

PupilRec: Leveraging Pupil Morphology for Recommending on Smartphones

Xiangyu Shen^{ID}, Graduate Student Member, IEEE, Hongbo Jiang^{ID}, Senior Member, IEEE, Daibo Liu, Member, IEEE, Kehua Yang^{ID}, Feiyang Deng^{ID}, John C. S. Lui^{ID}, Fellow, IEEE, Jiangchuan Liu^{ID}, Fellow, IEEE, Schahram Dustdar^{ID}, Fellow, IEEE, and Jun Luo^{ID}, Senior Member, IEEE

Abstract—As mobile shopping has gradually become the mainstream shopping mode, recommendation systems are gaining an increasingly wide adoption. Existing recommendation systems are mainly based on explicit and implicit user behaviors. However, these user behaviors may not directly indicate users' inner feelings, causing erroneous user preference estimation and thus leading to inaccurate recommendations. Inspired by our key observation on the correlation between pupil size and users' inner feelings, we consider using the change of pupil size when browsing to model users' preferences, so as to achieve targeted recommendations. To this end, we propose PupilRec as a computer-vision-based recommendation framework involving a mobile terminal and a server side. On the mobile terminal, PupilRec collects users' pupil size change information through the front camera of smartphones; it then preprocesses the raw pupil size data before transmitting them to the server. On the server side, PupilRec utilizes the Tsfresh package and Random Forest algorithm to figure out the key time-series features directly implying user preferences. PupilRec then trains a neural network to fit a user preference model. Using this model, PupilRec predicts user preference to obtain a user-product matrix and further simplifies it by singular value decomposition. Finally, the real-time recommendation is achieved by a collaborative filtering module that retrieves recommended contents to users smartphones. We prototype PupilRec and conduct both experiments and field studies to comprehensively evaluate the effectiveness of PupilRec by recruiting 67 volunteers. The overall results show that PupilRec

can accurately estimate users' preference and can recommend products users interested in.

Index Terms—Energy optimization, multilayer perceptron (MLP), pupillary response, random forest, recommendation, user preference model.

I. INTRODUCTION

SHOPPING on mobile phones has been a trend nowadays [1], [2], in which *recommendation systems* [3] play a vital role. According to a survey of [4], during the Double Eleven festival in 2020, Tmall's order volume was 498.2 billion yuan and the total trading volume of JD was 271.5 billion yuan. During festivals, mobile shoppers may account for 69.31% of total consumers. Facing such a keen competition in the mobile shopping market, the effectiveness of the recommendation system predicting user preference for a shopping app has become a key factor to outbid others in revenue. In addition, with the rise of short video platforms, the demand for accurate recommendations becomes urgent. For example, in the largest Chinese short video platform Douyin, recommendation systems directly affect the traffic of the platform as well as the revenue of live-streaming sales [5].

Unfortunately, traditional recommendation systems mostly achieve their goals by simply following users' online behaviors. This seemingly plausible method often causes users to get recommendation about what they have just browsed and/or purchased, which can be highly inaccurate and in turn lead to user dissatisfaction. Consequently, it is imperative for a mobile shopping app to come up with effective recommendation schemes.

Specifically, existing recommendation systems mainly exploit explicit and implicit activities, such as Web browsing, commodity purchases, content-clicks, ratings, and comments [6], to model *user preference*. However, these complex behavioral activities may not be directly relevant to users' inner feelings: for example, users may click a product after being attracted by its title or Web picture, but may not be satisfied with its content after accessing the webpage; this potentially raises model uncertainty and brings about erroneous prediction [7]. In general, it is extremely challenging to build an accurate user preference model simply based on user activities. This can be attributed to the fact that people's inner feelings are usually not directly expressed through external behaviors, let alone online activities. In accordance with recent report of He *et al.* [8], due to the inadequacy of

Manuscript received 18 February 2022; revised 25 May 2022; accepted 3 June 2022. Date of publication 9 June 2022; date of current version 24 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant U20A20181, Grant 61732017, and Grant 61902060; and in part by the National Social Science Foundation of China under Grant 19ZDA103. This article was presented in part at IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 7–10 July 2021. [DOI: 10.1109/ICDCS51616.2021.00102]. (*Corresponding authors: Hongbo Jiang; Daibo Liu; Kehua Yang.*)

Xiangyu Shen, Hongbo Jiang, Daibo Liu, and Kehua Yang are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, Hunan, China (e-mail: shenxiangyu@hnu.edu.cn; hongbojiang@hnu.edu.cn; dblu@hnu.edu.cn; khyang@hnu.edu.cn).

Feiyang Deng is with the Chow Yei Ching School of Graduate Studies, The City University of Hong Kong, Hong Kong (e-mail: feiyadeng2-c@my.cityu.edu.hk).

John C. S. Lui is with the Computer Science and Engineering Department, The Chinese University of Hong Kong, Hong Kong (e-mail: csliu@cse.cuhk.edu.hk).

Jiangchuan Liu is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: jcliu@sfu.ca).

Schahram Dustdar is with the Distributed Systems Group, Vienna University of Technology, 1040 Vienna, Austria (e-mail: dustdar@infosys.tuwien.ac.at).

Jun Luo is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: junluo@ntu.edu.sg).

Digital Object Identifier 10.1109/JIOT.2022.3181607

2327-4662 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information.

the user preference model, the effectiveness of content delivery, though widely adopted by mobile shopping platforms and content pushing systems, is still far from satisfactory. Therefore, further research is necessary to break the limitations of indirect inferences from explicit and implicit behavioral activities by deeply mining more accurate and reliable user-related information that directly reflects users' inner feelings, and then better modeling users' preferences.

Given this situation, we raise a question: *can users' inner feelings be exploited to model user preference, thus to improve recommendation performance?* According to recent research [9], pupil is a human mind's window under control of the automatic nervous system. The change in pupil size when viewing particular content is closely correlated with users' mind [10]. In other words, physiological processes of pupillary response can potentially be employed to indicate the user's degree of interest in the viewed content. In addition, with the rapidly advancing mobile Internet technology and widespread use of smartphones with built-in front-facing camera and enhanced computing capability, people are now increasingly turning to mobile shopping, education, and entertainment. Consequently, it has become feasible to capture the user's pupillary response in real time on the mobile phones, while subjecting to the privacy protection policy.

In our study, we specifically investigate correlation between user preference and physiological process of pupillary response captured by built-in front-facing camera of mobile phones. On this basis, we do a further job of expressing user expectations more comprehensively and accurately, and then combine it with the traditional recommendation algorithm to achieve better recommendation. Achieving this goal requires several key technical challenges. First, the physiological process of pupillary response is intricate and complex: it is possible to extract some features from this process, but is difficult to identify the key features completely relevant to users' preferences. In addition, the diversity of pupillary responses caused by inherent individual differences imposes another challenge to the generality of a one-fit-all model. Therefore, users' preference model is required to be customized online.

Last but not least, mobile scenarios pose certain specific challenges, such as the changes in light intensity may heavily affect the quality of face images recorded by front-facing camera. Also, as PupilRec is implemented on a smartphone, energy consumption directly affects user experience, demanding a careful energy optimization.

Aiming to address these challenges, we hereby propose PupilRec as the first mobile recommendation system exploiting pupillary response (i.e., the change of pupil size in the time domain). As shown in Fig. 1, PupilRec leverages the relationship between pupillary responses and user's impression to the viewed contents in order to make meaningful recommendations. First, we perform extensive research to uncover the generic physiological process that underlies the pupillary response when watching certain content on mobile phones. This enables us to identify six key time-series features associated with the degree of user preference by using the Random Forest. In addition, to deal with pupillary responses' diversity due to intrinsic personal differences, we use a

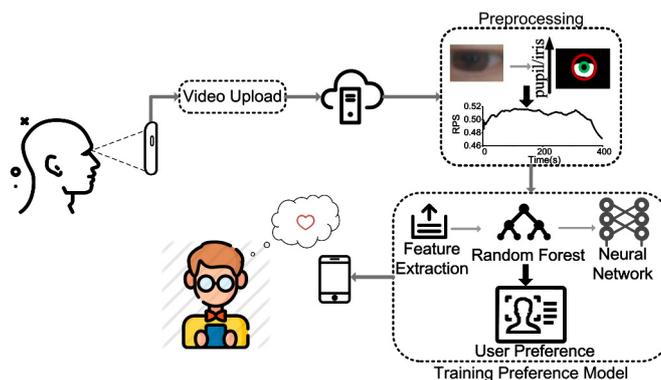


Fig. 1. PupilRec: a novel recommendation system observing your eyes, rather than your behaviors.

deep-learning-based approach to automatically train and adjust the importance of key features for each user; the generated personalized user preference model can then be correlated with the pupillary response of the user. Finally, to handle the mobile-specific challenges, we adopt a light-related model to achieve environment independence, and we also propose an optimization module for energy conservation in PupilRec. We prototype PupilRec on mobile phones and conduct extensive experiments to evaluate its performance in user preference identification and recommendation.

Experimental results show that PupilRec is able to accurately identify users' preferences and recommend products of interest to them. To conclude, our main contributions are as follows.

- 1) We conduct an insightful investigation of pupillary response when watching contents of different interest degrees. As far as we know, this is the first work exploring the quantitative correlation between users' pupillary responses and preferences. Meanwhile, it is also the first attempt to apply this relationship to mobile recommendation systems.
- 2) By using Random Forest to mine the general physiological process of pupillary response, a set of key time-series features is identified to characterize the user-preference-related pupillary response.
- 3) We suggest using neural networks to customize our user preference models to train and adjust the weight of key features individually.
- 4) We construct the mobile recommendation system PupilRec by combining the pupillary-response-based user preference model with the collaborative filtering (CF) algorithm widely used in traditional recommendation systems. In addition, we adopt a singular value decomposition (SVD) algorithm to improve the recommending efficiency.
- 5) To suppress the energy consumption on mobile phones, PupilRec conducts data collection and preprocessing on mobile clients, but it offloads computationally intensive module of preference model training to a server. PupilRec is also equipped with an energy optimization module to govern the operation time of the front-facing camera.

- 6) We tackle the influence of light intensity variation resorting to the relationship amongst pupil diameter, mental workload, and lighting conditions, so as to fully model the light intensity impact.
- 7) We verify the efficacy of PupilRec through extensively conducted experiments by recruiting 67 volunteers in total.¹ The results demonstrate that PupilRec achieves up to 83.58% accuracy on average and outperforms traditional recommendation systems relying on user active scoring.

