticate their users by forcing them to present themselves physically to the system. However, they are less preferable on smartphones due to their inherent usability issues [49]. As such, security experts are focusing on developing the usable authentication systems because they believe that behavioral biometrics will restructure the authentication landscape in the next 5–8 years [50].

In each subsections, we have included tables presenting the synopsis of each authentication ways being used as different authentication types along with the references that either indicating usability pros and cons or reporting security solutions and concerns.

3.1.1. Something You Know. As per the web report [51], average smartphone users get themselves engaged in 76 separate phone sessions, while heavy users (the top 10%) peaked to 132 sessions per day. PIN/passwords, and graphical patterns, require users to memorize their text, they had set earlier, to unlock their devices, every time they need to initiate the session (76 times a day). The capacity of the human brain to process the information varies from person to person [52]. Zhang et al. [53] found that users faced problems in remembering their passwords and more especially, to memorize and correctly recall numerous passwords. This encouraged users going for an easy or simple password which is quick to remember [54], but this opens plenty of opportunities for attackers to guess or crack their passwords, easily [55]. When the system enforces stringent password policies, users due to memorability issues [56], allow their browsers or password managers to save their username/password information to make future logins easier. However, users trusting their browsers or password managers are more likely to be a victim of a wide variety of attacks [57, 58]. Overall, 82% of end users are frustrated with managing passwords [59]. Clearly, this indicates the lack of usability, and a result, nearly, 75 million smartphones users in the US do not use any of PIN, pattern, or passwords because they consider them annoying and an obstacle in quick access to their smartphones [60].

From security perspective, PINs and passwords are vulnerable to various attacks, for example, guessing [61], because users choose date of births [57], easier digits (1111, 2222, etc.,) [62] to set up their PIN. Alternatively, Android

Table 1: Synopsis of knowledge-based schemes.

Modalities	Authentication types	Usability pros and cons indicated	Security solutions or concerns reported
PIN [57, 60, 62]; password [53, 60]; pattern [60, 64]	One-shot; static; periodic; single sign-on; unimodal	[48, 52–57, 60, 62, 71, 72]	[58, 61–70]

Table 2: Synopsis of token-based schemes.

Modalities	Authentication types	Usability pros and cons indicated	Security solutions or concerns reported
OTP [75]; device pairing [77, 79]	Multifactor; adaptive; dynamic; risk-based	[40, 79, 80–82]	[73–78]

users (40% of them) prefer graphical patterns for device unlocking. But this approach too requires users to remember them; hence users choose simple and less secure patterns, that is, if a user connects at least four dots without repeating any of them in their patterns, the maximum number of combinations are 389,112 which could be easily cracked by brute force [63]. Ye at al. [64] managed to crack 95% of 120 unique patterns collected from 215 independent users within just five attempts by recording their smartphone screen, remotely, while they were unlocking their devices. In addition, these schemes are more vulnerable to shoulder surfing than textual passwords [65].

Knowledge-based authentication schemes are generally used as one-shot, static, or unimodal authentication types (refer Table 1) due to usability issues they are prone to several attacks, such as smudge attacks [66], shoulder surfing or observation attacks [61, 67], dictionary-based attacks, or rainbow table password attacks [68]. Recently, Mehrnezhad et al. [69] demonstrated the recovery of entered PIN or password from the sensory data collected, while the users were entering their secrets. They installed PINlogger.js—a JavaScript-based side-channel attack, capable of recording motion and orientation sensor streams without requiring any user permission from the user. The attack resulted in 94% accuracy in recovering the correct PIN number in just three rounds of tries. Similarly, Sarkisyan et al. [70] demonstrated an approach to exploit smartwatch motion sensors to recover the entered PINs. They infested smartwatches with malware to get access to the smartwatch motion sensors and inferred user activities and PINs. In a controlled scenario, authors obtained PIN numbers within 5 guesses with an accuracy of at least 41% using random forest classifier over a dataset of 21 users.

3.1.2. Something You Have. As defined in Section 2.1.2, smartphones are being utilized for authentication purposes in several sensitive operations by the means of OTP via SMS, offline OTP using Apps, or pairing the wearable devices, for example, smartwatches, smartglasses, and smartcards. However, this idea of enhancing security with multifactor authentication, that is, topping *knowledge-based authentication* with *token-based authentication* (one-time passcode), eventually perishes too due to side-channel attacks, for example, MITM (man-in-the-middle) and MITPC/Phone (man-in-the-PC/phone) [73]. Software-based OTP solutions also do not guarantee the confidentiality of the generated passwords or the seeds as the mobile OS could be compromised, at the same time, could also suffer from denial-of-service attacks on the account of mobile OS crashes [74].

The adversaries by the means of real-time phishing or intercept attacks could reveal the users' secret information and valid OTP by breaking into their smartphones [75]. As per the Verizon Data Breach Investigations Report [76], NIST stopped recommending the users for two-factor authentication via SMS, as malicious code infesting mobile endpoints could surreptitiously capture second factors delivered by SMS or offline OTP generated using apps. Secure device pairing schemes allow access to the smartphones by pairing it with a trusted Bluetooth device like a smartwatch and use the same to unlock the phone. This concept from the usability point of view is a very elegant solution but not safe from insider attacks or sniffing attacks [77, 78].

Token-based authentication (TBA) schemes are used in multifactor, adaptive, dynamic, and risk-based authentication types (Table 2). Unfortunately, they could not add too much to the usability because the users are required to manage always an additional hardware for the sole purpose of authentication. As a result, Braz and Robert [40] gave usability rating 3 (out of 5) to one-time generator acquisition devices. Additionally, Belk mentioned that token-based authentication mechanism incurred more cost to users and are comparatively slower [79]. According to a study by Zink and Waldvogel [82], 83.3% users considered that SMS-based transaction authentication number is not a usable solution. Another in-depth usability study by Krol et al. [81] evaluated 2-factor authentication on 21 online banking customers (16 among 21 were having multiple accounts with more than one bank). Total 90 separate login sessions of all the participants were collected meticulously, over the period of 11 days. Their analysis showed approximately 13.3% faced problems due to mistyped credentials, misplaced token, forgotten credentials and so on.

3.1.3. Something You Have: Insertable Biometrics. Insertable biometrics [83–85] (Table 3) including implantable medical devices (IMDs) [86] and emerging technologies such as Bespoke devices [87, 88], neodymium magnets [89], NFC or RFID chips [90, 91], smart piercings [92, 93], and smart tattoos [93] are the newer addition to biometrics that

Table 3: Synopsis of insertable biometrics.

Modalities	Authentication types	Usability pros and cons indicated	Security solutions or concerns reported
Bespoke devices [87, 88]; neodymium magnets [89]; NFC or RFID chips [90, 91]; smart piercings [92, 93]; smart tattoos [93]	Continuous; multimodal; transparent	[94]	Data not available

Table 4: Synopsis of physiological biometrics.

Modalities	Authentication types	Usability pros and cons indicated	Security solutions or concerns reported
Face [99, 100]; eyes [10, 101]; iris [102]; fingerprint [103, 104]	One-shot; multifactor; multimodal	[49, 105–109]	[17, 95–97, 99, 100, 102–104, 110]

potentially can be used to provide increased usability over the existing solutions [94]. Researches are exploring the further possibilities of insertable biometrics as go-to solution for improving digital security and usability in smartphones.