The remaining part of this article is organized as follows. In Section II, we first review the related works of PupilRec. Then, we explore the relationship between pupil diameter and user preference in Section III. We describe PupilRec's technical details in Section IV, then we introduce the energy optimization module in Section V. The performance evaluations of PupilRec are reported in Section VI. Having discussed the limitations of this works in Section VII, we conclude with a summary of this work in Section VIII.

II. RELATED WORK

In this section, we review the efforts of researchers in modeling user preference and provide a comprehensive overview of PupilRec's advantages compared to the state-of-the-art works.

A. Diverse Preference Modeling

In recent years, researchers have paid more attention to modeling the varying preferences toward different items and proposed several methods. We roughly categorize those methods into two groups.

Methods in the first group exploit reviews to analyze each user's attention on different aspects of the target item and then integrate the attention weights into the matrix factorization for recommendation [11]–[14]. In particular, ALFM [13] applies a topic model on reviews to detect the user attention, and A³NCF [12] uses a neural attention network to learn user attention from reviews. Following the same idea, Chin *et al.* [14] have developed an end-to-end attentive neural network-based recommendation model, leveraging reviews and ratings to learn users' diverse preferences on different aspects of items. The AFM method [15] adopts a similar strategy and exploits other types of side information (e.g., item category) instead of reviews.

As to the methods in the second group, they model user preferences by analyzing the user's inherent attributes or external connections. For example, YouTubeNet [16] discretizes and splices users' demographic information, including gender, age, region, education, etc., and uses them the original input of the neural network. Given a social network, Mohsen [17] proposes to leverage a user's neighbor attributes to infer the user's preference. Different from [16] and [17], DREAM [18] relies on RNN to model users' preferences by portraying their changes over time.

¹Our study was IRB approved by our university. It does not raise any ethical issues.

All of the above methods are on the basis of users' properties or online activities. In this work, we use a different strategy to build a new model of user preference according to key features of pupillary responses, which breaks the limitations of indirectly inferring user preferences from explicit and implicit behavioral activities. To the best of our knowledge, this is the first work to probe relationship between users' pupillary responses and their preferences in quantitative terms.

B. Connecting Pupil With Human Psychology

In prior investigations of pupillary response, many researchers have probed the relevance of this physiological response to user's psychology. Pflieger *et al.* [19] explored the correlation between users' mental workload and their pupillary responses, requiring users to complete tasks of various difficulty levels. Foroughi *et al.* [20] tentatively proved pupillary response could be used to capture within-task learning changes.

None of these two proposals consider external factors. However, different kinds of interference may exist in an actual scenario, like abrupt changes in lighting conditions and mood status, etc. Wang *et al.* [21] investigated pupillary response as a cognitive workload under the influence of changing luminance condition and emotional arousal. The above works explored the pupillary response when completing static tasks. In addition, there are also works involving dynamic tasks, for example, Jiang *et al.* [22] studied the pupillary response during aiming in a teleoperation setting.

In this article, we consider the relationship between pupillary response and users' psychology from a distinct perspective. Specifically, we explore how changes in pupil diameter reflect users' preferences in mobile conditions, hence supplementing and promoting nowadays mobile recommendation systems.

III. EMPIRICAL STUDY

In this section, we start by exploring the feasibility of pupillary response to effectively indicate user preferences, and conduct extensive empirical studies in Section III-A; this helps motivating our study. By mining the change processes of pupillary response over time, we further use Random Forest to weigh the importance of a large amount of time-series features in Section III-B. These importance measurements finally enable us to identify a group of key features, revealing the possibility of modeling user preference by pupillary response.

A. Correlating Pupillary Response and Preference

Researches reported that people's pupils are prone to unconsciously 1) dilate when watching something interesting, aiming to acquire more information and 2) not dilate or even shrink when viewing unattractive contents, tending to impede the access to information [23], [24]. This physiological phenomenon prompts us to completely investigate the relationship between pupillary response and user preference.

In particular, we use *relative pupil size* (RPS) as a metric of pupillary response, where RPS denotes the pupil to iris ratio. The reason is that the human eye is at full size by the time the

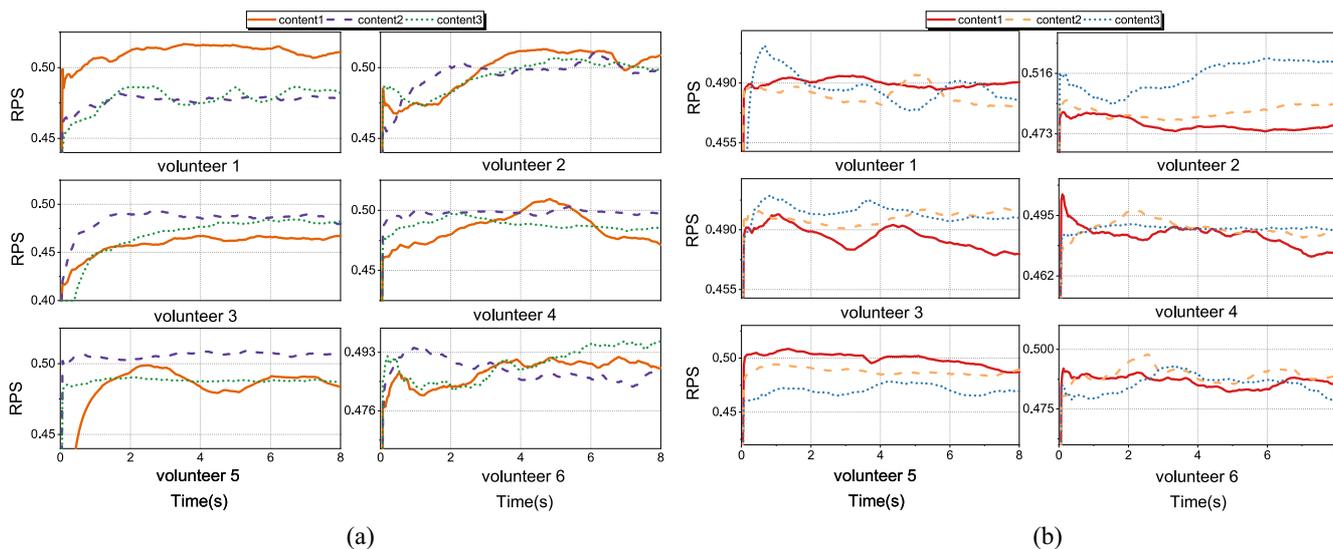


Fig. 2. Randomly selected six volunteers' RPS sequences in time domain while, respectively, viewing (a) interested and (b) uninterested contents.

person is 13 years old [25]. Therefore, we can consider the iris diameter as a constant and thus use it as a reference in a frame of recorded videos [25]. As a result, our RPS is invariant no matter what distance and angle the video is recorded at.

To illustrate how the pupil response is associated with user preferences, 30 volunteers are recruited to measure their RPS changes when viewing various kinds of contents like pictures and short videos on scenery, food, celebrity, goods, and so on. We let volunteers label each piece of content based on their respective degrees of preference for it. In this study, we utilize front-facing camera on mobile phones to capture volunteers' RPS in the time domain when viewing the contents. To ensure the reliable conduct of investigations and evidence-based tests, the lighting intensity is kept relatively stable between 220 and 260lux, and the distances between volunteers and their mobile phones remain largely constant at 30 cm.

For the sake of simplicity and with no generalization lost, six volunteers are randomly selected (three males and three females, aged from 20 to 29) and the representational process of pupillary response when viewing the content of varying interest/disinterest is plotted in Fig. 2. As illustrated in the figure, the RPS time-domain variation of the selected six volunteers viewing interested and uninterested contents are, respectively, plotted in Fig. 2(a) and (b). It can be clearly observed in Fig. 2 that RPS keeps changing over time when volunteers watch diverse content, regardless of whether they are watching the content of interest or disinterest. However, the changing patterns are evidently different when viewing interested contents and uninterested contents, if one compares Fig. 2(a) with (b). In particular, the overall "increasing" trend of RPS demonstrated in Fig. 2 is much more conspicuous. Intuitively, this phenomenon can be attributed to the fact that people intentionally dilate their pupils to obtain more information about the interested content. In contrast, pupils may shrink to protect against uninterested (or irrelevant) information entering their eyes (hence their mind).

Specifically, for each volunteer in Fig. 2, the changing patterns of interested and uninterested RPS sequences are diverse.

For instance, the overall trend of volunteer1's RPS sequence is increasing or fluctuating within a small range of 0.01 when viewing interested contents, while when viewing uninterested contents, the general trend is decreasing and the sequences fluctuations are relatively larger within a range of around 0.03. Another example is volunteer2, the RPS sequence trend of viewing interested contents is increasing as well, however, when watching uninterested contents, the RPS sequences show two forms of changing patterns: one kind of pattern is that the RPS sequences are constantly decreasing, such as the RPS sequences of content1 and content2. The other pattern is that the RPS sequence initially shows a decreasing trend and then slowly increases, such as the RPS sequence of content3. The rest of the volunteers' RPS sequences also present different patterns of change when viewing the interested and uninterested contents according to Fig. 2.

Unfortunately, even for the same volunteer, individual RPS changing processes can vary drastically, such as volunteer2 and volunteer5. Moreover, due to inherent individual differences, these change processes are poorly consistent across individuals. Therefore, it is difficult to straightforwardly determine a user's interest in what he/she is viewing based on a specific pattern of RPS change process. In order to precisely relate pupillary responses to user preferences, we must further clarify the key features of pupillary responses which are closely related to user preferences.

B. Key Features Relevant to User Preference

On the basis of the analysis in Section III-A, the changing process of RPS is considered as a time-domain sequential response, and a large number of time-series features which might be related to user preferences are extracted. Through segmentation and preprocessing of pupillary responses produced by watching diverse content, we extract features from the time series using a widely adopted Python module—Tsfresh [26]. The extracted features can be usable for describing or clustering time series. Furthermore, they also can be

TABLE I
DESCRIPTION ON PART OF THE USEFUL FEATURES

Feature name	Descriptions	Feature name	Descriptions
Front Trend	slope of the beginning part of RPS curve	E	absolute energy
Back Trend	slope of the ending part of RPS curve	Ar	autoregressive coefficient
Mean	global average	CWT	Ricker wavelet analysis
Variance	variance of middle part	FFT	Fourier transform coefficient
CE	time series complexity	Kurt	kurtosis
SampEn	sample entropy of middle part	Skewness	skewness

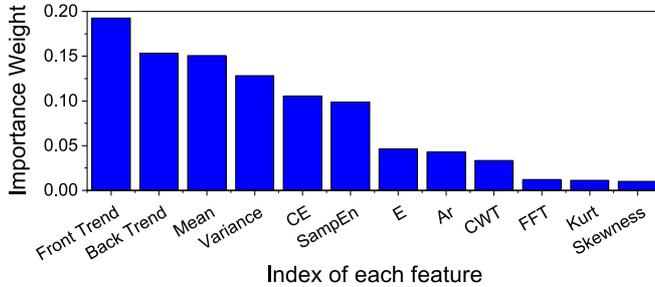


Fig. 3. Weighted importance of features by Random Forest.

utilized to build models that perform classifying or regression tasks on time series. In general, these features offer an opportunity to gain new insights into the time series and their attributes of dynamics.

To find the key features that effectively distinguish different RPS sequences, we employ the feature extraction tool of the Tsfresh module to extract and initially filter out potential useful features. We list the descriptions of some of the useful features as shown in Table I, for example, the front and back trend (Front Trend and Back Trend) of the pupillary response, the mean, the variance of RPS during the response process, and the sample entropy (*SampEn*), the complexity (*CE*) of this process, and so forth. Generally, in this empirical study, we investigate and measure over hundreds features of the time series.

In order to identify key features from the large number of features, we employ the ensemble learning—Random Forest [27] to measure the importance of each feature associated with user preference. The top 12 time-series features that have the maximum importance weight are plotted in Fig. 3. As shown in the figure, it is obvious that the first six features have significantly more weight than the other features. In general, the top six time-series features have over 82.9% of the total weight, with each feature exceeding 10%. In comparison, the remaining features each contribute less than 5% of the total weight. Therefore, we draw the conclusion that the top six time-series features are suitable for representing the inherent properties of pupillary responses that are closely related to user preferences. In particular, we briefly interpret the physical significance of the top six time-series features on the RPS changing process as follows.

- 1) *Mean Value of RPS Sequence*: We calculate the mean value of the RPS sequence during gaze. When viewing something attractive, people's pupils are prone to unconsciously dilate, aiming to acquire more information. Generally, for most volunteers in Fig. 2, the RPS of viewing the content of interest is larger than that of

disinterest. However, there is also the opposite situation caused by pupil oscillation [28]. For instance, for volunteer 3, the mean RPS for viewing interesting content is clearly smaller than that for viewing uninterested content.

- 2) *Slope*: Slope refers to the sudden change of pupillary response at the beginning or the end of gaze. Accordingly, we refer to the pupillary response's beginning slope and ending slope as Front Trend and Back Trend, respectively. To be specific, Front Trend and Back Trend, respectively, stand for the slopes of the first and last third of RPS sequence. As visualized in Fig. 2, the RPS sequences of interest increase steeply at the beginning and remain constant or decrease marginally at their end. Conversely, the RPS sequences in the case of viewing uninterested contents increase modestly at the start, then decrease, and end with a noticeable decrease, which is meant to impede the acquisition of information. Therefore, the slopes of RPS sequences can be used as a potentially valid indicator to distinguish between RPS sequences in the cases of interest and disinterest.
- 3) *Variance*: Variance represents the expected value of the deviation of the RPS series. When people are uninterested in what they are watching, their eyes wander around instead of fixating on the mobile phone's screen, resulting in a more rapid change in pupil diameter than in a gaze situation. The variance is derived with the middle third of the RPS sequence.
- 4) *Complexity Estimate (CE)*: Time-series complexity estimation (*CE*) serves to express the crests and troughs' complexity. A larger *CE* indicates more crests and troughs in the RPS sequence and vice versa. *CE* is derived by the middle third of RPS sequence.
- 5) *Sample Entropy (SampEn)*: Entropy is often adopted to represent the partial fluctuation of a time series [29]. Here, we use *SampEn* to represent the complexity of time series in terms of the probability of yielding a new pattern within it. The larger the *SampEn*, the greater the RPS sequence's complexity is. The middle third of the RPS sequence is used to compute the *SampEn*.

As shown above, we can model the relationship between pupillary responses and user preferences using the six key time-series features identified. Nevertheless, because of inherent individual differences, the RPS sequences vary from person to person, signifying that the importance weights of these key features are varied in people. This prevents us from building a one-fit-all model on all users. To address this issue, the importance of each user's key time-series features must be trained online individually so that the user's preference model will be fine-tuned correspondingly. In the following section, we elaborate on building a user preference model based on pupillary responses.

IV. PUPILREC DESIGN

In this section, we will present the details of PupilRec's designing. We start with an overview of PupilRec in Section IV-A. PupilRec uploads the videos recorded with

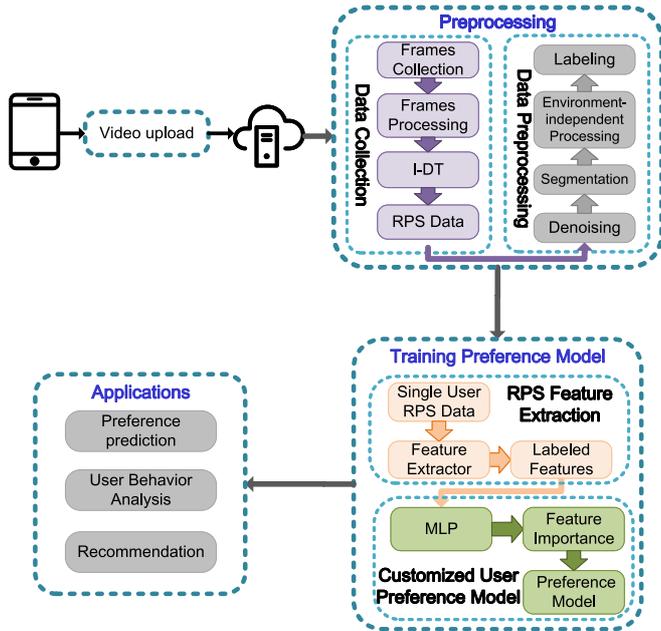


Fig. 4. System architecture of PupilRec.

smartphones to the server for preprocessing in Section IV-B, including data collection and data preprocessing, Then, the labeled RPS sequences are utilized for training preference model in Section IV-C. On that basis, key time-series features are extracted from RPS sequences in Section IV-C1, and multilayer perceptron (MLP) is adopted to automatically weight the importance of key features and then PupilRec introduces user preference model customization in Section IV-C2. Based on the customized preference model, PupilRec deploys SVD and CF to construct a mobile recommendation system in Section IV-D.

A. Design Overview

The system overview of PupilRec is shown in Fig. 4. PupilRec is composed of three modules: 1) *Preprocessing*; 2) *Training Preference Model*; and 3) *Applications*. The *Preprocessing* module includes two submodules, *Data Collection* and *Data Preprocessing*. In *Data Collection*, PupilRec uses a front-facing camera on mobile phone to collect information about the RPS of users while viewing content with different levels of interest on the phone screen. Then, PupilRec uploads the recorded videos to the server. On the server side, the dispersion-threshold identification (referred to as I-DT) method [30] is utilized to obtain RPS sequences within the gaze time for initial filtering. Building on this, in the *Data Preprocessing* module, the collected RPS sequences are normalized by, respectively, signal denoising, segmentation, environment-independent processing, and elaborate labeling.

In *Feature Extraction*, the feature extractor tool of Tsfresh package is adopted by PupilRec to extract key time-series features of pupillary response presented in Section III. In order to overcome the inherent diversity of feature weight due to individual differences, in the *Customized User Preference Model*, PupilRec employs the MLP to automatically train and assign

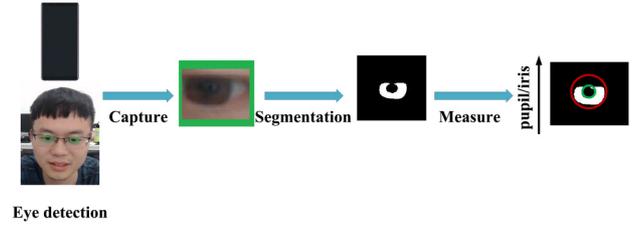


Fig. 5. General process of RPS data collection.

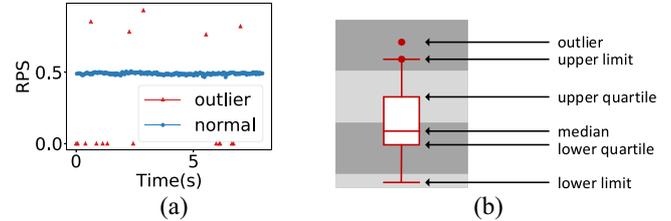


Fig. 6. (a) Abnormal values in RPS sequences and (b) box plot for outlier detection.

weights to the importance of each individual’s key features, and then yields a customized user preference model that is correlated with users’ pupillary response. Based on the personalized user preference model, PupilRec recommends to users by virtue of SVD and CF in *Application*.

B. Preprocessing

In this section, we introduce the procedures of collecting and preprocessing RPS sequences.

1) *Data Collection*: Fig. 5 depicts the general process of RPS data collection. When people browsing on mobile phones, PupilRec uses the front-facing camera and OpenCV’s Haar Cascade classifier [31] to capture RPS sequences. Note that the captured video relating to the pupil can be segmented into individual frames, each of which can be regarded as a picture containing pupil profiles. For each frame, a deep learning approach—U-net [32] is used to split the pupil and iris. Specifically, the U-net architecture allows the training of a deep learning network and the resulting model efficiently covers the eye region into a segmented image. Finally, the RPS value is derived by an algorithm that fits circles around the iris and pupil. The algorithm finds the circle with the smallest area of the iris and pupil, enclosing a 2-D point set.

2) *Data Preprocessing*: After capturing the RPS sequences, PupilRec further performs data preprocessing, consisting of denoising, segmentation, environment-independent processing, and the labeling of RPS sequences.