3.1.4. Something You Are: Physiological Biometrics. Mobile device manufacturers have started embedding biometric sensors in their flagship smartphones for reliable and convenient user authentication with the intuition that biometric approaches are better than their conventional authentication schemes. For example, Apple, Huawei, Lenovo (Motorola), Microsoft (Nokia), Samsung, and many other leading manufacturers have integrated fingerprint sensors, iris scanners, and face recognition algorithms, in some of their high-end devices. These advancements are akin to replacing a hay castle with a glass house to ward off attacks from sophisticated cyber pirates.

Physiological biometrics, for example, face, fingerprint, iris, and eyes, are commonly used as one-shot or multifactor/multimodel (combining with other modalities) authentication schemes for smartphones (Table 4). Unexpectedly, biometric systems have shown to be exposed to different types of attacks, for example, impersonation, replay, spoofing, and hill climbing [95], exposing their security loopholes. These schemes suffer from their data leakage; that is, a user's face can be easily found on social media websites, or his or her fingerprints can be extracted from the photos from their gestures, to mount a presentation attack [96] against them. Additionally, these solutions also suffer from lack of secrecy [97] and vulnerability to various spoofing attacks [98].

Recent research has shown that these schemes can be hacked very easily with almost negligible investment and efforts. For example, iPhone X face ID was hacked with 3D-printed mask costing just $150 approximately [100], while Samsung S8 facial recognition technology [99] was simply fooled with a photo of the owner. Similarly, German Chaos Computer Club cracked the Samsung Galaxy S8 iris scanner [102] with a dummy eye made from pictures of the iris, taken by a digital camera in a night mode, and covered it with a contact lens to match the curvature of the eye, within a month of S8 launch. The same club earlier cracked the iPhone 5S fingerprint sensor protection within two days after the device went on sale worldwide [103]. Their hacking team photographed the glass surface containing the fingerprint of a user and created a "fake fingerprint" using a thin film to unlock the phone. Japan's National Institute of Informatics (NII) researcher Isao Echizen [104] demonstrated that fingerprints can easily be recreated from photos, taken just from three meters distance, without the use of any sophisticated process and warned casually making a peace sign in front of a camera, which could lead to fingerprint theft.

From the usability perspective, smartphone users have not shown optimistic inclination to physiological biometric-based authentication schemes. For example, De Luca et al. [49] determined smartphone users felt like as if they are taking selfies all day to authenticate themselves. Additionally, the performance of these schemes is affected by several exogenous factors, such as accessories, camera movement, capturing distance, clothing, illumination, interoperability of the sensors, noise, occlusion, operators, postures, and training, which makes the authentication process more challenging and less usable to the user [106–109].

3.1.5. Something You Are: Behavioral Biometrics. Behavioral biometrics [111] is described as the future of user authentication. Thus, the focus of the research has been shifted to develop newer behavioral biometric-based solutions. For example, applications like e-wallet, m-commerce, and mobile banking are some of the sensitive domains, where behavioral biometric-based solutions have shown to be handy in authenticating the customers on their smartphones.

Although the behavioral modalities are not considered to be unique enough for identification purposes, they have proved to be sufficiently unique for user authentication [112, 113]. One or more modalities can be combined to increase their accuracy and enhance their usability. These schemes could be stitched to the existing user authentication mechanisms as an additional transparent authentication layer [8, 9, 114] enhancing the reliability of whole authentication process without affecting the usability. Behavioral biometric techniques could be deployed as adaptive, continuous, multimodal, risk-based, transparent authentication (Table 5).

Gait recognition is a process of identifying or verifying individuals on the basis of their walking style. In clinical applications, human gait was already getting utilized for the studies related to the health of a person, and nearly 25 key patterns from gait were detected using different techniques like

Table 5: Synopsis of behavioral biometrics.

Modalities	Authentication types	Usability pros and cons indicated	Security solutions or concerns reported
Touch [9, 113]; keystroke [115]; hold [8]; gait [116–118]; behavior profiling [119]	Adaptive; continuous; multimodal; risk-based; transparent	[3, 5, 10, 113, 119–121]	[8, 12, 29, 112, 113, 115–117, 122–127]

image processing, floor sensors, and sensors placed on the body [118]. Recently, smartphones and wearable devices have also started utilizing it for authentication purposes [128]. As users are not required to perform any explicit interaction with their devices, gait modality can be collected unobtrusively, and this leads to making it convenient for a user-friendly access system [116]. Muaaz and Mayrhofer [116] evaluated the security strength of a smartphone-based gait recognition system against zero-effort and live-minimal-effort impersonation attacks under realistic scenarios and achieved an equal error rate (EER) of 13% on a dataset of 35 participants. However, more testing is required to check the robustness against impersonation attacks. Hestbek et al. [117] introduced a method using wearable sensors and noncyclic feature extraction and achieved 18.92% half total error rate (HTER) on a dataset of 36 users. Similarly, the grip is another natural way to authenticate users. It is robust too as the finger movements and pressure applied while gripping the mobile device are visibly unseen and difficult to be replicated or imitated by the impostor. Murao et al. [124] proposed a grip-based authentication solution, which profiles grip gestures using pressure sensors mounted on the lateral and back sides of a smartphone and achieved a 2% ERR, which is equivalent to face recognition-based authentication.

Keystroke or touch dynamics refers to the typing characteristics (due to the timing differences) of individuals to fingerprint their identity. Researchers have proved its effectiveness in both fixed text and text independent scenarios. Since designing such systems does not require any additional dedicated hardware and data can be collected, unobtrusively, they have been widely tested and evaluated [9, 114]. Zheng et al. [115] proposed authentication mechanism based on tapping; they collected tapping data from over 80 users; and their system achieved high accuracy with averaged 3.65% EER. Another bimodal authentication scheme developed using client-server architecture for online financial environments achieved 96% true acceptance rate (TAR) and 0.01% false acceptance rate (FAR) using 15 training samples on a dataset of 95 users [9]. This scheme used motion-based touch-types biometrics, that is, touch typing and phone movements by users and collected data, transparently, while users entering their credentials to sign in to their banking apps using 8-digit PIN/password [9], while the "touchstroke" scheme used 4-digit PIN/password [114]. Buriro et al. [8], proposed, implemented, and evaluated the "Hold and Sign" scheme on commercially available smartphones and achieved 95% TAR on a dataset of 30 volunteers. This was a bimodal behavioral biometric based on user's smartphone holding style, by examining the hand and finger micromovements of users, while the users were signing on device's touchscreen. In an another approach, Buriro et al. [113] proposed multimodal behavioral biometrics (swipe, pickup movement, and voice) for user authentication on smartphones and reported 7.57% HTER in an experiment involving 26 participants.

Brunet et al. [123] experimented on voice modality for user authentication on a public database (Sphinx Database of the Carnegie Mellon University [129]). They digitized the user's voice and extracted Mel Frequency Cepstral Coefficients (MFCCs) features and computed the Euclidean distance to authenticate the user and reported an EER of 4.52%. Behavior profiling techniques were based on the applications, and the services utilized in past for generating a user profile and compared it against the current activity of a user in real-time [5]. If any significant variation is observed, the system could take action for a possible intrusion. Sultana et al. [119] combined social behavioral information of individuals that was extracted from the online social networks to fuse with traditional face and ear biometrics, to enhance the performance of the traditional biometric systems.

Studies suggest that no single biometric trait can ideally fit all the scenarios; however, by trying multimodal biometric approaches, most of the limitation of unimodal systems can be addressed [121, 122, 125]. The selection of proper modalities and combining them, systematically, most of the times increase the accuracy, usability, and security. In a study conducted by Saevanee et al. [126], the unimodal systems, namely, behavior profiling, keystroke dynamics, and linguistic profiling, were proved less accurate; they yielded an EER of 20%, 20%, and 22%, respectively. However, by applying matching-level fusion, the error rate was decreased, significantly (EER 8%). Additionally, the use of users' transparent characteristics for data collection and classification also increases the usability of the system. Thus, in order to furnish users with an adequate security, a better usability is also required to design the authentication solutions for smartphones.

3.2. Authentication Types

3.2.1. One-Shot Authentication. *One-shot authentication* schemes are designed to authenticate a user at the initiation of a session (subject's identity is verified only once, just before allowing access to the resources) [16, 18]. Roth et al. [18] also discussed the limitations of one-shot authentication, such as short sensing time, inability to rectify decisions, and enabling the access for potentially unlimited periods of time. Meng et al. [17] introduced the term one-off authentication for one-shot authentication. They also concluded that authenticating just once leaves the possibilities for impostors to gain the access to the current session and retrieve sensitive information from mobile phones.

3.2.2. Periodic Authentication. Bertino et al. [19] defined *periodic authorization* with a mathematical expression "{[begin,

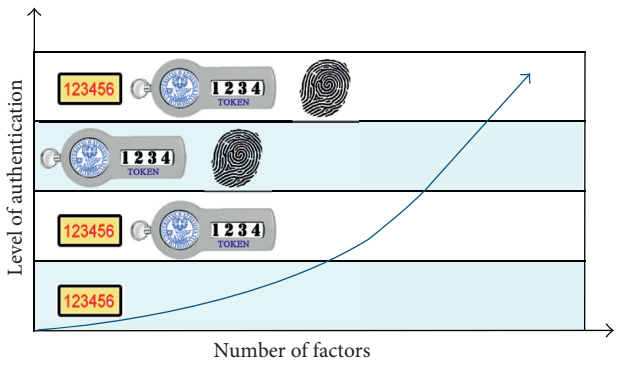


Figure 5: Factors of authentication [27].

end], P, auth}" holding of 3 prime attributes, where "begin" is authorization start date, "end" is either the constant ∞, or a deauthorization date after the start date, "P" is the duration of a session, and "auth" is an authorization function. Feng et al. [4] determined that periodic authentication or automatic logouts are more detrimental while one-shot authentication solutions are prone to a wide variety of attacks. Typing an error-free username and/or password on smartphone's keyboard is really a tedious task, especially when an average user initiates 76 phone sessions a day [51]. *Single sign-on* (SSO) has been seen as the solution to the problem.

3.2.3. Single Sign-On. *Single sign-on* (SSO) enables users to sign in to an app using a single or federated identity, for example, Facebook, Twitter, and Google+. But this concept is severely risky for mobile devices as they are more likely to be misplaced or could be inadvertently shared with someone. In an SSO system, the user is authenticated to a single identity provider (IDP) which acts as a trusted party between the user and multiple service providers (SPs), and on the demand of the user, IDP generates an authentication token for a specific SP asserting the users' identity; in turn, SP allows the user to access the services [20]. Users can access different applications using SSO, once they are authenticated to the system. SSO is further divided into two categories, that is, Enterprise Single Sign-ON (ESSO) and Reduced Sign-ON (RSSO) [21]. ESSO enables a user to enter the same id and password to sign into multiple applications within an enterprise domain. The system is considered the least secure because there could be potential curious adversary which can try to spoof and consequently resulting in an identity theft. Therefore, it is also known as RSSO.

3.2.4. Multifactor Authentication. Security experts also suggest the use of *multifactor authentication* by processing multiple factors, simultaneously, for the verification purposes [27]. In multifactor authentication, generally, a PIN or password is the baseline authentication standard, while more factors can be augmented from a wide variety of available sources to verify users (Figure 5). It could be observed in Figure 5 that as the number of factors increases, the level of authentication also increases. For an instance, if only PIN is used, the authentication level is minimum, but when other factors like tokens and fingerprints are added, the authentication level tends to increase proportionally.

Figure 6: Static authentication process [29].

The most common authentication mechanism is the secondary code that can be delivered either via SMS to the registered mobile number or can be obtained directly from a secure authenticator mobile app. Other forms of multifactor authentication involve the use of a smart card or smart token entitled to the user, biometrics like the face or fingerprint scans, or a dedicated code generator linked to user's account [25]. This concept is mainly influenced by the notions that not all the authentication factors could be hacked at the same time. Stanislav [26] in his paper explained various technical methods by which two-factor authentication can be implemented.

3.2.5. Static versus Dynamic Authentication. *Static authentication* process, like other authentication types, mainly consists of three steps: enrollment, presentation, and evaluation as illustrated in Figure 6, and the outcome of the evaluation is a binary decision [29]. In the enrollment step, system generates a feature template by processing the information gathered from the user, profiles the feature vectors with the label of the user, and saves it for the evaluation or matching. During the presentation step, system asks the user to confirm his or her credentials. In the final step, that is, evaluation, information given by the user is compared with the stored templates of the claimed identity. Conclusively, the access is granted or denied as per the match result.

Static authentication verifies the individual's identity only at the start of a session like one-shot authentication does, whereas in *dynamic authentication* the user is presented with a varying set of challenges to enable the dynamic scaling of access controls. Ren and Wu [28] explained dynamic authentication as a scheme that utilizes one-time password derived from the user's password, the authenticating time, and a unique attribute only known to the user.

3.2.6. Continuous Authentication. *Continuous authentication* is a mechanism to repeatedly verify the identity of a user for the entire duration of an authorized session as illustrated in Figure 7 [29]. More specifically a continuous authentication is an approach that constantly verifies a user's identity and locks the system once the change in users' identity is observed [29]. Continuous authentication process dynamically iterates in between the three steps involved (Figure 6) throughout the session. However, these iterations can be event-based or can be adjusted at fix intervals (periodically) or randomly [29]. A continuous authentication is an approach that constantly verifies a user's identity and locks the system once the change in user identity is observed. Thus, overcoming the limitations of one-shot authentication, where authentication happens only at the time of login, and any future changes in user identity go undetected [130]. Behavioral biometric-based

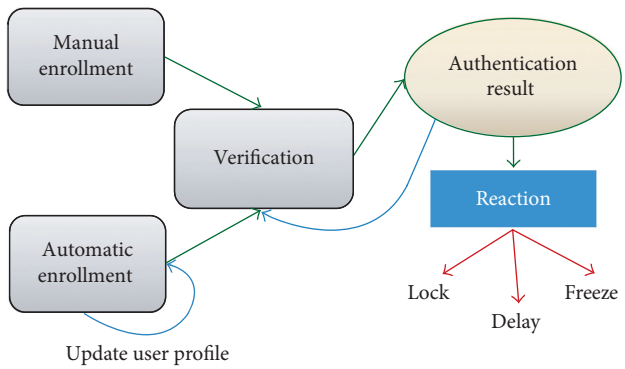


Figure 7: Continuous authentication process [29].

continuous authentication solutions have shown to be more attractive to the researchers of the domain because these behavioral modalities can be collected and utilized, unobtrusively, for authentication purposes [30].