There are still abnormal values in the captured RPS sequences from the Data Collection module, owing to the influence of changing light condition and human dynamic movement. As shown in Fig. 6(a), some outliers in the RPS sequence are close to 0 or 1, violating the pupillary response’s physiological phenomenon. Therefore, PupilRec implements a denoising process. Specifically, PupilRec utilizes Boxplots to find these outliers. The principles used to determine outliers by Boxplots are quartiles and interquartile ranges. There is a certain degree of resistance for quartiles. As shown in Fig. 6(b),

Algorithm 1: I-DT

Input : dispersion threshold DT , duration threshold T , eye center coordinate series
 $ECCS = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$
Output: Gazes GZ

while there are still points in $ECCS$ **do**
 Initialize window over first points within time T
 Dispersion of windows points
 $D = [\max(x) - \min(x)] + [\max(y) - \min(y)]$
 if $D \leq DT$ **then**
 Add additional coordinate (x, y) to the window until
 $D > DT$:
 $GZ += \text{window } \{(x_i, y_i), (x_{i+1}, y_{i+1}), \dots (x_w, y_w)\}$
 Remove window points from $ECCS$
 else
 Remove first points from $ECCS$

return GZ .

up to 25% of the data can be arbitrarily far from median without much disturbance to the quartiles. Consequently, outliers do not affect the shape of Boxplot, and can be identified precisely. In conclusion, Boxplot has an advantage in recognizing outliers. Then, the average of two points around the outlier is used to correct it.

Evidently, when browsing on mobile phones, people must spend different time periods gazing at individual contents [33]. Therefore, the gaze period can be used as a valid pupillary response. As gaze points generally cluster at a specific position, PupilRec identifies gaze as groups of consecutive points within a particular dispersion, or maximum separation. Based on this, PupilRec conducts the segmentation to identify the RPS sequence during gaze periods. Specifically, a moving window spanning consecutive eye center coordinates [represented as (x, y)] is utilized to examine possible gazes. The moving window starts from the beginning of the eye center coordinates series and initially spans a minimum number of points, which is determined by a given duration threshold and sampling frequency. Then, PupilRec checks the dispersion of the coordinates in the window by summing the difference between the maximum and minimum x and y values of the coordinates, namely, dispersion $D = [\max(x) - \min(x)] + [\max(y) - \min(y)]$. When the dispersion exceeds the dispersion threshold calculated by the I-DT tool [30], the window does not stand for a gaze, and then it moves one point to the next coordinate. Otherwise, the window is recognized as a gaze and will be extended to the next coordinate until the window's dispersion surpasses the given threshold. For a given start time and duration, the final window is recognized as the gaze at the window coordinates center. The window will keep moving until it reaches the ending of the coordinates series. Specifically, the detailed I-DT algorithm is as Algorithm 1 shows.

The surrounding environment also poses impact on RPS data, and the most influential factor is light. For example, the change of ambient light intensity will affect instant pupil diameter, resulting in misleading saltation of RPS in time domain. Therefore, PupilRec implements the environment-independent processing to suppress the influence of changing light intensity.



(a)



(b)

Fig. 7. Experimental setup and graphical user interface. (a) Capturing a users' RPS with the front-facing camera on a smartphone. (b) Graphic user interface of PupilRec.

Learning from a model of the relationship [19] among pupil diameter, mental workload, and lighting conditions, as (1) shows

$$PD = PD_{\text{light}} + PD_{\text{task}} \quad (1)$$

where PD stands for pupil diameter directly measured by PupilRec, PD_{light} is the average pupil diameter for a certain light condition, and PD_{task} is the normalized pupil diameter induced by a specific task. Specifically, in our study, PD_{light} is the average RPS under the light intensity when users use smartphone, and the light intensity is recorded by the light sensor of the phone's front camera. PD_{light} is also automatically adjusted when lighting conditions change. In other words, users do not need to provide PD_{light} data. In this study, the PD_{task} is the task-evoked pupil diameter when browsing on a mobile phone and $PD_{\text{task}} = PD - PD_{\text{light}}$.

Under conditions where the light intensity (recorded by built-in light sensor on mobile phones) remains constant during gaze, PupilRec simply subtracts the average pupil diameter PD_{light} under current illumination condition from the measured pupil diameter PD , and then PupilRec obtains the PD_{task} . In addition, when light intensity varies dynamically during gaze, PupilRec will efficiently locate moments of abrupt changes in light intensity and record them in order to split the gaze period into several shorter phrases. The above procedure will be repeated for each phrase to eliminate the influence of changing light intensity during gazing period.

To evaluate the PupilRec system, we ask all volunteers to carefully label the degree of interest for each content they view. We ensure the labels' accuracy by users labeling the same content repeatedly. Specifically, the contents are repeatedly viewed at regular intervals and in a different order each time. In this way, each content can be repeatedly labeled to ensure the accuracy of the labels. Note that this labeling process is only needed for research evaluations; PupilRec simply relies on the analysis of users' spontaneous pupillary response to derive the degree of interest in practice. To guarantee objectivity and precision, we have repeated this process a number of times, with a minimum of two days between the experiments. As demonstrated in Fig. 7, a volunteer rates degree of interest in each content she viewed. The interest degree is ranked as 1 (Uninterested), 2 (Possible uninterested), 3 (Fair), 4 (Possible interested), and 5 (Very interested). PupilRec further mines key time-series features from these well-labeled RPS data in the next section.

C. Training Preference Model

In this section, we elaborate the procedure of training a user preference model by extracting features from RPS sequences and tuning the model with a neural network.

1) *RPS Feature Extraction*: After data preprocessing, PupilRec performs feature extraction from labeled RPS sequence segments. Following the inspirations clarified in Section III, PupilRec uses the Python module—Tsfresh package to extract the six predefined key time-series features for each RPS segment.

For each RPS segment denoted as $\mathbf{S}_{\text{RPS}} = \{x_0, x_1, \dots, x_i, x_{i+1}, \dots, x_n\}$, we, respectively, mark the six key features as mean m , variance var , sample entropy ($SampEn$), slope of the front RPS curve k_1 , slope of the end RPS curve k_2 , and the time-series complexity estimate (CE).

For a given RPS segment \mathbf{S}_{RPS} , the mean value m of the RPS segment is calculated as

$$m = \frac{1}{n} \sum_{i=0, \dots, n-1} |x_{i+1} - x_i| \quad (2)$$

where x_i represents the RPS sequence for calculating the feature, and n denotes the length of RPS sequence.

The variance var of the middle part of RPS sequence \mathbf{S}_{RPS} is calculated as

$$var = \frac{\sum_{i=n/3, \dots, 2n/3} (x_i - m)^2}{n/3}. \quad (3)$$

In consideration of the fact that the real-time collected RPS sequence is not necessarily equal in length and CE is also influenced by time-series length, we divide the value of CE by the length of the RPS sequence to obtain the normalized CE . CE is denoted as (4), in which x_i is an RPS data in \mathbf{S}_{RPS}

$$CE = \sqrt{\sum_{i=1}^{n-1} (x_i - x_{i-1})^2}. \quad (4)$$

$SampEn$ is represented by (5), where A represents the number of template vector pairs having $d[x_{m+1}(i), x_{m+1}(j)] < r$, and B denotes the number of template vector pairs having $d[x_m(i), x_m(j)] < r$, in which m represents a given embedding dimension, r is tolerance, and x_i corresponds to an RPS data in the segmented RPS sequence

$$SampEn = -\log \frac{A}{B}. \quad (5)$$

The slope of the front RPS sequence k_1 and the slope of the end RPS sequence k_2 are the slopes of the regression line obtained by the least-square regression of the first and last third RPS sequence. The least-squares regression function is calculated by (6), where (x, y) represents a pair of observations, and $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ denotes the time series, y_i is an RPS data in \mathbf{S}_{RPS} , and $y = f(x, \omega)$ denotes the theoretical function, in which $\omega = [\omega_1, \omega_2, \dots, \omega_n]^T$ represents a parameter to be identified. $L_i(x) (i = 1, 2, \dots, m)$ denotes the residual function. In consequence, slope k is calculated as (7)

$$\min f(x) = \sum_{i=1}^m L_i^2(x) = \sum_{i=1}^m L_i^2[y_i, f(x_i, \omega_i)]$$

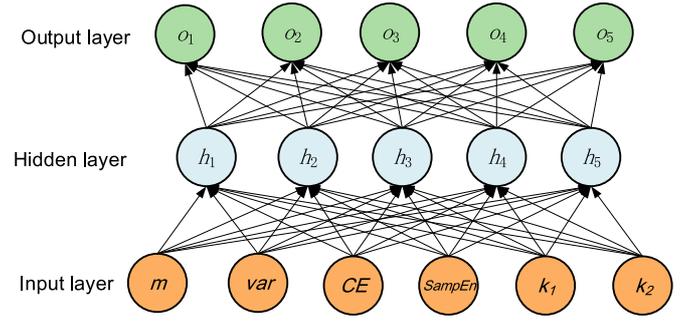


Fig. 8. MLP with hidden layer containing five hidden units for customization of the user preference model.

$$= \sum_{i=1}^m [y_i - f(x_i, \omega_i)]^2 \quad (6)$$

$$k = \frac{d \min f(x)}{dx}. \quad (7)$$

2) *Customized User Preference Model*: PupilRec customizes a user preference model through utilizing the key time-series features of each carefully labeled RPS sequence segment. As mentioned in Section III-A, it is infeasible to derive a heuristic model to find out users' complicated and variable preferences by simply using the quantitative features.

As illustrated in the *Data Preprocessing* module, in PupilRec, the interest degree representing user preference can be classified into five levels: 1) uninterested; 2) possible uninterested; 3) fair; 4) possible interested; and 5) very interested. Therefore, the customization of user preference can be considered as a representative multiclassification problem. As MLP is used to solve multiclassification problems and works effectively on nonlinear data [34], PupilRec adopts MLP to classify the labeled RPS sequence segments.

Specifically, MLP is a neural network composed of fully connected layers with at least one hidden layer as shown in Fig. 8. An activation function transforms the output of each hidden layer. The number of MLP layers and hidden units in each hidden layer are hyperparameters. Given an example of a single hidden layer, MLP calculates the output as follows:

$$\mathbf{H} = \phi(\mathbf{X}\mathbf{W}_h + \mathbf{b}_h) \quad (8)$$

$$\mathbf{O} = \mathbf{H}\mathbf{W}_o + \mathbf{b}_o \quad (9)$$

where \mathbf{H} denotes the hidden layer's output, and \mathbf{O} represents the MLP's output. In addition, ϕ denotes the activation function, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ refers to samples. The weight and bias of the hidden layer are $\mathbf{W}_h \in \mathbb{R}^{d \times h}$ and $\mathbf{b}_h \in \mathbb{R}^{1 \times h}$, respectively. Accordingly, weight and bias of the output layer are $\mathbf{W}_o \in \mathbb{R}^{h \times q}$ and $\mathbf{b}_o \in \mathbb{R}^{1 \times q}$, respectively. In this classification issue, the softmax operation is done on the output \mathbf{O} and the cross-entropy loss function is used in softmax regression. PupilRec inputs the labeled RPS sequence segments and captured key time-series features to the customized MLP for each user. Then, parameters are adjusted to fit the user's data. Finally, the user's customized preference model is obtained by MLP's classification results.