However, continuous authentication, active authentication, implicit authentication, and transparent authentication have been interchangeably used in many papers [10, 120, 131, 132]. Patel et al. [5] considered continuous authentication and active authentication systems as similar and explained it as continuous monitoring of the user activities after the initial access to the mobile device. Active authentication, as defined by Stolerman et al. [132], is the process of continuously verifying users based on their on-going interaction with the device. The Defense Advanced Research Projects Agency (DARPA) started Active Authentication program [133] in order to seek solutions by shifting the focus during authentication from the password to people themselves. The first phase of their Active Authentication program focused on the behavioral traits, that is, cognitive fingerprint, which could be processed without the need for additional sensors.

According to Fridman et al. [134], active authentication is the problem of continuously verifying the identity of an individual. They conducted an experiment using Android mobile devices and collected several biometric modalities, namely, text entered via soft keyboard, applications used, websites visited, physical location of the device as determined from GPS (when outdoors) or WiFi (when indoors), and stylometry, of 200 volunteers approximately for a period of at least 30 days. Their authentication system achieved an ERR of 0.05 (5%) after 1 minute of user interaction with the device, and an EER of 0.01 (1%) after 30 minutes in identifying a legitimate user. In another stylometric-based continuous authentication, an EER of 12.42% for message blocks of 500 characters is achieved using support vector machine (SVM) for classification [135]. However, stylometry-based authentication schemes must improve the accuracy, delays, and forgery.

Khan et al. [120] mentioned that *implicit authentication* employs behavioral biometrics in a continuous and transparent manner to recognize and validate smartphone users' identity and conducted a field study on implicit authentication usability and security perceptions with 37 participants. Their experiment indicated that 91% of participants found implicit authentication to be convenient and 81% perceived defined the protection level to be satisfactory.

3.2.7. Transparent Authentication. *Transparent authentication* [10] was suggested as an alternative authentication mechanism with minimal or no noticeable involvement of users. Transparent authentication implicitly authenticates the users on the basis of their unique interactions with the device and creates a logic for authentication decisions. Feng et al. [4] utilized the term transparent and continuous for their Finger-gestures Authentication System using Touchscreen (FAST) to protect the mobile system. The approach transparently captures the touch data without intervening to user's normal user-device interactions. After the user's login, FAST continues to authenticate the mobile user in the background using intercepted touch data from their normal user-smartphone interactions.

3.2.8. Risk-Based Authentication. ClearLogin [136] defines *risk-based user authentication* as a method which adapts authentication levels based on the apparent risks, to mitigate the potential intrusion, before they happen. Existing *risk-based user authentication* schemes generate a risk profile to determine the complexity of challenge to authenticate a user during a session, that is, higher-risk profiles lead to stronger authentication, whereas usual authentication scheme should be sufficient in normal scenarios [137]. Identity Automation [138] considers *risk-based user authentication* similar to *adaptive authentication* because they adapt to the stringency of authentication processes based on the likelihood that access to a given system could result in its compromise.

Earlier risk-based user authentication mechanisms were mainly based on contextual or historical user information or both [139]. Furthermore, these systems use ad hoc or simplistic risk management models based on some rule-based techniques, which are proved to be ineffective due to human factors [140]. However, nowadays as NuData Security [34] mentioned risk-based authentication schemes are getting fueled by behavior piercing technology that gives maximum security with minimal interruption to the user experience. Risk-based user authentication can be applied from two different perspectives: proactive or re-active [12]. When applied proactively, risk-based authentication actively anticipates the genesis of potential attacks, failures, or any kind of security issues and takes prompt action. In contrast, re-active risk-based authentication accepts some of the risks until the risk score goes beyond the permissible threshold level, and consequently, reauthentication is required.

3.2.9. Adaptive Authentication. *Adaptive authentication* [141] is a way by which two- or multifactor authentication can be configured and deployed by doing risk assessment. Thus, it is a method for selecting the appropriate authentication factors accustomed to the situation accordingly to the user risk profile and tendencies. It can be deployed as follows:

(i) By setting static policies based on risk levels for different factors, such as user role, resource importance, location, time of day, or day of the week

(ii) By learning day-to-day activities of users based on their habits to generate dynamic policies

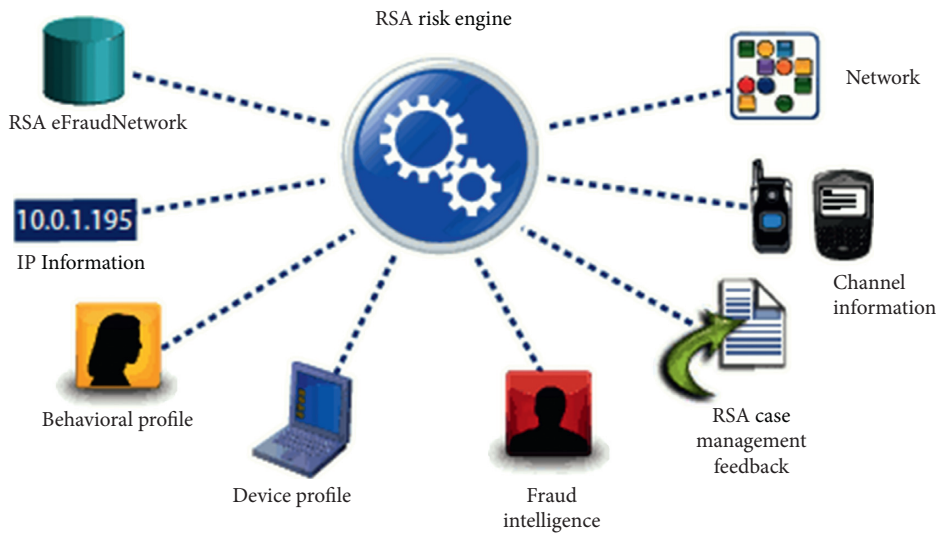


Figure 8: RSA adaptive authentication [36]. The RSA Risk Engine measures over one hundred indicators and assigns a unique risk score to each activity.

(iii) Lastly, by combing of both static and dynamic policies

Hulsebosch et al. [35] exploited the ability to sense and use context information to augment or replace the traditional static security measures by making them more adaptable to a given context and thereby less intrusive to derive context sensitive adaptive authentication. RSA Risk Engine [36] used self-learning risk model and adapts itself on the basis of received feedback. The feedback loop includes case resolution and genuine or failed authentication results as well as chargeback files for *adaptive authentication* for e-commerce (Figure 8).

3.2.10. Unimodal and Multimodal Authentication Systems. *Unimodal authentication systems* use single modality for establishing user identity, whereas *multimodal authentication systems* include multiple modalities (sources of information) [39]. Unimodal and multimodal terms are more associated with biometric systems where person recognition is based on distinctive personal traits or characteristics [37]. Unimodal physiological biometric based on face, fingerprint, and iris are already deployed on the smartphones; however, multimodal systems are yet to be deployed. Behavioral biometric-based solutions based on touch-stroke dynamics, voice, gait, and so on have been widely tested and evaluated by researchers; however, their deployment to the smartphones is still awaited.

Jain et al. [38] showed that multimodal biometric systems driven by multiple biometric sources perform, generally, better recognition performance as compared to unimodal systems. As per the type of multiple modalities being used, multimodal biometric systems can be further divided into three categories: (1) multiphysiological, (2) multibehavioral, and (3) hybrid multimodal systems [142]. The multiphysiological category includes multimodal biometric systems, where only physiological traits, such as face, fingerprint, and iris, are fused at different levels, whereas the multibehavioral system combines data from keyboard, mouse, and graphical user interface interactions. Hybrid multimodal system [143] fused face, ear, and signature with social network analysis at the decision level to enhance the biometric recognition performance.

Researchers have been actively working on combining different modalities to develop multimodal solutions; however, these systems have yet to appear on the real products.

4. Conclusion

In this paper, we presented the gist of ways and types of user authentication concepts in the context of smartphones. We surveyed the different state-of-the-art solutions proposed over the years and attempted to homogenize correlated buzzwords used in this field, with the motivation to assist new researchers in understanding these concepts. Then, we evaluated the related work on the ways and types of user authentication mechanisms available for smartphones, on the basis of their usability and security. Also, we discussed design goals for usable authentication systems and usability evaluation methods.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement no. 675320.

References

[1] Motorola, "Timely achievements," 2017, https://www.motorola.com/us/about/motorola-history-milestones.

[2] N. Triandopoulos, A. Juels, R. L. Rivest, and J. Brainard, "Multi-server one-time passcode verification on respective high order and low order passcode portions," US Patent 9,454,654, 2016.

[3] D. Crouse, H. Han, D. Chandra, B. Barbello, and A. K. Jain, "Continuous authentication of mobile user: fusion of face image and inertial measurement unit data," in *Proceedings of the International Conference on Biometrics (ICB)*, pp. 135–142, IEEE, Phuket, Thailand, May 2015.

[4] T. Feng, Z. Liu, K.-A. Kwon et al., "Continuous mobile authentication using touchscreen gestures," in *Proceedings of the IEEE Conference on Technologies for Homeland Security (HST)*, pp. 451–456, IEEE, Waltham, MA, USA, November 2012.

[5] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: recent progress and remaining challenges," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, 2016.

[6] D. C. Dutt, A. B. Somayaji, and M. J. K. Bingham, "System and method for behavioural biometric authentication using program modelling," US Patent App. 15/059,692, 2016.

[7] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and i know it's you!: implicit authentication based on touch screen patterns," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 987–996, ACM, New York, NY, USA, 2012.

[8] A. Buriro, B. Crispo, F. Del Frari, and K. Wrona, "Hold and sign: a novel behavioral biometrics for smartphone user authentication," in *Proceedings of the 2016 IEEE Security and Privacy Workshops (SPW)*, pp. 276–285, IEEE, San Jose, CA, USA, May 2016.

[9] A. Buriro, S. Gupta, and B. Crispo, "Evaluation of motion-based touch-typing biometrics in online financial environments," in *Proceedings of the 16th International Conference of the Biometrics Special Interest Group (BIOSIG)*, Darmstadt, Germany, September 2017.

[10] H. Crawford and K. Renaud, "Understanding user perceptions of transparent authentication on a mobile device," *Journal of Trust Management*, vol. 1, no. 1, p. 7, 2014.

[11] B. Causey, "Adaptive authentication: an introduction to risk-based authentication," 2013, http://searchsecurity.techtarget.com/tip/Adaptive-authentication-An-introduction-to-risk-based-authentication.

[12] I. Traore, I. Woungang, M. S. Obaidat, Y. Nakkabi, and I. Lai, "Online risk-based authentication using behavioral biometrics," *Multimedia tools and applications*, vol. 71, no. 2, pp. 575–605, 2014.

[13] M. B. Ayed, "Method for adaptive authentication using a mobile device," US Patent 8,646,060, 2014.

[14] F. B. Schneider, "Something you know, have, or are," 2005, https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html.

[15] N. Forsblom, "Were you aware of all these sensors in your smartphone?," 2015, https://blog.adtile.me/2015/11/12/were-you-aware-of-all-these-sensors-in-your-smartphone/.

[16] A. Buriro, "Behavioral biometrics for smartphone user authentication," Ph.D. thesis, University of Trento, Trento, Italy, 2017.

[17] W. Meng, D. S. Wong, S. Furnell, and J. Zhou, "Surveying the development of biometric user authentication on mobile phones," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1268–1293, 2015.

[18] J. Roth, X. Liu, A. Ross, and D. Metaxas, "Biometric authentication via keystroke sound," in *Proceedings of the International Conference on Biometrics (ICB)*, pp. 1–8, IEEE, Madrid, Spain, June 2013.

[19] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati, "An access control model supporting periodicity constraints and temporal reasoning," *ACM Transactions on Database Systems*, vol. 23, no. 3, pp. 231–285, 1998.

[20] F. Feldmann, *Binding Credentials: Securing (SSO) Authentication*, Ruhr University Bochum, Bochum, Germany, 2016.

[21] HuntingtonVentures, "Single sign on: the business of authentication," 2006, https://archive.is/20140315095827/.

[22] A. Salazar, "SSO vs. centralized authentication," 2014, https://stormpath.com/blog/sso-vs-centralized-auth.

[23] VMware, "VMware identity manager documentation center," 2017, https://pubs.vmware.com/vidm/index.jsp?topic=%2Fcom.vmware.wsair-administration%2FGUID-1E5128A5-1394-4A50-8098-947780E38166.html.

[24] Google, "G suite: single sign-on on an android device," 2016, https://support.google.com/a/users/answer/2758865?hl=en.

[25] R. Ritchie, D. Rubino, K. Michaluk, and P. Nickinson, "The future of authentication: Biometrics, multi-factor, and co-dependency," 2013, https://www.androidcentral.com/talk-mobile/future-authentication-biometrics-multi-factor-and-co-dependency-talk-mobile.

[26] M. Stanislav, *Two-Factor Authentication*, IT Governance Ltd., Ely, UK, 2015.

[27] D. G. Warnock and C. C. Peck, "A roadmap for biomarker qualification," *Nature biotechnology*, vol. 28, no. 5, pp. 444-445, 2010.

[28] X. Ren and X.-W. Wu, "A novel dynamic user authentication scheme," in *Proceedings of the International Symposium on Communications and Information Technologies (ISCIT)*, pp. 713–717, IEEE, Gold Coast, QLD, Australia, October 2012.

[29] I. Traoré and A. A. E. Ahmed, *Introduction to Continuous Authentication, in Continuous Authentication Using Biometrics: Data, Models, and Metrics: Data, Models, and Metrics*, University of Victoria, Victoria, BC Canada, 2011.

[30] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, 2013.

[31] A. Buriro, B. Crispo, and Y. Zhauniarovich, "Please hold on: unobtrusive user authentication using smartphone's built-in sensors," in *Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pp. 1–8, IEEE, New Delhi, India, February 2017.

[32] Wikipedia, "Risk-based authentication," 2017, https://en.wikipedia.org/wiki/Risk-based_authentication.