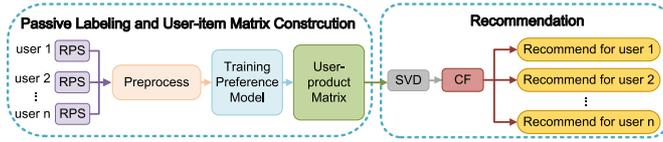


Fig. 9. Application framework.

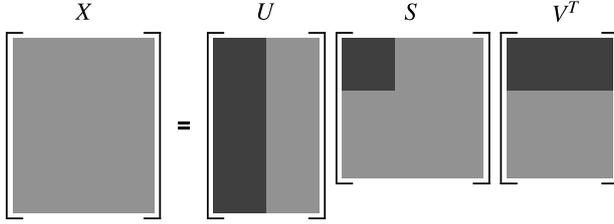


Fig. 10. Calculation process of SVD.

After finishing customizing user preference model, PupilRec runs as a daemon. While a user browsing service platforms adopting PupilRec to model user's preference, PupilRec captures RPS sequences corresponding to each user's viewed content through the front-facing camera. Thereafter, by denoising and segmentation, PupilRec extracts the key time-series features of each segmented RPS sequence. After inputting these key features into MLP, PupilRec infers the degree of user's interest in the counterpart.

D. From Preference to Recommendation

Based on the customized user preference model in Section IV-C2, PupilRec utilizes SVD and CF to produce recommendations to users. The application architecture is illustrated in Fig. 9.

1) *Singular Value Decomposition*: When people browsing on mobile phones, through the correlation between user preference and instant pupillary response, PupilRec acquires ratings of the viewed product and establishes a user-product rating matrix. PupilRec recommends to users based on CF via this matrix [35]. Moreover, to improve the recommendation efficiency, SVD is used for simplifying the user-product matrix. SVD is a well-known matrix factorization technique factoring a matrix X into three matrices as (10) shows

$$X_{m \times n} = U_{m \times r} S_{r \times r} V_{r \times n}^T \quad (10)$$

$$X_{m \times n} \approx U_{m \times k} S_{k \times k} V_{k \times n}^T, \quad k < r. \quad (11)$$

In (10), the matrix S is a diagonal matrix containing singular values of matrix X . According to the optimal truncation property of SVD, we can get a refined matrix, if discard all but the k ($k < r$) largest singular values and corresponding singular vectors. Therefore, the original matrix X can be approximated by (11). Fig. 10 is the schematic of the above calculation process.

In our study, 30 volunteers are required to watch 100 different food images, forming a user-food matrix. Since the number of users is smaller than the number of food types, we choose to transform the food set into a low-dimensional space, namely, user taste space. Then, PupilRec targets users based on the similarity of personal tastes. Considering that

there may be extreme data, we use a Pearson correlation coefficient to calculate the similarity among different users' tastes. The Pearson correlation coefficient is calculated as (12) shows, where T_1 and T_2 represent two vectors of different users in taste space, μ_{T_1} and μ_{T_2} are the mean value of T_1 and T_2 , respectively, and σ_{T_1} and σ_{T_2} denote the variance of T_1 and T_2 , respectively

$$\rho_{T_1, T_2} = \frac{\text{cov}(T_1, T_2)}{\sigma_{T_1} \sigma_{T_2}} = \frac{E[(T_1 - \mu_{T_1})(T_2 - \mu_{T_2})]}{\sigma_{T_1} \sigma_{T_2}}. \quad (12)$$

On that basis, PupilRec runs on the server side and obtains good recommendation results. However, the operating frequency of the system will be reduced when encountering a large-scale database. For example, it may run once per day and can only run offline. Therefore, under this condition, PupilRec cannot respond quickly to new data added by the user and timely complete recommendations. Inspired by the updated SVD method proposed by Brand [36], we realize the timely recommendation. Specifically, the updated SVD method can add, modify, or withdraw a single item at any time on the thin SVD. In fact, the thin SVD is the original SVD being reduced in dimensionality. Then, the timely recommendation is achieved. Therefore, the updated SVD method is appropriate for mobile recommendation. The basic principle of the updated SVD algorithm is as follows.

Given the SVD of a matrix X , we want to find the SVD of the matrix $X + ab^T$, where a and b are column vectors. Given the SVD $X = USV^T$, let $m = U^T a$, $p = a - Um$, $p' = \text{sqrt}(p^T p)$, and $P = p/p'$. Similarly let $n = V^T b$, $q = b - Vn$, $q' = \text{sqrt}(q^T q)$, and $Q = q/q'$. Then, we first find the SVD $U'S'V'^T$ of the matrix as (13) shows. And finally, the SVD of our new matrix is given by (14). Since we can use low-rank approximations of U , S , and V , the algorithm is very efficient. Specifically, Brand [36] showed that the time complexity of the updated SVD is $O(mnk)$, where m and n are the dimensions of the matrix, and k is the reduced rank of the approximation

$$\begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} m \\ p \end{bmatrix} \begin{bmatrix} n \\ q \end{bmatrix} \quad (13)$$

$$X + ab^T = ([U, P]U')S'([V, Q], V')^T. \quad (14)$$

2) *Collaborative Filtering*: After processing the user-item matrix of multiple users through SVD, we use the CF algorithm to predict labels of items having no value in user-item matrix, so as to realize personalized recommendation for each user.

CF is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the CF approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B 's opinion on a different issue than that of a randomly chosen person.

CF includes user-based and product-based CF algorithms. The selection of these two kinds of algorithms depends on the number of users and products. Specifically, the calculation time of product-based and user-based CF will increase with the increasing number of products and users, respectively. Moreover, in real-life application, the number of users

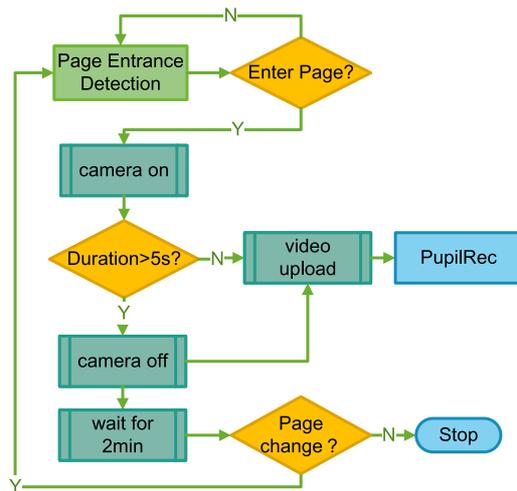


Fig. 11. Energy optimization module.

is much larger than products. Therefore, in order to improve computing efficiency, we choose the product-based CF as the recommendation algorithm.

Specifically, the product-based CF recommends to users through finding similarities between pairs of products. Then, we adopt cosine similarity as similarity measurement. Accordingly, PupilRec conducts a recommendation process by listing products most similar to users' already-rated products and returning the top N products to users.

V. ENERGY OPTIMIZATION

Since PupilRec is implemented on mobile phones, energy consumption should be considered as an important way to improve recommendation performance without affecting user experience. We mainly optimize energy consumption by governing the operation time of the front-facing camera on mobile phones. The specific process is as follows.

- 1) As shown in Fig. 11, the energy optimization module does the page entrance detection at first. If a user has been detected entering a certain page, the module will turn on the camera, otherwise, continue the page entrance detection.
- 2) The energy optimization module then implements camera duration detection. If a camera turns on for more than 5 s, the module will turn off it and upload the recorded video to server; otherwise, the module will upload the video shorter than 5 s directly to server without switching off camera. The uploaded videos are input into PupilRec.
- 3) The module checks whether the user switches the current page within 2 min. If the page is switched, it will return to step 1). Otherwise, it will stop the detection and wait for the user to enter a new page.

To supplement, in step 2), we set the duration threshold as 5 s in that we only need pupil information for the first impression generated within 5 s [37]. Moreover, in step 3), we set the waiting time for page change as 2 min. The reason is that if a person stays on a same page for over 2 min [38], he/she

must be interested in this page with huge possibility and may keep watching for a longer time. Therefore, to save the energy, there is no need to keep monitoring on the same page.

VI. EVALUATION

We prototype the implementation of PupilRec on HUAWEI P40. Data analysis is performed on a desktop with an Intel i7-5700 CPU and 16 G RAM running Windows 10 with JetBrains PyCharm 2019 software. The PupilRec prototype captures an image of the user's pupil using the front-facing camera. In this section, we evaluate the validity of PupilRec. We first present the experimental setup and evaluation metrics in Section VI-A. We then justify the use of the key time-series features to model user preference in Section VI-B. Finally, we perform extensive experiments to estimate the overall performance, including recommendation performance of PupilRec, and investigate the influence of different factors on performance in Section VI-C.

A. Experimental Setup and Metrics

In this section, we introduce the settings and evaluation metrics of the experiments.

1) *Experimental Setting*: A sum of 30 volunteers (15 females and 15 males) aged from 18 to 35 participate in the evaluation. The participants have normal vision or wear glasses without color. Participants were compensated based on their participation time in the study (\$5 for 2 h) and compliance rate (\$0.5 for each completed task). All collected data were kept anonymous and the Institutional Review Board of our university authorized all the study procedures.

The experiments are conducted in a normal quiet office environment. Participants are sitting and looking at the content on the mobile phone. The overall brightness of ambient and screen light to participants' eyes is about 240 lux. The distance between participants' eyes and the front-facing camera on mobile phone is about 32 cm. The participants are required to view 100 pictures/short videos randomly shown on the screen, and they could switch to the next picture/video at will, imitating real-world scenarios where a user browses on the mobile phone. Meanwhile, they are asked to label each viewed picture/video with one of the five levels mentioned in Section IV-B2. The front-facing camera will capture eye area of participants while viewing the displayed pictures/videos. Specifically, the videos recorded by the camera have a resolution of 720 p and a frame rate of 30 frames/s. In addition, the training set accounts for 70% of total validated data, and the test set is the remaining 30% .

2) *Evaluation Metrics*: PupilRec adopts MLP to customize user's preference model. Moreover, as MLP is utilized to resolve the multiclassification problem in PupilRec, we evaluate the performance of PupilRec using typical indicators of multiclassifications. Specifically, the indicators are Kappa coefficient, Jaccard similarity coefficient, Hamming distance, and Hinge loss. The indicators are clarified as follows.

- 1) *Kappa Coefficient*: The Kappa coefficient is a model evaluation parameter derived from the calculation of confusion matrix. The Kappa coefficient is determined by

$$k = \frac{P_0 - P_e}{1 - P_e} \quad (15)$$

where P_0 denotes the result of dividing the number of correctly classified samples in each category by the total number of samples, i.e., the overall classification accuracy. Suppose the real sample size of each category is a_1, a_2, \dots, a_c , the predicted sample size of each category is b_1, b_2, \dots, b_c , and the total sample size is n , then

$$P_e = \frac{a_1 \times b_1 + a_2 \times b_2 + \dots + a_3 \times b_3}{n \times n}. \quad (16)$$

The range of the Kappa coefficient is $[-1, 1]$. However, it usually falls between 0 and 1. On that basis, the range can be divided into five groups to indicate different levels of consistency: $[0.0, 0.20]$ for slight consistency, $[0.21, 0.40]$ for fair consistency, $[0.41, 0.60]$ for moderate consistency, $[0.61, 0.80]$ for substantial consistency, and $[0.81, 1]$ for almost perfect [39].

- 2) *Hamming Distance*: The Hamming distance is suitable for evaluating multiclassification results. In simple terms, it measures the distance between the predicted label and the real label, with a value between 0 and 1. Specifically, a distance of 0 means the predicted result is completely identical to the real result. In contrast, a distance of 1 means the model is the exact opposite of what we want.
- 3) *Jaccard Similarity Coefficient*: The differentiation between the Jaccard similarity coefficient and Hamming distance is in the denominator. The coefficient is 1 when the predicted result is exactly the same as the actual situation, 0 when the predicted result is exactly the opposite of the actual situation, and a distance between 0 and 1 when the predicted result is a proper subset or true superset of the actual situation.
- 4) *Hinge Loss*: The goal of Hinge loss is of keeping the misclassified user preferences far enough away from the correctly classified user preferences. If this distance attains a threshold value, then the error in misclassified user preferences can be recognized as 0. Otherwise, computational error is accumulated. Therefore, the Hinge loss value ranges between 0 and 1. In particular, when the value is 0, it means that the multiclass model is completely accurate in its classification. In contrast, a value of 1 means that the model does not work at all.

B. Rationality of Key Time-Series Features

To further justify the six key time-series features described above, 30 volunteers (15 males and 15 females) aged from 18 to 35 were arranged to join in the experiment. Each participant consecutively watched 100 pictures/short videos previously prepared. After watching a content on the mobile phone, the participants were required to label it as interested or uninterested and then switched to the next one. The light intensity remained around 240 lux. The experiment was conducted in a quiet office environment.

As a result of the experiment mentioned above, Fig. 12 shows probability density distribution diagrams of the six key

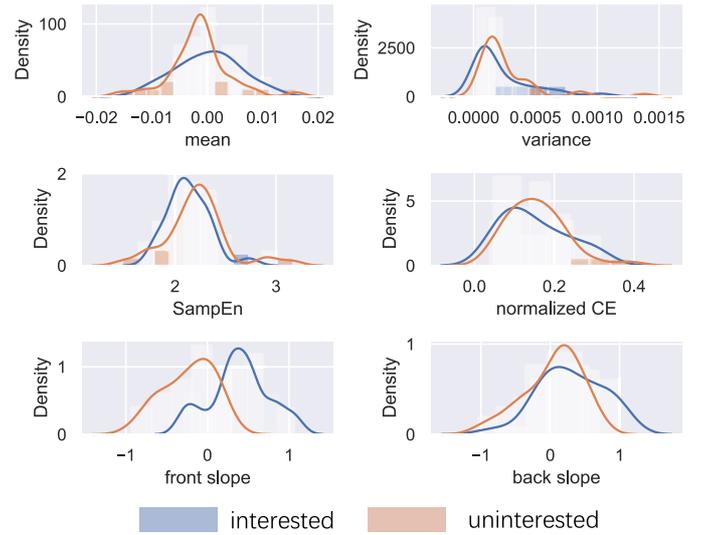


Fig. 12. Statistics histograms of key time-series features in interested and uninterested cases.

time-series features for both cases of interest and uninterest. From Fig. 12, we can see that that in the case of interest, the peak of the kernel density estimation curve of the mean is to the right of the uninterest, i.e., the average RPS of interest is mostly larger than the uninterest. This is in accordance with the conclusion proposed in Section III-A: when the user is interested, the pupil will dilate, while when the user is not interested, the pupil will not dilate or even shrink. Similarly, Fig. 12 shows that the variance of the RPS sequences for the interested case is significantly smaller than that of the uninterested one, indicating that the fluctuations of the RPS sequences in the uninterested case are larger than those in the interested one. This also agrees with the conclusion in Section III-A that when users are uninterested in what they are watching, their eyes will wander around instead of gazing at the content, resulting in pupil size fluctuations.

The probability density distribution of the front and back slopes of the RPS sequences are also plotted at the bottom of Fig. 12. As for the kernel density estimation curve of the front slope, we can find that the curve in the interested case is to the right of the uninterested one. Moreover, the horizontal coordinate of the peak of the curve in the case of interest is positive, while the one in the case of uninterest is negative. This phenomenon suggests that the trend of RPS sequences in the case of interest is generally increasing at the beginning, and conversely, the trend of RPS sequences in the case of uninterest is monotonically decreasing. This result agrees with the previous findings presented in Section III-A: the RPS sequences are initially skewed to increase and decrease when people viewing interested and uninterested content, respectively.

For the kernel density estimation curve of the back slope, there is not much difference between the cases of interest and uninterest. The horizontal coordinate of the curves' peak values are very near the origin of coordinates, and their absolute values are less than the front slope, indicating that most RPS sequences show only a slight increase at the end in both the cases of interest and uninterest. Moreover, we can notice that

TABLE II
RESULTS OF PREDICTION ACCURACY

	Mean	Median	Standard Deviation
Kappa coefficient	0.8358	0.8518	0.0402
Jaccard similarity coefficient	0.7939	0.7894	0.0499
Hamming distance	0.1864	0.1858	0.0387
Hinge loss	0.1179	0.1143	0.0272

the back slope of the RPS in the case of interest is smaller than that of uninterest. The reason is that when people are gazing at things of interest, they tend to remain in this state when they are switching to the next content, causing slight growth of RPS. In contrast, when watching something that does not interest them, people get more likely to leave current content and immediately move to new content, causing a greater increase in RPS.

In a similar way, we can observe from Fig. 12 that the kernel density estimation curves about *SampEn* and *CE* for the interested case is to the left of the uninterested one. This phenomenon indicates that the *SampEn* and *CE* of the RPS sequences in the interested case are smaller than those in the uninterested case. This is also in agreement with the findings mentioned in Section III-A that the fluctuations in the middle part of the RPS sequences in the interested case are smaller than those in the uninterested ones.

In conclusion, it is reasonable to use the proposed key time-series features to express the change process of RPS sequences and correlate it with user preference.

C. System Performance

We first evaluate the overall performance of PupilRec in different environments and then investigate the impacts of different factors on the performance.

1) *Prediction Accuracy*: To estimate the accuracy of PupilRec's prediction, we recruit 30 volunteers to participate in the experiment. Each person views 100 images on a mobile phone screen and can turn to the next picture whenever he/she want. The Kappa coefficient, Jaccard similarity coefficient, Hamming distance, and Hinge loss are calculated. The results are demonstrated in Table II and Fig. 13. The mean values of the Kappa coefficient, Jaccard similarity coefficient, Hamming distance, and Hinge loss are 0.8358, 0.7939, 0.1864, and 0.1179, respectively. The Kappa coefficient, Jaccard similarity coefficient, Hamming distance, and Hinge loss have median values of 0.8518, 0.7894, 0.1858, and 0.1143, respectively. These results indicate that PupilRec predicts user preferences with a relatively high accuracy after training.

2) *Impact of Light Intensity*: To investigate the effect of light intensity on the performance of PupilRec, we configure three different light intensities by tuning the brightness of the ambient light. Specifically, we vary the light intensity by adjusting the number of lamps in the office, i.e., weak light intensity Li_w (110–150 lux) of one lamp, medium light intensity Li_m (220–260 lux) of two lamps, and strong light intensity Li_s (280–300 lux) of three lamps. There are 30 volunteers aged from 18 to 35 participating in this experiment. Each

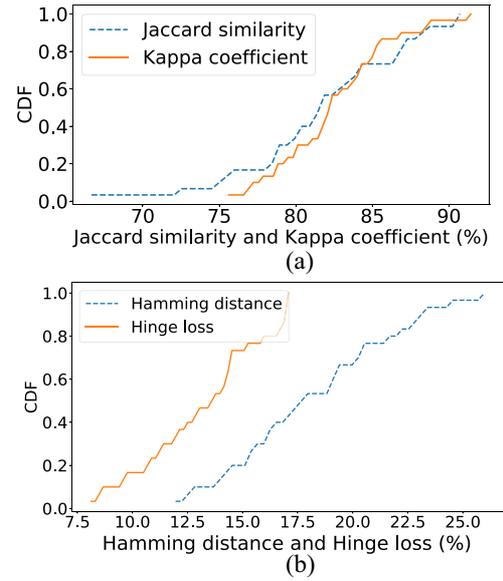


Fig. 13. Overall performance of PupilRec. (a) CDF plot of Jaccard similarity coefficient and Kappa coefficient. (b) CDF plot of Hamming distance and Hinge loss.

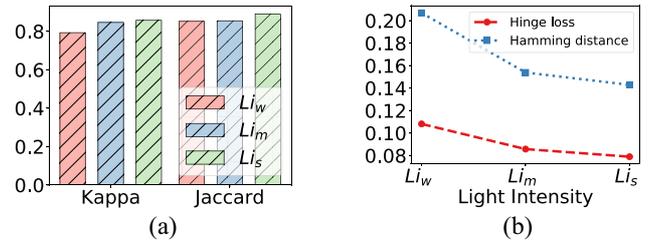


Fig. 14. PupilRec's performance under different lighting conditions. (a) Kappa coefficient and Jaccard similarity coefficient. (b) Hamming distance and Hinge loss.

volunteer watches different groups of pictures in a quiet, normal office under three lighting conditions, with 100 pictures in each group. We use RPS sequences captured under each light intensity to train PupilRec and then measure PupilRec's performance under these three light intensities.