[33] A. J. Harris and D. C. Yen, "Biometric authentication: assuring access to information," *Information Management & Computer Security*, vol. 10, no. 1, pp. 12–19, 2002.

[34] NuData Security, "What is risk based authentication?," 2017, https://nudatasecurity.com/blog/ecommerce/what-is-risk-based-authentication/.

[35] R. Hulsebosch, M. S. Bargh, G. Lenzini, P. Ebben, and S. M. Iacob, "Context sensitive adaptive authentication," in *Proceedings of the European Conference on Smart Sensing and Context*, pp. 93–109, Springer, Kendal, UK, October 2007.

[36] RSA, "RSA adaptive authentication system," 2017, https://www.rsa.com/content/dam/rsa/PDF/h9096-rsa-risk-engine-sb-11-2.pdf.

[37] A. K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross, "Biometrics: a grand challenge," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 935–942, IEEE, Cambridge, UK, August 2004.

[38] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[39] A. Ross and A. Jain, "Information fusion in biometrics," *Pattern recognition letters*, vol. 24, no. 13, pp. 2115–2125, 2003.

[40] C. Braz and J.-M. Robert, "Security and usability: the case of the user authentication methods," in *Proceedings of the 18th Conference on l'Interaction Homme-Machine*, pp. 199–203, ACM, New York, NY, USA, 2006.

[41] K.-P. Yee, "User interaction design for secure systems," in *Proceedings of the 4th International Conference on Information and Communications Security*, pp. 278–290, Singapore, December 2002.

[42] Microsoft, "Key principles of software architecture," 2018, https://msdn.microsoft.com/en-us/library/ee658124.aspx.

[43] J. Brooke, *SUS-A Quick and Dirty Usability Scale: Usability Evaluation in Industry*, Taylor and Francis, Oxford, UK, 1996.

[44] Usability, "System usability scale (sus)," 2017, https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[45] M. F. Theofanos, R. J. Micheals, and B. C. Stanton, "Biometrics systems include users," *IEEE Systems Journal*, vol. 3, no. 4, pp. 461–468, 2009.

[46] ISO, *Human-Centred Design Processes for Interactive Systems*, International Organization for Standardization, Geneva, Switzerland, 1999.

[47] M. A. Sasse, "Red-eye blink, bendy shuffle, and the yuck factor: a user experience of biometric airport systems," *IEEE Security & Privacy*, vol. 5, no. 3, pp. 78–81, 2007.

[48] M. Harbach, E. Von Zezschwitz, A. Fichtner, A. De Luca, and M. Smith, "It's a hard lock life: a field study of smartphone (un) locking behavior and risk perception," in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, pp. 9–11, Menlo Park, CA, USA, July 2014.

[49] A. De Luca, A. Hang, E. Von Zezschwitz, and H. Hussmann, "I feel like I'm taking selfies all day!: towards understanding biometric authentication on smartphones," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1411–1414, ACM, New York, NY, USA, 2015.

[50] T. Sloane, "Behavioral biometrics: the restructuring of the authentication landscape," 2017, https://www.mercatoradvisorygroup.com/Webinars/Behavioral_Biometrics__The_Restructuring_of_the_Authentication_Landscape/.

[51] M. Winnick, "Putting a finger on our phone obsession," 2016, https://blog.dscout.com/mobile-touches.

[52] N. Cowan, C. C. Morey, Z. Chen, A. L. Gilchrist, and J. S. Saults, "Theory and measurement of working memory capacity limits," *Psychology of Learning and Motivation*, vol. 49, pp. 49–104, 2008.

[53] J. Zhang, X. Luo, S. Akkaladevi, and J. Ziegelmayer, "Improving multiple-password recall: an empirical study," *European Journal of Information Systems*, vol. 18, no. 2, pp. 165–176, 2009.

[54] A. Adams and M. A. Sasse, "Users are not the enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 40–46, 1999.

[55] M. Awad, Z. Al-Qudah, S. Idwan, and A. H. Jallad, "Password security: password behavior analysis at a small university," in *Proceedings of the 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, pp. 1–4, IEEE, Ras Al Khaimah, UAE, December 2016.

[56] S. Komanduri, R. Shay, P. G. Kelley et al., "Of passwords and people: measuring the effect of password-composition policies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2595–2604, ACM, New York, NY, USA, 2011.

[57] J. Bonneau, S. Preibusch, and R. J. Anderson, "A birthday present every eleven wallets? The security of customer-chosen banking pins," in *Proceedings of the 16th International Conference Financial Cryptography and Data Security*, vol. 7397, pp. 25–40, Springer, Kralendijk, Bonaire, February-March 2012.

[58] D. Silver, S. Jana, D. Boneh, E. Y. Chen, and C. Jackson, "Password managers: attacks and defenses," in *Proceedings of the 23rd USENIX Security Symposium*, pp. 449–464, San Diego, CA, USA, August 2014.

[59] T. Bhattasali, K. Saeed, N. Chaki, and R. Chaki, "A survey of security and privacy issues for biometrics based remote authentication in cloud," in *Proceedings of the IFIP International Conference on Computer Information Systems and Industrial Management*, pp. 112–121, Springer, Ho Chi Minh City, Vietnam, November 2014.

[60] PandaSecurities, "No password? You're asking to be hacked," 2016, https://www.pandasecurity.com/mediacenter/tips/smartphone-risk-dont-use-password.

[61] C. Katsini, M. Belk, C. Fidas, N. Avouris, and G. Samaras, "Security and usability in knowledge-based user authentication: a review," in *Proceedings of the 20th Pan-Hellenic Conference on Informatics*, p. 63, ACM, Patras, Greece, November 2016.

[62] J. K. Thorpe, *On the Predictability and Security of User Choice in Passwords*, Carleton University, Ottawa, ON, Canada, 2008.

[63] Secure Group, "Lock pattern, pin, or password: What is the most reliable way to lock a phone," 2017, https://blog.securegroup.com/lock-pattern-pin-or-password-what-is-the-most-reliable-way-to-lock-a-phone.

[64] G. Ye, Z. Tang, D. Fang et al., "Cracking android pattern lock in five attempts," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA, February-March 2017.

[65] F. Tari, A. Ozok, and S. H. Holden, "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords," in *Proceedings of the Second Symposium on Usable Privacy and Security*, pp. 56–66, ACM, Pittsburgh, PA, USA, July 2006.

[66] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, *Smudge Attacks on Smartphone Touch Screens*, Vol. 10, Woot, Carrollton, TX, USA, 2010.

[67] X. Suo, Y. Zhu, and G. S. Owen, "Graphical passwords: a survey," in *Proceedings of the 21st Annual Computer Security Applications Conference*, p. 10, IEEE, Tucson, AZ, USA, December 2005.

[68] CAPEC-Release1.6, "Common attack pattern enumeration and classification," 2016, http://capec.mitre.org.

[69] M. Mehrnezhad, E. Toreini, S. F. Shahandashti, and F. Hao, "Stealing pins via mobile sensors: actual risk versus user perception," *International Journal of Information Security*, pp. 1–23, 2016.

[70] A. Sarkisyan, R. Debbiny, and A. Nahapetian, "Wristsnoop: smartphone pins prediction using smartwatch motion sensors," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, Abu Dhabi, UAE, December 2015.