We measure the Kappa coefficient, Jaccard similarity coefficient, Hamming distance, and Hinge loss for these three light intensities, and the results are presented in Fig. 14. As illustrated in Fig. 14, the lighting conditions indeed have an influence on the PupilRec's performance. Specifically, the Kappa coefficient of Li_w , Li_m , and Li_s is 85.18%, 85.48%, and 88.89%, respectively, namely, the Kappa coefficient increases with increasing light intensity. Conversely, the Hinge loss and Hamming distance decrease with the increase of light intensity. This implies that the higher light intensity improves the performance of PupilRec to some extent, due to the fact that PupilRec detects pupil images with greater clarity at higher light intensity.

3) *Impact of Shooting Distance*: The quality of the recorded video is affected by the distance between the eye and the front-facing camera on mobile phones. In this section, we estimate the effect of shooting distance on the accuracy of PupilRec. A total of 30 volunteers between the ages of

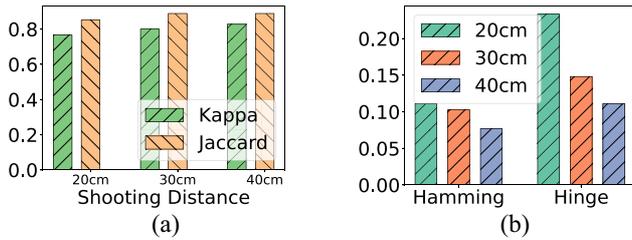


Fig. 15. PupilRec's performance under different shooting distances. (a) Kappa coefficient and Jaccard similarity coefficient. (b) Hamming distance and Hinge loss.

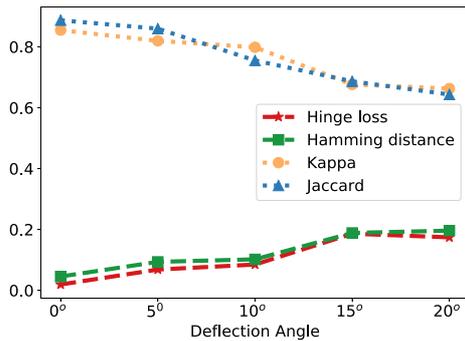


Fig. 16. PupilRec's performance under different deflection angles.

18 and 35 participated in the experiment (15 males and 15 females). Each volunteer watches different groups of pictures at three different shooting distances in a quiet general office, with 100 photographs in each group. Furthermore, volunteers are holding the mobile phone and facing the camera without offset. We maintain the light intensity at a medium level L_{im} by leaving two lamps on in the office. Then, we evaluate the performance of PupilRec at different shooting distances (20, 30, and 40 cm).

The result is displayed in Fig. 15. We can observe that the increase of the Kappa coefficient and Jaccard similarity coefficient gradually becomes slow with increasing distance, and so do the decreases of the Hamming distance and Hinge loss. Therefore, the distance between the eye and the front-facing camera has a negligible effect on the PupilRec performance within a certain range. The reason for this is that in practice, the distance between the eyes and the screen is usually within the range of 20 to 40 cm when users are browsing normally on mobile phones.

4) *Impact of Deflection Angle*: In order to evaluate the impact of deflection angle, we recruited 30 volunteers aged between 18 and 35 (15 males and 15 females). In the experiment, the volunteers were required to watch different groups of pictures from five different deflection angles in a quiet general office, with 100 photographs in each group.

The results are presented in Fig. 16. As shown in Fig. 16, with the increase of deflection angle, the Kappa coefficient and Jaccard similarity coefficient are decreasing. We can also observe that the Hinge loss and Hamming distance are increasing, meaning that when a person's head is deflected in relation to the mobile phone's screen, PupilRec's performance will be

deteriorated. Specifically, in Fig. 16, as the deflection angle is over 10° , these four metrics obviously vary greatly. This phenomenon can be explained that when a user is browsing normally on the phone, the deflection angle between the eye and the screen ranges from -10° to 10° . When the deflection angle exceeds this range, PupilRec will not be able to clearly recognize the user's pupil and iris. Therefore, PupilRec is suited for scenarios where the deflection angle between users' eyes and the screen is small, namely, the user is facing the screen.

5) *Recommendation Performance*: For the sake of evaluating PupilRec's recommendation performance, we compare it with the traditional manual-scoring-based CF algorithm.

In the experiment, we recruited 37 volunteers aged between 18 and 35, including 19 males and 18 females. Each volunteer watched 50 pictures randomly selected from the same 100 food pictures. Moreover, the pictures viewed by each volunteer may not be exactly the same. Then, volunteers are required to manually label contents while watching on mobile phones. Note that the labels are different from the previous labels as ground truth, and these labels are not necessarily the real feelings of volunteers at that time. At the same time, PupilRec returns recommended content for each participant based on their pupillary responses. Meanwhile, recommended contents based on participants' manual labeling are also generated by virtue of CF. Finally, each participant is required to carefully label these two kinds of recommended contents as ground truth, one is given by PupilRec, and the other is based on traditional CF with users' manual labels.

We calculate the root-mean-square-error (RMSE) value between the ground truth and these two recommended contents' labels. RMSE is calculated as (17), where u represents the user, i represents the item, r_{ui} is the actual label of the user u on i , and \hat{r}_{ui} is the predicted label given by the recommendation algorithm, and T is the user-item set

$$\text{RMSE} = \sqrt{\frac{\sum_{u,i \in T} (r_{ui} - \hat{r}_{ui})^2}{|T|}}. \quad (17)$$

The RMSE of the recommendation system based on users' manual labeling is 0.86, while PupilRec is 0.78. This is in accordance with the previously mentioned conjecture, that is, users' manual labeling sometimes do not reflect users' true feelings. Fortunately, the changes in pupils can to some extent indicate users' inner thoughts. In conclusion, PupilRec can recommend accurately based on users' inner feelings.

VII. DISCUSSION

In this section, we discuss the effects of various environmental conditions on PupilRec.

- 1) *Extreme Light Intensity*: In our experiments, we investigate the performance of PupilRec over a range of light intensities (110–300 lux). When the light intensity is at extremely high or low levels, for example, when the user is in the dark or under strong outdoor light, the images of the user's pupil collected by PupilRec will become blurry. In addition, the accuracy of pupil detection will be reduced in particularly high light conditions.

Nonetheless, we expect that most users would not use PupilRec in such a condition.

- 2) *Extreme Shooting Distance*: We consider the performance of PupilRec when the user is viewing normally on the mobile phone screen. However, when users' eyes are very close to the front-facing camera, for instance, when the user suffers from myopia, PupilRec cannot detect the user's face and therefore cannot detect the pupil. Furthermore, when users' eyes are far from the screen, the front-facing camera may collect blurred facial images because of the limited resolution of the camera. Fortunately, as users always choose the most comfortable way of using their phones, these scenarios rarely take place in reality.
- 3) *User Wearing Glasses*: Volunteers participating in the experiment wear regular glasses or no glasses. However, in real life, when users wear tinted glasses, such like sunglasses, the resolution of the eyes in the pictures will reduce. Being the first step toward a pupil-driven recommendation system, PupilRec by far has not taken such cases into account.
- 4) *Camera Resolution and Frame Rate*: To explore the effect of different camera resolutions, we record videos using mobile phones with different cameras resolutions of 360P, 480P, 720P, 1080P and a fixed frame rate of 60 frames/s. The results demonstrate that the videos recorded by the 360P and 480P cameras are too blurry and PupilRec cannot segment the user's pupils and iris well. Moreover, to determine the effect of different camera frame rates, we use a camera with a fixed resolution of 720P to record videos at frame rates of 30, 60, and 120 frames/s, and the results show little difference. However, we plan to have more in-depth evaluation on these issues, as well as the issue of phone diversity.

VIII. CONCLUSION

In this article, we have proposed PupilRec as a computer-vision-based recommendation system, including a mobile terminal and a server side. On the mobile terminal, PupilRec has collected pupil-related videos through the front-facing camera on mobile phones. On the server side, upon preprocessing pupil size information, PupilRec has figured out the key time-series features and trained a neural network to fit a user preference model. On that basis, We have prototyped PupilRec and conducted experimental and field studies to thoroughly evaluate the efficacy of PupilRec by recruitment of 67 volunteers. The overall results have shown that PupilRec can accurately estimate a user's preference for a certain product, and can recommend products users interested in. Moreover, the experimental results show that PupilRec can overcome the environmental impacts such as changing light intensities and varying shooting distances. In general, PupilRec provides us with a prototype for exploring the relationships between pupil size and user preferences, shedding lights on a viable yet innovative idea for realizing mobile recommendation systems based on user inner feelings. In future work, we will expand the diversity of experiments in terms of devices, subjects,

and environmental conditions to further improve our PupilRec system.

REFERENCES

- [1] H. Jiang, X. Shen, and D. Liu, "PupilMeter: Modeling user preference with time-series features of pupillary response," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2021, pp. 1031–1041.
- [2] W. Huang, W. Tang, H. Jiang, J. Luo, and Y. Zhang, "Stop deceiving! An effective defense scheme against voice impersonation attacks on smart devices," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5304–5314, Apr. 2022.
- [3] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Cham, Switzerland: Springer, 2011, pp. 1–35.
- [4] J. He, F. Li, Z. Li, and H. Liu, "The effect of mobile marketing design on consumer mobile shopping," *Complexity*, vol. 2021, pp. 1–10, Apr. 2021.
- [5] A. Shutsko, "User-generated short video content in social media. A case study of TikTok," in *Proc. Int. Conf. Human Comput. Interact.*, 2020, pp. 108–125.
- [6] X. Xu *et al.*, "Understanding user behavior for document recommendation," in *Proc. Web Conf. (WWW)*, 2020, pp. 3012–3018.
- [7] Y. Yuan, X. Luo, M. Shang, and D. Wu, "A generalized and fast-converging non-negative latent factor model for predicting user preferences in recommender systems," in *Proc. Web Conf. (WWW)*, 2020, pp. 498–507.
- [8] G. He, J. Li, W. X. Zhao, P. Liu, and J.-R. Wen, "Mining implicit entity preference from user-item interaction data for knowledge graph completion via adversarial learning," in *Proc. Web Conf. (WWW)*, 2020, pp. 740–751.
- [9] J. M. Groot *et al.*, "probing the neural signature of mind wandering with simultaneous fMRI-EEG and pupillometry," *NeuroImage*, vol. 224, Jan. 2021, Art. no. 117412.
- [10] C. Wangwittatana, X. Ding, and E. C. Larson, "PupilNet, measuring task evoked pupillary response using commodity RGB tablet cameras: Comparison to mobile, infrared gaze trackers for inferring cognitive load," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–26, Jan. 2018.
- [11] J. Chen, F. Zhuang, X. Hong, X. Ao, X. Xie, and Q. He, "Attention-driven factor model for explainable personalized recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2018, pp. 909–912.
- [12] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. Kankanhalli, "A³NCF: An adaptive aspect attention model for rating prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 3748–3754.
- [13] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: Rating prediction with ratings and reviews," in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 639–648.
- [14] J. Y. Chin, K. Zhao, S. Joty, and G. Cong, "ANR: Aspect-based neural recommender," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2018, pp. 147–156.
- [15] Y. Guo, Z. Cheng, L. Nie, X.-S. Xu, and M. Kankanhalli, "Multi-modal preference modeling for product search," in *Proc. 26th ACM Int. Conf. Multimedia (MM)*, 2018, pp. 1865–1873.
- [16] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst. (RecSys)*, 2016, pp. 191–198.
- [17] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 135–142.
- [18] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2016, pp. 729–732.
- [19] B. Pflieger, D. K. Fekety, A. Schmidt, and A. L. Kun, "A model relating pupil diameter to mental workload and lighting conditions," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2016, pp. 5776–5788.
- [20] C. K. Foroughi, J. T. Coyne, C. Sibley, T. Olson, C. Moclair, and N. Brown, "Pupil dilation and task adaptation," in *Proc. Int. Conf. Augmented Cogn.*, 2017, pp. 304–311.
- [21] W. Wang, Z. Li, Y. Wang, and F. Chen, "Indexing cognitive workload based on pupillary response under luminance and emotional changes," in *Proc. Int. Conf. Intell. User Interfaces*, 2013, pp. 247–256.
- [22] X. Jiang, M. S. Atkins, G. Tien, R. Bednarik, and B. Zheng, "Pupil responses during discrete goal-directed movements," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2014, pp. 2075–2084.

- [23] E. H. Hess and J. M. Polt, "Pupil size as related to interest value of visual stimuli," *Science*, vol. 132, no. 3423, pp. 349–350, 1960.
- [24] E. H. Hess, "Attitude and pupil size," *Sci. Amer.*, vol. 212, no. 4, pp. 46–54, 1965.
- [25] V. W. S. Tseng, S. Abdullah, J. M. R. Costa, and T. Choudhury, "AlertnessScanner: What do your pupils tell about your alertness," in *Proc. 20th Int. Conf. Human-Comput. Interact. Mobile Devices Services (MobileHCI)*, Sep. 2018, p. 41.
- [26] "Tsfresh." [Online]. Available: <https://tsfresh.readthedocs.io/en/latest/index.html> (Accessed: Dec. 21, 2021).
- [27] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *J. Comput. Graph. Stat.*, vol. 15, no. 1, pp. 118–138, 2006.
- [28] A. T. Duchowski *et al.*, "The index of pupillary activity: Measuring cognitive load vis-à-vis task difficulty with pupil oscillation," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2018, pp. 1–13.
- [29] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *Amer. J. Physiol. Heart Circulatory Physiol.*, vol. 278, no. 6, pp. H2039–H2049, 2000.
- [30] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proc. Eye Tracking Res. Appl. Symp. ETRA*, Nov. 2000, pp. 71–78.
- [31] "HaaR cascade classifier." 2017. [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html
- [32] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.
- [33] S. Angelastro, B. De Carolis, and S. Ferilli, "learning and predicting user pairwise preferences from emotions and gaze Behavior," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Companion Vol.*, 2019, pp. 72–79.
- [34] R. Collobert and S. Bengio, "Links between perceptrons, MLPs and SVMs," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 23.
- [35] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.
- [36] M. Brand, "Fast online SVD revisions for lightweight recommender systems," in *Proc. SIAM Int. Conf. Data Min. (SDM)*, 2003, pp. 37–46.
- [37] L. Harrison, K. Reinecke, and R. Chang, "Infographic aesthetics: Designing for the first impression," in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst. (CHI)*, 2015, pp. 1187–1190.
- [38] C. Liu, R. W. White, and S. T. Dumais, "Understanding Web browsing Behaviors through Weibull analysis of dwell time," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2010, pp. 379–386.
- [39] M. J. Warrens, "Kappa coefficients for dichotomous-nominal classifications," *Adv. Data Anal. Classif.*, vol. 15, no. 1, pp. 193–208, 2021.



Hongbo Jiang (Senior Member, IEEE) received the Ph.D. degree from Case Western Reserve University, Cleveland, OH, USA, in 2008.

He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. He was a Professor with Huazhong University of Science and Technology, Wuhan, China. His current research focuses on computer networking, especially, wireless networks, data science in Internet of Things, and mobile computing.

Prof. Jiang has been serving on the editorial board of IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON MOBILE COMPUTING, *ACM Transactions on Sensor Networks*, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and IEEE INTERNET OF THINGS JOURNAL. He was also invited to serve on the TPC of IEEE INFOCOM, ACM WWW, ACM/IEEE MobiHoc, IEEE ICDCS, and IEEE ICNP. He is an Elected Fellow of The Institution of Engineering and Technology, a Fellow of The British Computer Society, a Senior Member of ACM, and a Full Member of IFIP TC6 WG6.2.



Daibo Liu (Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2018.

He was a Visiting Researcher with the School of Software, Tsinghua University, Beijing, China, from 2014 to 2016, and the Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, WI, USA, from 2016 to 2017. He is currently an Assistant Professor with the College of Computer Science and Electronic

Engineering, Hunan University, Changsha, China. His research interests cover the broad areas of low-power wireless networks, mobile and pervasive computing, and system security.

Dr. Liu is a member of ACM.



Kehua Yang received the Ph.D. degree in computer science and engineering from Southeast University, Nanjing, China, in 2005.

He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His major research interests include embedded systems, cyber-physical systems, and automotive systems.



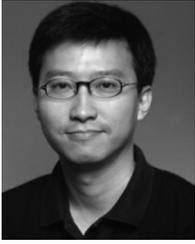
Xiangyu Shen (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree in computer science and technology with the College of Computer Science and Electronic Engineer, Hunan University, Changsha, China.

He has published papers in IEEE TVT 2018 and IEEE ICDCS 2021. His research interests focus on mobile sensing, mobile computing, and machine learning.



Feiyang Deng received the B.S. degree in electromagnetic field and wireless technology from Harbin Institute of Technology, Harbin, China, in 2019. He is currently pursuing the Ph.D. degree with The City University of Hong Kong, Hong Kong.

His research interests include antenna, millimeter-wave radar, machine learning, and computer vision.



John C. S. Lui (Fellow, IEEE) was born in Hong Kong. He received the Ph.D. degree in computer science from the University of California at Los Angeles, Los Angeles, CA, USA, in 1992.

He is currently the Choh-Ming Li Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong, where he was the Chairman from 2005 to 2011. His current research interests are in communication networks, network/system security (e.g., cloud security and mobile security), network economics, network sciences (e.g., online social networks and information spreading), cloud computing, large-scale distributed systems, and performance evaluation theory.

Prof. Lui received various departmental teaching awards and the CUHK Vice-Chancellors Exemplary Teaching Award. He is also a co-recipient of the Best Paper Award at IFIP WG 7.3 Performance 2005, IEEE/IFIP NOMS 2006, and SIMPLEX 2013. He has been serving on the editorial board of the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *Performance Evaluation*, and the *International Journal of Network Security*. He is an Elected Member of IFIP WG 7.3, a Fellow of the Association for Computing Machinery (ACM), and a Senior Research Fellow of the Croucher Foundation, and was the Chair of the ACM SIGMETRICS.



Jiangchuan Liu (Fellow, IEEE) received the B.Eng. degree (*cum laude*) in computer science from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree in computer science from The Hong Kong University of Science and Technology, Hong Kong, in 2003.

He is a University Professor with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. He was an EMC-Endowed Visiting Chair Professor with Tsinghua University from 2013 to 2016. He worked as an Assistant

Professor with The Chinese University of Hong Kong, Hong Kong, and as a Research Fellow with Microsoft Research Asia, Beijing. His research interests include multimedia systems and networks, cloud and edge computing, social networking, online gaming, and Internet of Things/RFID/backscatter.

Dr. Liu is a co-recipient of the inaugural Test of Time Paper Award of IEEE INFOCOM 2015, the ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award in 2013, and the ACM Multimedia Best Paper Award in 2012. He has served on the editorial boards of IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and IEEE INTERNET OF THINGS JOURNAL. He is a Steering Committee Member of IEEE TRANSACTIONS ON MOBILE COMPUTING and a Steering Committee Chair of IEEE/ACM IWQoS from 2015 to 2017. He is the TPC Co-Chair of IEEE INFOCOM'2021. He is a Fellow of The Canadian Academy of Engineering and an NSERC E.W.R. Steacie Memorial Fellow.



Shahram Dustdar (Fellow, IEEE) received the Ph.D. degree in business informatics from the University of Linz, Linz, Austria, in 1992.

He is currently a Full Professor of Computer Science (Informatics) with a focus on Internet technologies heading the Distributed Systems Group, TU Wien, Wien, Austria. He has been the Chairman of the Informatics Section of the Academia Europaea since December 2016.

Prof. Dustdar was a recipient of the ACM Distinguished Scientist Award in 2009 and the IBM Faculty Award in 2012. He is an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology*. He is on the editorial board of IEEE. He has been a member of the IEEE Conference Activities Committee since 2016, the Section Committee of Informatics of the Academia Europaea since 2015, and the Academia Europaea: The Academy of Europe, Informatics Section since 2013.



Jun Luo (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from EPFL (Swiss Federal Institute of Technology Lausanne), Lausanne, Switzerland, in 2006.

From 2006 to 2008, he has worked as a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. In 2008, he

joined the faculty of the School of Computer Science and Engineering, Nanyang Technological University, Singapore, where he is currently an Associate Professor. His research interests include mobile and pervasive computing, wireless networking, machine learning and computer vision, as well as applied operations research. More information can be found at <http://www.ntu.edu.sg/home/junluo>.