[71] F. Breitinger and C. Nickel, "User survey on phone security and usage," in *Proceedings of the Biometrics and Electronic Signatures (BIOSIG)*, pp. 139–144, Darmstadt, Germany, September 2010.

[72] M. Meeker and L. Wu, "Kleiner Perkins Caufield and Byers (KPCB): internet trends," 2017, http://www.kpcb.com/internet-trends.

[73] H. Choi, H. Kwon, and J. Hur, "A secure OTP algorithm using a smartphone application," in *Proceedings of the Seventh International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 476–481, IEEE, Sapporo, Japan, July 2015.

[74] H. Sun, K. Sun, Y. Wang, and J. Jing, "TrustOTP: transforming smartphones into secure one-time password tokens," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 976–988, ACM, Denver, CO, USA, October 2015.

[75] B. Cha, N. Kim, and J. Kim, "Prototype analysis of OTP key-generation based on mobile device using voice characteristics," in *Proceedings of the International Conference on Information Science and Applications (ICISA)*, pp. 1–5, IEEE, Jeju, Korea, April 2011.

[76] Verizon, "How long since you took a hard look at your cybersecurity?," 2017, http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017.

[77] I. Agadakos, C.-Y. Chen, M. Campanelli et al., *Jumping the Air Gap: Modeling Cyber-Physical a Ack Paths in the Internet-of-Things*, ACM, New York, NY, USA, 2017.

[78] M. Fomichev, F. Alvarez, D. Steinmetzer, P. Gardner-Stephen, and M. Hollick, "Survey and systematization of secure device pairing," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 517–550, 2018.

[79] M. Belk, P. Germanakos, C. Fidas, and G. Samaras, "A personalization method based on human factors for improving usability of user authentication tasks," in *Proceedings of International Conference on User Modeling, Adaptation, and Personalization*, pp. 13–24, Springer, Aalborg, Denmark, July 2014.

[80] M.-C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418, 2014.

[81] K. Krol, E. Philippou, E. De Cristofaro, and M. A. Sasse, ""They brought in the horrible key ring thing!" Analysing the usability of two-factor authentication in UK online banking," arXiv preprint arXiv:1501.04434, 2015.

[82] T. Zink and M. Waldvogel, "X. 509 user certificate-based two-factor authentication for web applications," in *Proceedings of the 10. DFN-Forum Kommunikationstechnologien*, pp. 51–61, Berlin, Germany, May 2017.

[83] K. J. Heffernan, F. Vetere, L. M. Britton, B. Semaan, and T. Schiphorst, "Insertable digital devices: voluntarily under the skin," in *Proceedings of the 2016 ACM Conference Companion Publication on Designing Interactive Systems*, pp. 85–88, ACM, Brisbane, QLD, Australia, June 2016.

[84] P. Strohmeier, C. Honnet, and S. Von Cyborg, "Developing an ecosystem for interactive electronic implants," in *Proceedings of the Conference on Biomimetic and Biohybrid Systems*, pp. 518–525, Springer, Edinburgh, UK, July 2016.

[85] M. Janiak, C. Schaub, D. Lynam, B. Howe, G. Wachter, and G. Krueger, "Biometric authentication device for use with a personal digital assistant," US Patent App. 09/854,078, 2001.

[86] K. J. Heffernan, "Insertables workshop," 2016, https://insertables.wordpress.com/.

[87] TechTarget, "Bespoke," 2017, http://whatis.techtarget.com/definition/bespoke.

[88] M. Gupta, C. Holloway, B. M. Heravi, and S. Hailes, "A comparison between smartphone sensors and bespoke sensor devices for wheelchair accessibility studies," in *Proceedings of the IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 1–6, IEEE, Singapore, April 2015.

[89] First4magnets, "Use of neodymium magnets," 2016, https://www.first4magnets.com/tech-centre-i61/information-and-articles-i70/neodymium-magnet-information-i82/common-applications-of-neodymium-magnets-i88.

[90] P. Urien and S. Piramuthu, "Framework and authentication protocols for smartphone, NFC, and RFID in retail transactions," in *Proceedings of the IEEE 8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 77–82, IEEE, Melbourne, VIC, Australia, April 2013.

[91] I. J. Forster and A. N. Farr, "Method for preventing unauthorized diversion of nfc tags," US Patent App. 15/659,941, 2017.

[92] H. Hassan, S. Wacquant, and H. B. Seifert, "Vehicle driver monitoring system," US Patent App. 15/463,293, 2017.

[93] M. Cuff, "Smart digital tattoos," 2014, http://www.stylus.com/scckpj.

[94] K. J. Heffernan, F. Vetere, and S. Chang, "Towards insertables: devices inside the human body," *First Monday*, vol. 22, no. 3, 2017.

[95] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, *Guide To Biometrics*, Springer Science & Business Media, Berlin, Germany, 2013.

[96] R. Ramachandra and C. Busch, "Presentation attack detection methods for face recognition systems: a comprehensive survey," *ACM Computing Surveys*, vol. 50, no. 1, pp. 1–37, 2017.

[97] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: accomplishments, challenges, and opportunities," *Pattern Recognition Letters*, vol. 79, pp. 80–105, 2016.

[98] A. Hadid, N. Evans, S. Marcel, and J. Fierrez, "Biometrics systems under spoofing attack: an evaluation methodology and lessons learned," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 20–30, 2015.

[99] S. Kovach, "Business insider-Samsung's Galaxy S8 facial recognition feature can be fooled with a photo," 2017, http://www.businessinsider.com/samsung-galaxy-s8-facial-recognition-tricked-with-a-photo-2017-3?IR=T.

[100] J. Titcomb, "Hackers claim to beat iPhone X's face id in one week with 115 mask," 2017, http://www.telegraph.co.uk/technology/2017/11/13/hackers-beat-iphone-xs-face-one-week-115-mask/.

[101] H. Saevanee, N. Clarke, S. Furnell, and V. Biscione, "Continuous user authentication using multi-modal biometrics," *Computers & Security*, vol. 53, pp. 234–246, 2015.

[102] A. Hern, "The guardian-Samsung Galaxy S8 iris scanner fooled by German hackers," 2017, https://www.theguardian.com/technology/2017/may/23/samsung-galaxy-s8-iris-scanner-german-hackers-biometric-security.

[103] A. Charles, "The guardian-iPhone 5S fingerprint sensor hacked by Germany's Chaos Computer Club," 2013, https://www.theguardian.com/technology/2013/sep/22/apple-iphone-fingerprint-scanner-hacked.

[104] C. McGoogan and D. Demetriou, "The telegraph-peace sign selfies could let hackers copy your fingerprints," 2017, http://www.telegraph.co.uk/technology/2017/01/12/peace-sign-selfies-could-let-hackers-copy-fingerprints.

[105] C. Bhagavatula, B. Ur, K. Iacovino, S. M. Kywe, L. F. Cranor, and M. Savvides, "Biometric authentication on iPhone and Android: usability, perceptions, and influences on adoption," in *Proceedings of the Workshop on Usable Security (USEC)*, pp. 1-2, San Diego, CA, USA, January 2015.

[106] A. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*, Vol. 479, Springer Science & Business Media, Berlin, Germany, 2006.

[107] M. Kumar, A. Insan, N. Stoll, K. Thurow, and R. Stoll, "Stochastic fuzzy modeling for ear imaging based child identification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1265–1278, 2016.

[108] M. Sultana, M. Gavrilova, and S. Yanushkevich, "Multi-resolution fusion of DTCWT and DCT for shift invariant face recognition," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 80–85, IEEE, Hong Kong, China, October 2014.

[109] Y. Xu, X. Fang, X. Li et al., "Data uncertainty in face recognition," *IEEE transactions on cybernetics*, vol. 44, no. 10, pp. 1950–1961, 2014.

[110] C. Song, A. Wang, K. Ren, and W. Xu, "EyeVeri: a secure and usable approach for smartphone user authentication," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*, pp. 1–9, IEEE, San Francisco, CA, USA, April 2016.

[111] IBIA, "Behavioral biometrics," 2017, https://www.ibia.org/biometrics-and-identity/biometric-technologies/behavioral-biometrics.

[112] L. M. Mayron, "Behavioral biometrics for universal access and authentication," in *Proceedings of the International Conference on Universal Access in Human-Computer Interaction*, pp. 330–339, Springer, Los Angeles, CA, USA, August 2015.

[113] A. Buriro, B. Crispo, F. Del Frari, J. Klardie, and K. Wrona, "ITSME: multi-modal and unobtrusive behavioural user authentication for smartphones," in *Proceedings of the International Conference on Passwords*, pp. 45–61, Springer, Cambridge, UK, December 2015.

[114] A. Buriro, B. Crispo, F. Del Frari, and K. Wrona, "Touchstroke: smartphone user authentication based on touch-typing biometrics," in *Proceedings of the International Conference on Image Analysis and Processing*, pp. 27–34, Springer, Genoa, Italy, September 2015.

[115] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: user verification on smartphones via tapping behaviors," in *Proceedings of the IEEE 22nd International Conference on Network Protocols (ICNP)*, pp. 221–232, IEEE, Raleigh, NC, USA, October 2014.

[116] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: from authentication to imitation," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3209–3221, 2017.

[117] M. R. Hestbek, C. Nickel, and C. Busch, "Biometric gait recognition for mobile devices using wavelet transform and support vector machines," in *Proceedings of the 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 205–210, IEEE, Vienna, Austria, April 2012.

[118] A. Muro-De-La-Herran, B. Garcia-Zapirain, and A. Mendez-Zorrilla, "Gait analysis methods: an overview of wearable and non-wearable systems, highlighting clinical applications," *Sensors*, vol. 14, no. 12, pp. 3362–3394, 2014.

[119] M. Sultana, P. P. Paul, and M. L. Gavrilova, "Social behavioral information fusion in multimodal biometrics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[120] H. Khan, U. Hengartner, and D. Vogel, "Usability and security perceptions of implicit authentication: convenient, secure, sometimes annoying," in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS 2015)*, pp. 225–239, Santa Clara, CA, USA, July 2015.

[121] N. Poh and J. Korczak, "Hybrid biometric person authentication using face and voice features," in *Proceedings of the 3rd International Conference Audio- and Video-Based Biometric Person Authentication (AVBPA)*, vol. 1, pp. 348–353, Springer, Halmstad, Sweden, June 2001.

[122] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 955–966, 1995.

[123] K. Brunet, K. Taam, E. Cherrier, N. Faye, and C. Rosenberger, "Speaker recognition for mobile user authentication: an Android solution," in *8ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d'Information (SAR SSI)*, p. 10, Mont-de-Marsan, France, September 2013.

[124] K. Murao, H. Tobise, T. Terada, T. Iso, M. Tsukamoto, and T. Horikoshi, "Mobile phone user authentication with grip gestures using pressure sensors," *International Journal of Pervasive Computing and Communications*, vol. 11, no. 3, pp. 288–301, 2015.

[125] J. Kittler, J. Matas, K. Jonsson, and M. R. Sánchez, "Combining evidence in personal identity verification systems," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 845–852, 1997.

[126] H. Saevanee, N. Clarke, and S. Furnell, "Multi-modal behavioural biometric authentication for mobile devices," in *Information Security and Privacy Research*, pp. 465–474, Springer, Heidelberg, Germany, 2012.

[127] Y. Tang, N. Hidenori, and Y. Urano, "User authentication on smart phones using a data mining method," in *Proceedings of the International Conference on Information Society (i-Society)*, pp. 173–178, IEEE, London, UK, June 2010.

[128] S. M. Welten, *Sensing with Smartphones*, 2013.

[129] Y. Obuchi, "PDA speech database," 2006, http://www.speech.cs.cmu.edu/databases/pda/index.html.

[130] S. Eberz, K. B. Rasmussen, V. Lenders, and I. Martinovic, "Evaluating behavioral biometrics for continuous authentication: challenges and metrics," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 386–399, ACM, Abu Dhabi, UAE, April 2017.

[131] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa, "Active user authentication for smartphones: a challenge data set and benchmark results," in *Proceedings of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS 2016)*, pp. 1–8, IEEE, Buffalo, NY, USA, September 2016.

[132] A. Stolerman, A. Fridman, R. Greenstadt, P. Brennan, and P. Juola, "Active linguistic authentication revisited: real-time stylometric evaluation towards multi-modal decision fusion," in *Proceedings of the Tenth Annual IFIP WG 11.9 International Conference on Digital Forensics*, vol. 11, pp. 1–11, Vienna, Austria, January 2014.

[133] R. P. Guidorizzi, "Security: active authentication," *IT Professional*, vol. 15, no. 4, pp. 4–7, 2013.

[134] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, "Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location," *IEEE Systems Journal*, vol. 11, no. 2, pp. 513–521, 2017.

[135] M. L. Brocardo, I. Traore, and I. Woungang, "Toward a framework for continuous authentication using stylometry," in *Proceedings of the IEEE 28th International Conference on Advanced Information Networking and Applications (AINA 2014)*, pp. 106–115, IEEE, Victoria, Canada, May 2014.

[136] ClearLogin, "Risk-based authentication," 2017, http://www.clearlogin.com/glossary/risk-based-authentication/.

[137] B. Schneier, "Risk-based authentication," 2013, https://www.schneier.com/blog/archives/2013/11/risk-based_auth.html.

[138] Identity Automation, "Risk-based authentication," 2017, https://www.identityautomation.com/iam-platform/rapididentityidentity-access-management/multi-factor-authentication/risk-based-authentication/.

[139] D. Hintze, E. Koch, S. Scholz, and R. Mayrhofer, "Location-based risk assessment for mobile authentication," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pp. 85–88, ACM, Heidelberg, Germany, September 2016.

[140] Y. Y. Haimes, *Risk Modeling, Assessment, and Management*, John Wiley & Sons, Hoboken, NJ, USA, 2015.

[141] IdentityAutomation, "What is adaptive authentication?," 2017, http://blog.identityautomation.com/what-is-adaptive-authentication.

[142] A. Jain and A. Kumar, "Biometric recognition: an overview," in *Second Generation Biometrics: The Ethical, Legal and Social Context*, E. Mordini and D. Tzovaras, Eds., pp. 49–79, Springer, Dordrecht, Netherlands, 2012.

[143] P. P. Paul, M. L. Gavrilova, and R. Alhajj, "Decision fusion for multimodal biometrics using social network analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 11, pp. 1522–1533, 2014.